IMPERIAL COLLEGE LONDON
DEPARTMENT OF COMPUTING

Analysing datafied life

Xian Yang

05/2016

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Computing of Imperial College London and the Diploma of Imperial College London, May 2016

Abstract

Our life is being increasingly quantified by data. To obtain information from quantitative data, we need to develop various analysis methods, which can be drawn from diverse fields, such as computer science, information theory and statistics. This thesis focuses on investigating methods for analysing data generated for medical research. Its focus is on the purpose of using various data to quantify patients for personalized treatment.

From the perspective of data type, this thesis proposes analysis methods for the data from the fields of Bioinformatics and medical imaging. We will discuss the need of using data from molecular level to pathway level and also incorporating medical imaging data. Different preprocessing methods should be developed for different data types, while some post-processing steps for various data types, such as classification and network analysis, can be done by a generalized approach. From the perspective of research questions, this thesis studies methods for answering five typical questions from simple to complex. These questions are detecting associations, identifying groups, constructing classifiers, deriving connectivity and building dynamic models. Each research question is studied in a specific field. For example, detecting associations is investigated for fMRI signals. However, the proposed methods can be naturally extended to solve questions in other fields.

This thesis has successfully demonstrated that applying a method traditionally used in one field to a new field can bring lots of new insights. Five main research contributions for different research questions have been made in this thesis. First, to detect active brain regions associated to tasks using fMRI signals, a new significance index, CR-value, has been proposed. It is originated from the idea of using sparse modelling in gene association study. Secondly, in quantitative Proteomics analysis, a clustering based method has been developed to extract more information from large scale datasets than traditional methods. Clustering methods, which are usually used in finding subgroups of samples or features, are used to match similar identities across samples. Thirdly, a pipeline originally proposed in the field of Bioinformatics has been adapted to multivariate analysis of fMRI signals. Fourthly, the concept of elastic computing in computer science has been used to develop a new method for generating functional connectivity from fMRI data. Finally, sparse signal recovery methods from the domain of signal processing are suggested to solve the underdetermined problem of network model inference.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Declaration of originality

I declare that the content of thesis is composed by myself, and the work it presents is my own. All use of the previously published work of others has been listed in the bibliography.

Acknowledgements

I would like to express my thanks to:

- my supervisor, Professor. Yike Guo, for his support and professional guidance. Without his continuous encouragement and great patience, this thesis would not be possible.
- all my lovely friends and colleagues in Imperial College London for their constant help in all aspects of my research as well as their friendship that brought much happiness to my life at Imperial.
- my parents and my husband for their constant support, love and encouragement.

Dedication

To my dearest Baba, Mama, husband and daughter.

“A model is a lie that helps you see the truth.”

Howard Skipper

Table of Contents

Abstract 1

Copyright Declaration 2

Declaration of originality 3

Acknowledgements 4

Dedication 5

1 Introduction 24

1.1 Motivation and objectives 24

1.2 Contributions 26

1.3 Thesis organization 28

1.4 Publications 28

2 Medicine as data science 33

2.1 Introduction 33

2.2 Using various molecular data 36

2.2.1 Roles of molecular datasets in medical practice 36

2.2.2 Processing high throughput molecular data 38

2.3 Moving from molecular level to pathway level 41

2.3.1 Mapping molecules to pathways 41

2.3.2 Modelling pathways 41

2.4 Incorporating imaging data 44

2.4.1 Roles of imaging data in medical practice 44

2.4.2 Using functional MRI in brain disease study 44

2.5 Similarities of computational methods for different data types 52

2.6 Summary of analysis methods 56

2.7 Conclusion 57

3 Questions and methods in datafied life 59

3.1 Introduction 59

3.2 Detecting associations 59

3.2.1 1-to-1 association 60

3.2.2 1-to-N association 61

3.2.3 Our interest: M-to-N association 68

3.3 Identifying groups 69

3.3.1 Clustering analysis 69

3.3.2 PCA 72

3.3.3 TDA 72

3.3.4 Our interest: hierarchical clustering for large scale dataset 73

3.4 Constructing classifiers 74

3.4.1 Selection of classification methods 74

3.4.2 Construction of classifiers 76

3.4.3 Our interest: the balance of stability and predictive performance of features 77

3.5 Deriving connectivity 78

3.5.1 Methods of inferring skeleton 79

3.5.2 Methods of inferring skeleton and direction 81

3.5.3 Our interest: a parameter free inference method 82

3.6 Building dynamic models 82

3.6.1 Three main problem classes 82

3.6.2 Inference methods 83

3.6.3 Our interest: large scale network model inference under limited measurements 86
3.7 Conclusion 86
4 Detecting associations: a Lasso based approach 87
4.1 Motivation 87
4.2 Background 89
4.3 Method 92
4.3.1 Single-subject analysis 92
4.3.2 Multi-subject analysis 98
4.3.3 Permutation test 101
4.3.4 Smoothing parameter determination 102
4.4 Experimental results 105
4.4.1 Simulation analysis 105
4.4.2 Empirical study 118
4.5 Discussion 124
5 Grouping measurements: a clustering based approach 126
5.1 Motivation 126
5.2 Background 129
5.3 Method 131
5.3.1 Data binning 131
5.3.2 Ion matching within each bin 132
5.3.3 Adjustment between overlapping bins 134
5.3.4 Parallel implementation 137
5.3.5 The whole process 137

5.3.6 Mechanisms of handling batch effect 138
5.3.7 Performance evaluation 138
5.3.8 GUI 141
5.4 Experiment and results 143
5.4.1 Experimental design 143
5.4.2 Data Acquisition 144
5.4.3 Results 146
5.5 Discussion 163
6 Obtaining classifiers: a cross validation based approach 166
6.1 Motivation 166
6.2 Background 169
6.3 Method 171
6.3.1 Overview of our method 171
6.3.2 Preliminaries of SBL 173
6.3.3 Feature selection algorithm 176
6.3.4 Performance evaluation method to balance stability and predictive performance 178
6.3.5 The whole process 180
6.4 Experimental Results 181
6.4.1 Simulated data analysis 182
6.4.2 Real data analysis 190
6.5 Discussion 195
7 Deriving connectivity: an elastic approach 197

7.1 Motivation 197
7.2 Background 199
7.3 Method 201
7.3.1 Minimum partial correlation 201
7.3.2 PC algorithm 203
7.3.3 Our method: elastic PC algorithm (ePC) 207
7.4 Experimental results 211
7.4.1 Simulation dataset analysis 212
7.4.2 Real dataset analysis 216
7.5 Discussion 225
8 Building linear dynamic models: a sparse learning approach 227
8.1 Motivation 227
8.2 Background 228
8.3 Method 234
8.3.1 Inferring concentrations of species from observations 236
8.3.2 Finding active species whose concentrations vary over time 244
8.3.3 Inferring transition matrix for active species 244
8.3.4 Obtaining the complete transition matrix 246
8.3.5 Perturbation 247
8.4 Experiment and results 248
8.4.1 Experimental settings 249
8.4.2 Results 251
8.5 Discussion 264

9 Conclusions....266
9.1 Summary of thesis achievements....266
9.2 Future work....267
Bibliography....270

List of Tables

Table 3-1 Different distance functions in similarity matrix calculation 70

Table 3-2 Typical linkage functions 71

Table 4-1 Surface area vs. accuracy, specificity, sensitivity and AUROC in single-subject analysis 113

Table 4-2 Surface area vs. accuracy, specificity, sensitivity and AUROC in multi-subject analysis. 118

Table 4-3 Surface areas of single-subject analysis with the smoothing parameter varies from 1 to 10. 121

Table 5-1 The format of output matrix generated by the ion matching method 139

Table 5-2 The format of matrix containing peptide identities of DRAMIs 139

Table 5-3 An example to show ions with the same peptide identity appear in different rows 140

Table 5-4 An example to show some rows whose ions do not have identical peptide identity and their identities also appear in other rows 141

Table 5-5 Experimental design and rationale 143

Table 5-6 Standard mix compositions 145

Table 5-7 Ratios of top three intensity sums for four known proteins. 151

Table 5-8 Details of sub-datasets of Dataset 3 154

Table 5-9 A summary of matching results 158

Table 5-10 Examples of Case 1 159

Table 5-11 Examples of Case 2 160

Table 5-12 Examples of Case 3 161

Table 5-13 A summary of the matching results for Dataset 4 162

Table 6-1 Results obtained by the accuracy method with different pairs of L and M. 186

Table 6-2 Results obtained by the stability method with different pairs of L and M.....187

Table 6-3 Results obtained by our method with different pairs of L and M. 187

Table 6-4 Optimal results from different methods for Dataset 2 and Dataset 3 189

Table 6-5 Optimal results from different methods for Subject 1 and Subject 2. 194

Table 7-1 Values of c-sensitivity (%) for 28 simulation networks using different methods. 213

Table 7-2 Links detected by different methods. 221

Table 8-1 Details of 6 benchmark networks 250

Table 8-2 RMSE of the matrix A for all 6 networks under different number of perturbations 257

Table 8-3 AUROC values of network structure inference for all 6 benchmark networks under different number of perturbations 259

List of Figures

Figure 2-1 An illustration of using various data types for disease prediction (from (Winslow et al. 2012)). 34

Figure 2-2 A pipeline of treating patients according to the results of predictive model (from (Coveney et al. 2014)) 35

Figure 2-3. The steps of carrying out the UBIOPRED project. 37

Figure 2-4 The overview of Proteomics analysis in UBIOPRED project 40

Figure 2-5 Process of obtaining pathway models for understanding sub-phenotypes 42

Figure 2-6 The integrated pathway of p38 MAPK and GR from (Holehouse et al. 2012) 43

Figure 2-7 The overview of FSL preprocessing steps. 47

Figure 2-8 The overview of the study in (Tomassini et al. 2012) 48

Figure 2-9 Examples of various fMRI analyses (from (Turk-Browne 2013)) 51

Figure 2-10 Schematic summary of the conceptual basis of DCM (from (Friston et al. 2003) and (Ashburner et al. 2013)). 52

Figure 2-11 Data preprocessing steps in Proteomics/Lipidomics Mass spectrometry and fMRI for similar purposes 54

Figure 2-12 Comparison of some analysis methods in Bioinformatics and fMRI analysis. 55

Figure 2-13 A summary of analysis methods for answering research questions from simple to complex (from (Wolkenhauer 2014)). 58

Figure 3-1 Contour plots of different prior distributions over two parameters (Tipping 2006). 66

Figure 3-2 Summary of statistical measurements of performance (from (Wikipedia n.d.)) 76

Figure 3-3 The process of constructing a classifier for prediction on an independent dataset via cross validation 78

Figure 3-4 Examples of using partial correlation to infer direct links. 80

Figure 3-5 The outputs of the adaptive weighting ABC SMC method as scatter plots. 85

Figure 4-1 The illustration of linear models. 88

Figure 4-2 The fMRI analysis for significant voxel detection using GLM. 90

Figure 4-3 An example of neighbouring voxels. 93

Figure 4-4 An illustration of CR-value calculation. 97

Figure 4-5 Relations of matrices used in multi-subject analysis to detect significant features across all subjects. 99

Figure 4-6 Design matrix containing two regressors. 105

Figure 4-7 Design of simulations in two cases. 107

Figure 4-8 Single-subject analysis for Case1 using the task correlated regressor. 109

Figure 4-9 Single-subject analysis for Case2 using the task correlated regressor 110

Figure 4-10 Single-subject analysis for Case1 using the task correlated regressor and its derivative 111

Figure 4-11 Single-subject analysis for Case2 using the task correlated regressor and its derivative 112

Figure 4-12 The boxplots of AUROC values obtained from 100 samples of simulation data for single-subject analysis. 115

Figure 4-13 Multi-subject analysis for Case1 using the task correlated regressor. 116

Figure 4-14 Multi-subject analysis for Case2 using the task correlated regressor. 117

Figure 4-15 Results of single-subject analysis using coefficients, GLM P-values and CR-values. 120

Figure 4-16 Histograms of CR-values in single-subject analysis. 121

Figure 4-17 Multi-subject analysis using group GLM P-values and CR-values. 122

Figure 4-18 An example of differences in group CR-values and group GLM P-values. 123

Figure 5-1 An example plot of fragments in a sample file in three measurement directions. 126

Figure 5-2 Intensities of fragments in different samples along the m/z direction. 127

Figure 5-3 An illustration of grouping similar fragments. 128

Figure 5-4 An example to show the adjustment method. 135

Figure 5-5 Flowchart of doing ion matching in parallel 137

Figure 5-6 The graphic user interface of the developed tool 142

Figure 5-7 Output of the matching algorithm in 3D space defined by m/z, retention time and drift axis. 147

Figure 5-8 Boxplots of MZ difference. 148

Figure 5-9 Boxplots of DT difference. 148

Figure 5-10 Boxplots of RT difference. 149

Figure 5-11 Boxplots of count. 150

Figure 5-12 Boxplots of differences in MZ, DT and RT between the mean values of identified ions and unidentified ions within each ion group. 151

Figure 5-13 Boxplot for ratios between Pepmix 1 and Pepmix 2 of other proteins. 152

Figure 5-14 Heatmap and dendrogram of Sub-dataset 1. 155

Figure 5-15 Heatmap and dendrogram of Sub-dataset 2. 156

Figure 5-16 Heatmap and dendrogram of Sub-dataset 3. 157

Figure 5-17 Heatmap and dendrogram of Sub-dataset 4. 158

Figure 6-1 The process of cross validation for classification 166

Figure 6-2 The process of cross validation with the embedded feature selection approach. 168

Figure 6-3 An illustration of the MVPA study (from (Norman et al. 2006)). 170

Figure 6-4 The flowchart of optimal feature selection through cross validation 172

Figure 6-5 The flowchart of the RS-SBL feature selection algorithm.....178

Figure 6-6 An example of x to show the measurement matrix is locally correlated. 182

Figure 6-7 Values of $\widetilde{P}_q$ with q varying from 1 to 30. 184

Figure 6-8 Values of $\widetilde{P}_q$ with q varying from 1 to 30. 185

Figure 6-9 Plots of scores obtained from different methods with $L, M = (100,1600)$.....190

Figure 6-10 Values of $\widetilde{P}_q$ with q varying from 1 to 100 for subject 1. 192

Figure 6-11 Values of $\widetilde{P}_q$ with q varying from 1 to 100 for subject 2. 193

Figure 7-1 Examples of getting indirect links due to transitive effect.....198

Figure 7-2 An example of the elastic PC algorithm. 210

Figure 7-3 Values of c-sensitivity obtained at different significant levels from the elastic PC algorithm for 28 simulated networks. 215

Figure 7-4 The proportion of saved computation efforts at different significant levels from the elastic PC algorithm for 28 simulated networks. 216

Figure 7-5 Images of matrix $\widehat{\Omega}$ from different methods.....218

Figure 7-6 The proportion of saved computation efforts at different significant levels from the elastic the PC algorithm for a real fMRI dataset.....220

Figure 7-7 The changes between results with two adjacent significant levels from the elastic PC algorithm. 220

Figure 7-8 Links detected by various methods among the following ROIs: Calcarine, Lingual, Cuneus and Occipital_Sup. 223

Figure 7-9 The top 1% links selected by the elastic PC algorithm. 224

Figure 8-1 Comparison of reductionist and holistic approaches. 231

Figure 8-2 An illustration of sparse signal recovery.....233

Figure 8-3 The flowchart of the whole inference process. 235

Figure 8-4 RMSE of Δx_t for all 6 benchmark networks under different number of perturbations using Group SBL and SBL. 252

Figure 8-5 Boxplots of estimation errors of Δx_t for all 6 benchmark networks under different perturbations using Group SBL. 253

Figure 8-6 Boxplots of estimation errors of Δx_t for all 6 benchmark networks under different perturbations using SBL. 253

Figure 8-7 Image of Δx_t for the p58 network under 3 perturbations from Group SBL.....254

Figure 8-8 Image of Δx_t for the p58 network under 3 perturbations from SBL.....255

Figure 8-9 The sensitivity and specificity of detecting real time-variant species by changing the value of threshold ε_x. 256

Figure 8-10 Bar charts of RMSE for transition matrix A inference.....257

Figure 8-11 ROC curves of network structure inference.....259

Figure 8-12 The sensitivity and specificity of detecting real network structure by changing the value of threshold ε_a. 260

Figure 8-13 RMSE of Δx_t for the n-39 network under 6 perturbations with different levels of measurement noise.....261

Figure 8-14 RMSE of A for the n-39 network under 6 perturbations with different levels of measurement noise.....262

Figure 8-15 RMSE of Δx_t for the n-39 network under 6 perturbations with different number of measurements. 263

Figure 8-16 RMSE of A for the n-39 network under 6 perturbations with different number of measurements.....263

List of abbreviations

ABC	Approximate Bayesian Computation
AUROC	Area under receiver operating characteristic curve
BMI	Body mass index
BOLD	Blood oxygen level-dependent
CID	Collision induced dissociation
CR-value	Critical regularization value
CS	Corticosteroids
CT	Computed tomography
CV	Cross-validation
DCM	Dynamic causal modelling
DRAMI	Drift, retention time, accurate mass, intensity
DT	Drift time
EKF	Extended Kalman Filter
EM	Expectation maximization
ePC	Elastic PC algorithm
FDR	False discovery rate
FP	False positive
FN	False negative
fMRI	Functional magnetic resonance imaging
FWHM	Full width at half maximum
GIANT	Genetic investigation of anthropometric traits
GLM	General linear model

GR	Glucocorticoid receptor
GS	Global silencing
GUI	Graphical user interface
GWAS	Genome-wide association study
HCP	Human Connectome Project
HDR	Hemodynamic response
HRF	Haemodynamic response function
ICOV	Regularized inverse covariance
LARS	Least angle regression
LASSO	Least absolute shrinkage and selection operator
LC-IMS/MS^{E} instrument	Ion mobility supported lipid chromatography and mass spectrometry
LC-MS/MS	Lipid chromatography tandem mass spectrometry
LiNGAM	Linear, Non-Gaussian, Acyclic causal Models
LC	Liquid Chromatography
LDPC	Low-density parity-check code
LOOCV	Leave-one-out cross-validation
LOWESS	Local weighted scatterplot smoothing
LPS	Lipopolysaccharide
MAPK	Mitogen Activated Protein Kinase
MCMC	Markov Chain Monte Carlo
MIC	Maximal information coefficient
MKP-1	MAPK phosphatase-1
MPC	Minimum partial correlation

MRI	Magnetic resonance imaging
MS	Mass spectrometry
MVPA	Multivariate Pattern Analysis
MZ	Accurate mass
ND	Network deconvolution
OLS	Ordinary least squares regression
PCA	Principle component analysis
PET	Positron emission tomography
PLGS	ProteinLynx Global Server
PPM	Posterior probability map
ROC	Receiver operating characteristic
RBC	Red blood cell
RMSE	Root-mean-square deviation
ROI	Region of interest
RS	Random subspace
RSS	Residual sums of squares
RT	Retention time
SAMBA	Statistical-Algorithmic Method for Bicluster Analysis
SBL	Sparse Bayesian learning
SMC	Sequential Monte Carlo
SNP	Single-nucleotide polymorphisms
SPM	Statistical Parametric Mapping
SVM	Support vector machine

SWATH	Sequential windowed acquisition of all theoretical fragment ion mass spectra
TAK1	TGFβ kinase-1
TDA	Topological data analysis
TFCE	Threshold free cluster enhancement
TP	True positive
TWIM	Travelling wave ion mobility
UBIOPRED	Unbiased BIOmarkers in PREDiction of respiratory disease outcomes
UKF	Unscented Kalman Filter

1 Introduction

1.1 Motivation and objectives

Nowadays, our life is being intensively quantified by data, which we call it as datafied life. It is a hot topic to maximally extract knowledge from large volume of data, which can be generated from emails, social media, health records, shopping habits, daily exercise records, lab experiments, etc. Analysis methods drawn from various fields, including computer science, statistics, mathematics and information theory are of particular interest to uncover the pattern of data. This thesis focuses on investigating analysis methods to solve problems in the domain of medical research.

One trend in medical practise is personalization. That is, medical decisions and treatments are being tailored to individual patient based on the context of patient's various profiles such as Genomics, Proteomics, Lipidomics and Metabolomics content. High throughput instruments are used to generate large scale datasets. To succeed in personalized medicine, analysis methods, including those dedicated to specific data types and those shared among various data, should be well developed. As it is impossible to investigate methods for all different datasets on various research purposes, this thesis studies methods to answer some typical research questions in analysing quantitative datasets, such as association detection, classification model construction and network analysis.

This thesis attempts to investigate questions raised in both Bioinformatics and medical imaging processing fields. Bioinformatics is interested in developing methods and software tools for the understanding of biological data as well as explaining biological processes. It is an interdisciplinary research field, combining computer science, statistics with biology. My research in Bioinformatics is inspired by a project called UBIOPRED, which I will discuss in detail in Chapter 2. When I am doing research in Bioinformatics, I find there are lots of similar problems raised in other fields. Some common analysis methods can be used across fields. Medical imaging is a new type of life science, processing images of the human body for medical science. One typical imaging technique is functional magnetic resonance imaging (fMRI). Processing fMRI data requires the knowledge of computer science and mathematics as well. This thesis also investigates some analysis methods for fMRI analysis.

The main objective of this thesis is to show that analysis methods, which are traditionally used in a specific research field, can be potentially adapted to other fields bringing a massive amount of new insights into research questions. This thesis aims at highlighting and

demonstrating the importance of applying cross-filed analysis methods. Analysis methods proposed in this thesis are motivated by the following questions observed in Bioinformatics and fMRI:

- One typical fMRI analysis focuses on finding significant voxels activated by stimuli. The performance of active voxel detection is greatly influenced by the noise level. When there are large noises, significant voxels are likely to be wrongly detected. To improve sensitivity and accuracy of signal detection from highly noisy fMRI data, potentially important information, such as spatial smoothness and connectivity constraints, therefore should be incorporated. Our objective is using spatial smoothness to facilitate active voxel detection in fMRI analysis.
- Large amount of datasets are being generated by mass spectrometry machines to quantify Proteomics, Lipidomics and Metabolic profiles of biological samples. Proteins, lipids and metabolites are segmented into small compounds; and the chemical and physical characteristics of their fragments are quantified in the machine. Mass spectrometry machines record measurements of samples in individual files. Each file contains a list of fragments, where each fragment is specified by quantitative values from measurement. To compare the intensity of fragments across samples, we first need to group similar fragments from different samples using their quantitative measurements. This is for the case that the identities of fragments are not known. Due to variations of instrument and experimental environment, quantitative measurements of fragments referring to the same identity may be slightly different for different samples. Our objective is developing a method to group similar measurements across samples.
- Multivariate Pattern Analysis (MVPA) in fMRI decodes patterns of brain activity by constructing predictive models using signals from multiple voxels of fMRI images. A recent trend in MVPA is generating a predictive map as well as a classifier. Predictive map contains relevant voxels used for prediction, which can indicate which brain regions participate in a functional brain activity. It is therefore meaningful that such a predictive map is stable with respect to slight perturbation of data. Our objective is to find a voxel set that is robust to the variation of datasets and also has predictive power.
- In neuroscience, functional connectivity can help to explore relationships among different function regions of a brain in completion of a task and further understand complex brain activities. The functional connectivity is constructed by detecting the

temporal correlation between spatially remote neurophysiological events. Correlation, mutual information or other similarity metric between each pair of nodes can be used. However, the resulting network contains a massive amount of indirect links owing to transitive effects. Thus more sophisticated methods, such as Bayes net and graphic Lasso methods are being used. The performance of these methods is dependent on some parameters. Our objective is to propose a parameter-free method for functional connectivity inference.

- Mechanistic models of biological pathways can describe the time evolution of intracellular molecules in response to stimuli and give a detailed insight into pathway dynamics. The model construction requires quantitative dataset, collected from wet-lab experiments, measuring the dynamic behaviour of networks. The number of time-course measurements and the interval between two sequential measures are crucial in the network modelling. Lack of measurements or too long observation gap will result in impaired observation of real network dynamics and hence incorrect models. Therefore, to obtain a sound network model, we need to carry out sufficient measurements. However, measuring all species in a cell independently is quite challenging and may be even impossible. Our objective is to propose an idea to infer network model under limited measurements.

1.2 Contributions

This thesis develops a set of analysis methods for answering research questions from simple to complex, which are detecting associations, identifying groups, constructing classifiers, deriving connectivity and building dynamic models. The proposed methods can be generalized and adapted to various datasets. This thesis gives a comprehensive study in quantitative data analysis. The main contribution is successfully adapting methods originally used in one particular field to other fields. More specifically, this thesis makes the following contributions:

- An extension of Lasso based approach, originated from gene association study, is developed to detect active voxels in fMRI analysis. The performance is enhanced by incorporating smooth constraints based on the observation that that active voxels are typically found adjacent to other active voxels in a cluster. We propose a new index, the critical regularization value (CR-value), to find significant voxels. We also extend formulation of the CR-value for multi-subject analysis. The experiment results show that using CR-value active voxels can be accurately detected.

- A hierarchical clustering based approach, frequently used for sample and feature subgroup detection, is proposed to group similar chemical identities across different samples which have similar quantitative measurements. As high throughput instruments are increasingly used to generate quantitative dataset, one challenge is how to make clustering work on large scale dataset. Processing large dataset requires large memory and processing power. Especially, hierarchical clustering needs to generate large distance matrices for large datasets. We introduce a binning process to facilitate the application of hierarchical clustering to group similarity measurements for large scale datasets. Our experiment shows that using our approach, some discriminant features (e.g. Lysophosphatidylcholine), which cannot be discovered by traditional methods, can be detected by our method.
- The method to balance the predictive performance and stability of feature selection, originally proposed for biomarker discovery in the field of Bioinformatics, is adapted to fMRI analysis. By applying this method, a predictive model for brain activities can be obtained and in the meanwhile brain regions that are related to activities can be acquired. A set of simulation data and real data collected from the Human Connectome Project show that the proposed method works better than traditional methods.
- An elastic approach, the concept raised in cloud computing, is developed to infer functional connectivity for fMRI signals. The proposed elastic PC method can approach to the real network structure given no time constraint. With a time constraint, the elastic method can maximally approximate the network structure by using the calculations from previous steps. We illustrate our method using the resting-state fMRI data of 10 unrelated subjects from the Human Connectome Project. The results show that the proportion of saved computation effort varies from 9.2% to 72%.
- An idea of using sparse signal recovery methods, popularly used for signal processing, is generated to infer large scale network model under limited measurements. It is based on the following observations: 1) as different biological reactions occur over various timescales, concentrations of only a few species in a cell vary significantly in each specific timescale whilst a large fraction of species remain stable; 2) molecules which share some structural similarities can be measured together. These two observations meet the two prerequisites of sparse signal recovery respectively. Therefore, we suggest using sparse signal recovery methods to infer networks without

measuring all involved species individually. To obtain the sparsity of variations, we suggest investigating cell-scale networks at different timescales separately, such as seconds, minutes, hours, days and weeks. In each timescale, the combined-measurement is designed by considering the cross-activity of antibodies. We propose a method based on sparse Bayesian learning to infer the model for each timescale, explaining the observed dynamic relationships among molecules. In the simulation experiments, predictive performance can be maintained while the number of measurements (antibodies) is reduced to 50% of the total number of proteins.

1.3 Thesis organization

This thesis is organised as follows:

Chapter 2 introduces the background of this thesis. It shows the necessity of applying analysis methods for various data in medical research. Molecular data, pathway data and medical data has been discussed. Examples of showing similarities in different data type analysis are given.

Chapter 3 discusses five main research questions that are frequently asked in datafied life research. It reviews various methods for solving different questions and also points out our research interests.

Chapter 4 introduces a linear modelling approach to detect active voxels in fMRI analysis.

Chapter 5 introduces a clustering based approach to group similar measurements in Proteomics mass spectrometry data analysis.

Chapter 6 introduces a cross validation based approach to find voxels which have predictive performance and also stable to perturbation of data.

Chapter 7 introduces an elastic approach to generate functional connectivity in fMRI analysis.

Chapter 8 introduces a new idea of applying sparse learning methods to infer biological network model.

1.4 Publications

- **The Critical Regularization Value: Incorporating Spatial Smoothness to Enhance Signal Detection in Highly Noisy fMRI Data.** In: *7th international IEEE/EMBS conference on neural engineering. 2015*(Yang et al. 2015)

Comparing serially acquired fMRI scans is a typical way to detect functional brain changes in different conditions. However, this approach introduces additional variation on physical and physiological conditions, which results in substantial noise. To improve sensitivity and accuracy of signal detection in such highly noisy fMRI data, potentially important information should be incorporated. This paper proposes a new significance indicator, the critical regularization value (CR-value), which detects significantly changed voxels by taking both the magnitude of the voxel-wise signal variation and spatial smoothness into account. The CR-value allows voxels that survive in a stronger sparse constraint to be considered as more significant. We demonstrate our method using a simulation dataset and a real fMRI dataset collected from the previous study.

- **Minimum Partial Correlation: An Accurate and Parameter-Free Measure of Functional Connectivity in fMRI.** *In: 2015 international conference on brain informatics and health. 2015* (Nie et al. 2015)

  The most widely used method for inferring functional connectivity is full correlation, but it cannot differentiate direct and indirect effects. This disadvantage is often avoided by fully partial correlation, but this method suffers from Berkson's paradox. Some advanced methods, such as regularised inverse covariance and Bayes nets, have been applied. However, the connectivity inferred by these methods usually depends on crucial parameters. This paper suggests minimum partial correlation as a parameter-free measure of functional connectivity in fMRI. An algorithm, called elastic PC-algorithm, is designed to approximately calculate minimum partial correlation. Our experimental results show that the proposed method is more accurate than full correlation, fully partial correlation, regularised inverse covariance, network deconvolution algorithm and global silencing algorithm in most cases.

- **Balancing the Stability and Predictive Performance for Multivariate Voxel Selection in fMRI Study.** In: *The 2014 International Conference on Brain Informatics and Health. 2014* (Yan et al. 2014)

  Classical MVPA methods select voxels based on their prediction power; the selected ones are those that provide the best prediction performance. This accuracy based voxel selection method can guarantee the prediction performance, but it cannot ensure that all the selected ones are relevant. The interpretation of brain activity is therefore not ideal. This paper addresses this issue by introducing the concept of stability to the

MVPA studies. If only the stability is emphasized in the selection process, the probability of selecting irrelevant voxels is highly reduced with the sacrifice of the prediction accuracy. This paper, therefore, proposes a method to combine the stability assessment with the prediction accuracy assessment.

- **Integration of Sparse Bayesian Learning and Random Subspace for fMRI Multivariate Analysis.** In: *The 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2014* (Yan et al. 2014)

  Multivariate Pattern Analysis (MVPA) is frequently used to decode cognitive states from brain activities in fMRI study. Due to the discrepancy between sample and feature size, MVPA methods have suffered from the overfitting problem. This paper proposes a novel MVPA method to integrate linear Sparse Bayesian Learning with random subspace method. Benefiting from the random subspace method, spatial correlation and feature-to-sample ratio are largely reduced.

- **Inferring cell-scale signalling networks via compressive sensing.** In: *PloS one 9, no.4. 2014* (Nie et al. 2014)

  Signalling network inference is a central problem in system biology. Previous studies investigate this problem by independently inferring local signalling networks and then linking them together via crosstalk. Since a cellular signalling system is in fact indivisible, this reductionist approach may have an impact on the accuracy of the inference results. Preferably, a cell-scale signalling network should be inferred as a whole. However, the holistic approach suffers from three practical issues: scalability, measurement and overfitting. Here we make this approach feasible based on two key observations: 1) variations of concentrations are sparse due to separations of timescales; 2) several species can be measured together using cross-reactivity. This paper proposes a method, CCELL, for cell-scale signalling network inference from time series generated by immunoprecipitation using sparse Bayesian learning. Instead of exhaustively measuring all individual species, high accuracy is achieved from relatively few measurements.

- **An iterative parameter estimation method for biological systems and its parallel implementation.** In: *Concurrency and Computation: Practice and Experience 26, no. 6: 1249-1267. 2014* (Yang et al. 2014)

  One difficulty in building a mechanistic model of biological systems lies in determining correct parameter values. This paper proposes a novel parameter

estimation method to infer unknown parameters, such as kinetic rates, from noisy experimental observations. Derived from the approximate Bayesian computation sequential Monte Carlo algorithm, our method predicts the distribution of each parameter rather than a single value via several intermediate distributions. Motivated by the computational intensity of the method, we improve the approximate Bayesian computation sequential Monte Carlo method in two aspects. First, to increase the efficiency, a windowing method is developed to reduce the parameter-searching space, and an adaptive sampling weight mechanism is introduced to make the intermediate distributions converge to the target distributions in a much quicker manner. Second, to speed up the estimation process, we implement our method in a parallel computing environment.

- **Modelling and performance analysis of clinical pathways using the stochastic process algebra PEPA.** In: *BMC bioinformatics 13.Suppl 14: S4. 2012* (Yang et al. 2012)

  Clinical pathway informatics is regarded as an efficient way to solve a series of hospital challenges. To date, conventional research lacks a mathematical model to describe clinical pathways. Existing vague descriptions cannot fully capture the complexities accurately in clinical pathways and hinders the effective management and further optimization of clinical pathways. Given this motivation, this paper presents a clinical pathway management platform, the Imperial Clinical Pathway Analyser (ICPA). By extending the stochastic model performance evaluation process algebra (PEPA), ICPA introduces a clinical-pathway-specific model: clinical pathway PEPA (CPP). ICPA can simulate stochastic behaviours of a clinical pathway by extracting information from public clinical databases and other related documents using CPP. Thus, the performance of this clinical pathway, including its throughput, resource utilisation and passage time can be quantitatively analysed.

- **Developing a novel integrated model of p38 mapk and glucocorticoid signalling pathways.** In: *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on. IEEE, 2012.* (Holehouse et al. 2012)

  Glucocorticoid (GC) resistance is a key mechanism by which traditional asthma treatments become ineffective for patients, yet the molecular characteristics of the associated regulatory changes are largely unknown. Significant evidence suggests that crosstalk between p38 Mitogen Activated Protein Kinase (MAPK) and GC signalling

pathways may contribute to this resistance. Based on a number of studies, a simplified GC signalling pathway model was developed and integrated with a pre-existing model of the p38 MAPK pathway. It is predicted that with experimental data, the validity and use of this model can be confirmed, corrections and updates can be made where necessary, and that through the two pathways' interface points the existence and scale of crosstalk can be examined.

- **Automating Mass Spectrometry Proteomics Analysis.** In: *Fourth International Conference on Bioinformatics and Computational Biology. 2012* (Yang et al. 2012)

  Mass spectrometry (MS) techniques give us an insight into protein abundance for different biological states. Proteins whose concentration levels are significantly different for the patients in different pathology states are regarded as biomarkers. The joint profiles of biomarkers can construct a classifier for disease state prediction. Hence, it is necessary to derive a MS data analysis pipeline which can optimally extract information from quantitative proteomic data. This paper proposes an automatic proteomic MS data analysis pipeline including data pre-processing, feature selection and classification steps.

- **System Biology Approach to Study Cancer Related Pathways.** In: *Systems Biology in Cancer Research and Drug Discovery. Springer Netherlands, 39-67. 2012* (Guo & Yang 2012)

  This book chapter discusses a system biology approach to investigate the mechanism of cancer, especially the characteristic of the p53 pathway. Pathway modelling methods together with parameter estimation approaches are used to simulate the dynamic features of biological pathways. The core regulation part of the p53 pathway is analysed as an example, where its network motifs are identified by model selection methods based on the observation of experimental data. With the constructed mechanistic model, biological pathways can be simulated through in silico experiments to facilitate cancer research.

2 Medicine as data science

2.1 Introduction

Can we imagine what will our children and grandchildren find when they visit their doctors in the later part of this century? 'They may encounter computer-based avatars of themselves, programmed with their individual genetic makeup and physiological conditions.'(Coveney et al. 2014). Using their personal information, doctors will develop personalized treatment strategies. 'One size fit all' medicine will be fully out of date.

Personalized medicine aims at providing tailor-made medical decisions, practices and products to individual patients. The term personalized medicine, precision medicine, stratified medicine and P4 medicine describe the same concept (Wimmer & Shohamy 2012). Personalized medicine represents a crucial paradigm shift in clinical medicine. In this mode of medical practice, appropriate diagnosis and optimal therapies are based on patients' physiological states, clinical records and molecular profiles. For example, the use of genetic content is predominant in personalized medicine, based on which patients can be stratified (R. Smith 2012). Modern techniques such as genome sequencing can reveal mutations in DNA that may potentially result in various diseases (Dudley & Karczewski 2013). Especially in cancer diagnosis, patients are treated based on the presence or absence of mutations in particular genes in their tumour genomes (Lu et al. 2014). We anticipate that genome sequencing will be used more frequently in the near future. Within a few years, direct-to-consumer business will be widely established to generate personalized genomics for individuals.

In order for the personalized medicine to come true, adequate computational methods need to be developed to analyse acquired data. For instance, to find whether a mutation in DNA is related to a certain disease, researchers often conduct a computational study, called Genome-wide association study (GWAS) (Pearson & Manolio 2008; Manolio 2010). The most common approach in GWA studies is investigating whether the allele frequency of each SNP is significantly different between the case and control groups, where Chi-square test is required (Clarke et al. n.d.). For quantitative phenotypic data, such as height, weight and gene expression, alternative statistics implemented in bioinformatics software (e.g., SNPTEST and PLINK) are commonly used (Purcell et al. 2007). After statistic tests have been done, the following steps include creating a Manhattan plot and correcting p values for multiple testings.

Along with methods dedicated for a specific data type, approaches to integrate various datasets from different sources are also required (Gomez-Cabrero et al. 2014; Joyce & Palsson 2006). We need to extract and integrate disease-related information which is implicitly stored in various data. Figure 2-1 from (Winslow et al. 2012) gives an illustration of using different molecular data for disease prediction. The subplot on the left of Figure 2-1 gives a list of molecules (save in L) and also their measurements (save in X). Investigating their roles in health and disease requires analysing relationships among them in the context of a network (subplot in the middle). Learning the distributions of concentrations of these molecules, not individually but collectively as a multivariate probability distribution, makes disease diagnosis possible (subplot in the right).

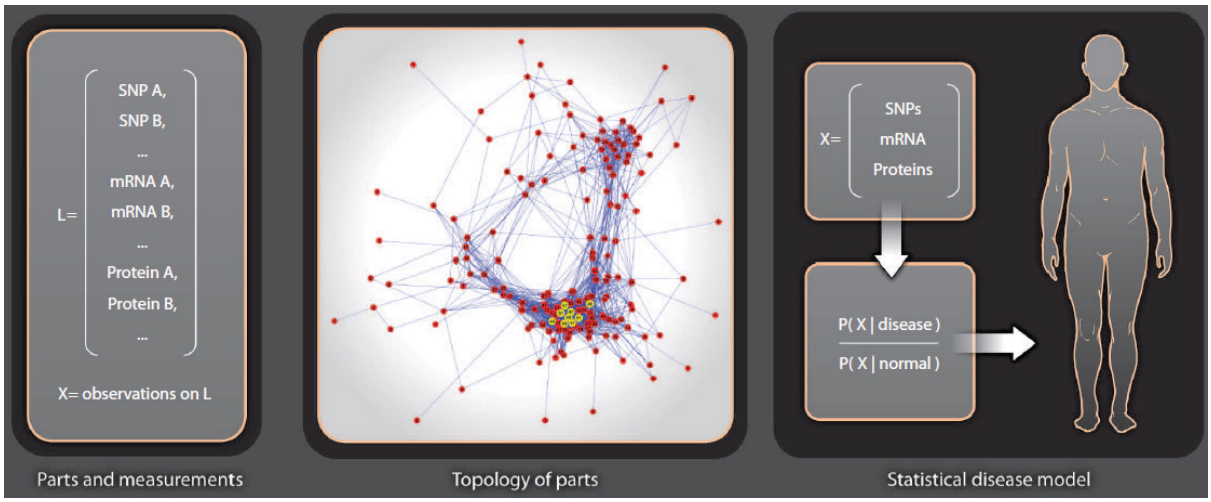


Figure 2-1 An illustration of using various data types for disease prediction (from (Winslow et al. 2012)). The panel on the left shows the names of molecules, where X contains their observations. The panel in the middle shows molecular networks constructed, where nodes correspond to molecules in L and edges represent molecular interactions. The panel on the right shows how a statistical model predicts disease based on the likelihood of X.

An example of using mixed datasets in medical practice is shown in Figure 2-2 from (Coveney et al. 2014). The pipeline in Figure 2-2 is to predict whether the tumour volume will shrink or not after preoperative chemotherapy in nephroblastoma. Data from diverse sources, including clinical data, imaging data and molecular data are fed into the pipeline to construct a predictive model, called as Oncosimulator. In the clinical trial, new patients can be randomized into two treatment arms: all patients in Arm A will receive the current

preoperative chemotherapy, while patients in Arm B will receive the treatment according to the prediction of the model. In Arm B, if the model predicts that the tumour volume will reduce after treatment, the patient will receive the preoperative chemotherapy; otherwise, the patient will directly go forward to surgery without first taking chemotherapy. By comparing the outcomes of two treatment arms, we can find the benefits of using the predictive model in the treatment. This example shows a way of treating patients using quantitative datasets. To get the pipeline running, analysis methods are of great importance in predictive model construction.

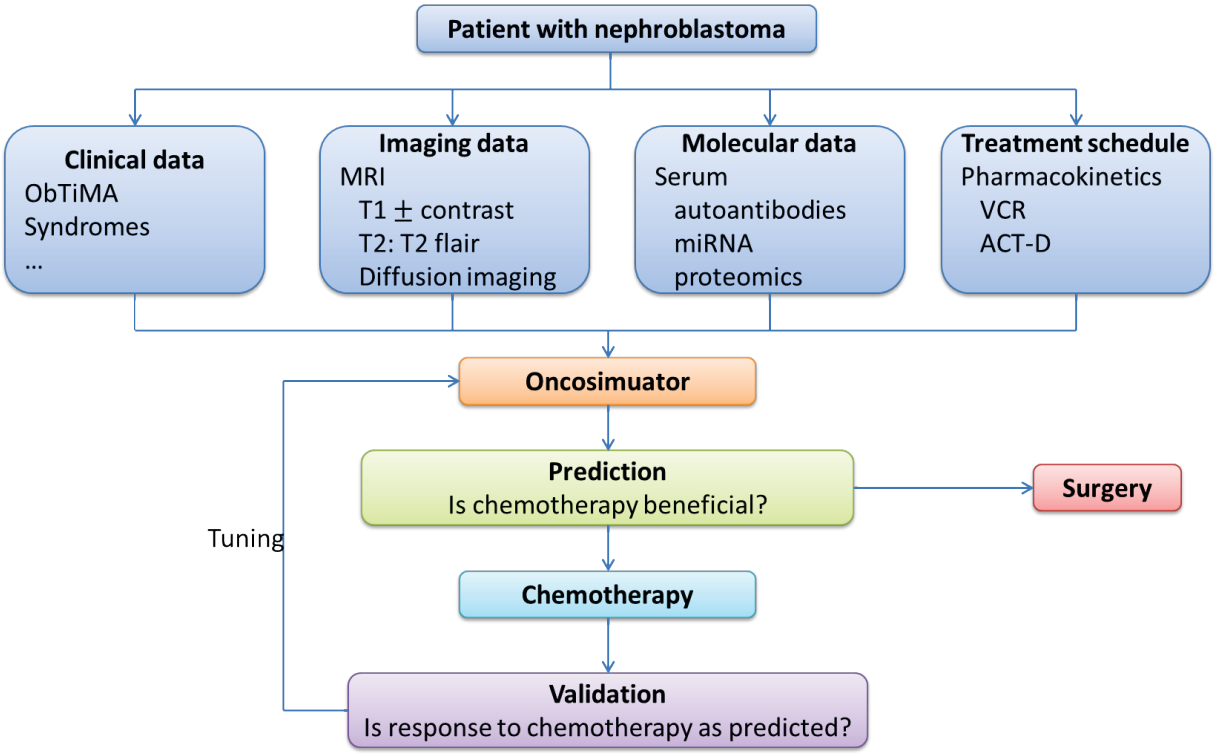


Figure 2-2 A pipeline of treating patients according to the results of predictive model (from (Coveney et al. 2014)).

In our vision, the modern medicine can be achieved mainly by quantitative data analysis. This chapter discusses doing the medical research in the discipline of data science. The following parts of this chapter discuss the use of various data types for medical study, including molecular level data and also imaging data. For better explanation, some examples will be introduced. For instance, the UBIOPRED project using various data for severe asthma phenotyping will be mentioned.

2.2 Using various molecular data

2.2.1 Roles of molecular datasets in medical practice

To succeed in personalized medicine, we should incorporate and integrate many different types of molecular and physiological data in medical research (Chen et al. 2012). Some typical molecular data types are gene sequencing, gene expression (measured by the microarray or RNA-seq technology) and protein expression (measured by the mass spectrometry technology) (Schneider & Orchard 2011). Different data types can profile the biological system in different aspects. By analysing them, we can for instance find biomarkers of a specific pathological state, or construct a predictor for disease diagnosis.

Let me use an example to show the importance of using molecular datasets. One important project I have taken part in is the UBIOPRED (Unbiased BIOmarkers in PREDiction of respiratory disease outcomes) project (Shaw et al. 2015). Using samples and medical information from hundreds of severe asthmatics, this project aims to identify different subtypes of severe asthma. It will potentially accelerate the discovery of novel diagnostic and therapeutic targets. One focus of UBIOPRED is that it generates various high-throughput Omics datasets, including GWAS, transcriptomics, Proteomics, Lipidomics and Breathomics to characterise severe asthma subtypes. Moreover, this project also generates other data types, such as histological, morphological, clinical and patient report outcome datasets. By profiling patients in different aspects, we can ultimately obtain a comprehensive model for asthma phenotyping.

Figure 2-3 shows the process of carrying out the UBIOPRED project: 1) Large adult and paediatric patient registries are created for cross-sectional and longitudinal cohort studies in well-characterised severe asthmatics and controls. 2) An unbiased and innovative systems biology strategy is used for classifying patients into distinct severe asthma phenotypes. High-dimensional analyses integrate a “handprint” of biomarkers that are derived from staged sifting of molecular (‘omics’), histological, clinical, PRO data, etc. 3) The accuracy of the generated handprints to identify severe asthma phenotypes are validated with regards to disease progression and onset/severity of exacerbations. 4) The handprints are refined with: a) pre-clinical animal and in vitro human models b) human experimental in vivo models of exacerbations and loss of control. The core step of this project, which is Step 2), largely relies on using and analysing various molecular data. Without using these data, we can hardly detect appropriate phenotypes and hence find drugs for subtyped patients.

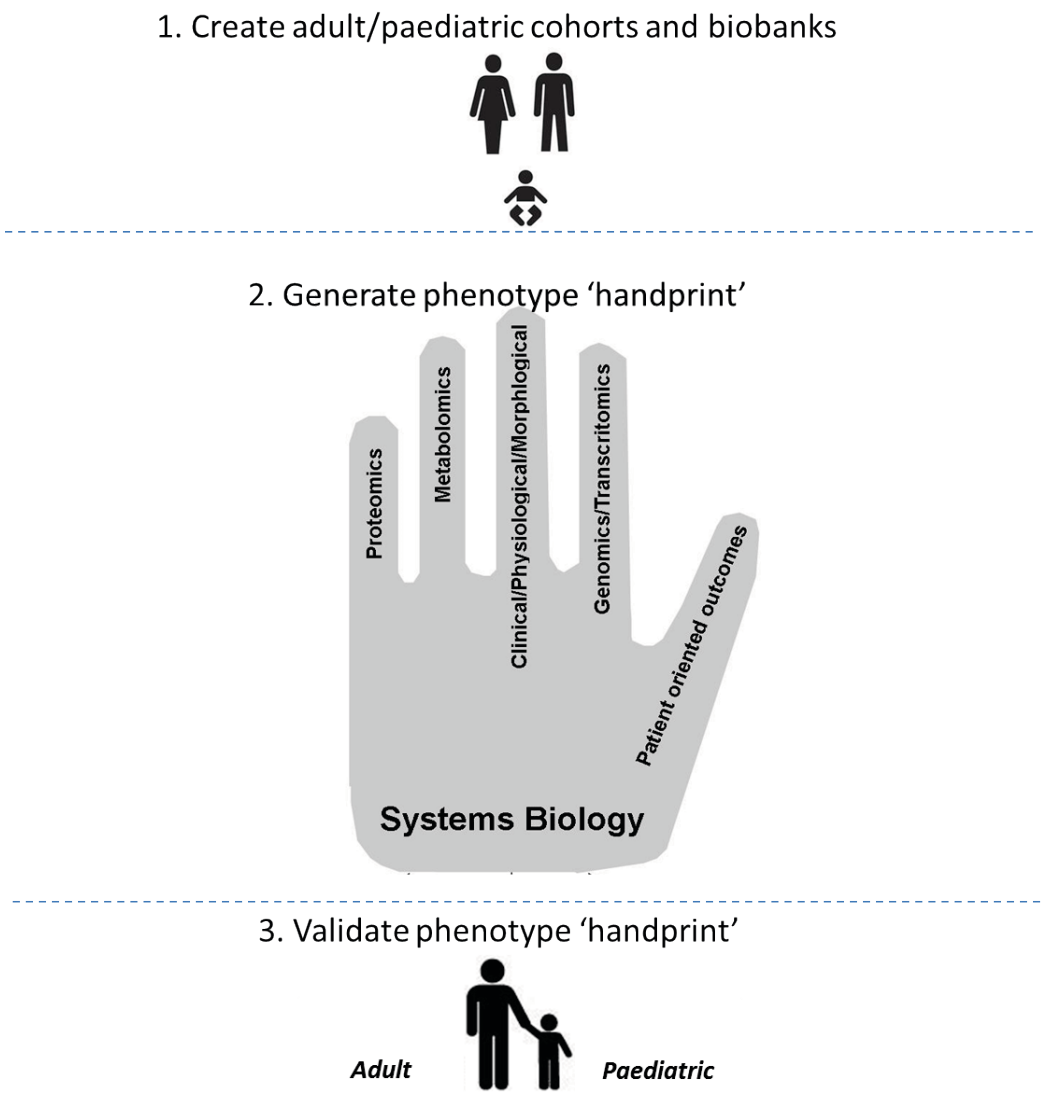


Figure 2-3. The steps of carrying out the UBIOPRED project.

2.2.2 Processing high throughput molecular data

Each type of molecular data requires its own techniques and methods for its profiling and analysis. Generally, the techniques that are used to generate molecular data are generally high throughput in nature. Thus, large scale quantitative datasets are created, giving challenges to data processing. As a consequence, each combination of data type and technology needs specific bioinformatics pipelines and methodologies for the analysis.

Let us take the analysis of high throughput datasets in UBIOPRED as an example. In UBIOPRED, among all kinds of different Omics datasets, Proteomics and Lipidomics datasets are created by mass spectrometry (MS) machine. MS technique helps identify the amount and type of chemicals in a sample by measuring its chemical and physical characteristics and also the abundance of gas-phase ions (Zhang et al. 2010). Here, we introduce the Proteomics data analysis pipeline.

The Proteomics MS dataset in UBIOPRED is processed following two approaches. The first approach is the traditional one, which is database dependent. ProteinLynx Global Server (PLGS) (Waters) produces curated mass data files from MS^E information, which we call DRAMI (Drift, Retention time, Accurate Mass, Intensity) (Silva et al. 2005). DRAMI are utilized for database searching in PLGS, producing peptide information files. Absolute quantitation of every identified peptide is achieved through the use of internal standards that have been introduced at known concentration. Only identified peptides are used in discriminant feature detection between asthma subtypes; ions which cannot be identified by database search will be discarded. Therefore, this approach only partly uses the information generated by high-throughput MS machine. Moreover, it is biased by the completeness of database. To overcome this problem, we have developed another approach that uses all information gathered in the mass spectrometer to find discriminant features between asthma subtypes. Once classifying features are determined, we can interrogate the data with the use of the ion accounting file, to identify whether these discriminating features are peptides or other molecules. Compared with other database dependent approaches, this approach is capable of providing more biomarker candidates for differential tests, from a larger pool of molecule types, and thus more information can be retrieved from the MS data.

Figure 2-4 summarises the overview of Proteomics analysis in the UBIOPRED project. 351 serum and 270 sputum samples will be processed in duplicate under quality control and experimental design. Then LC-IMS-MS^E (ion mobility supported Liquid Chromatography

and MS instrumentation) measures chemical identities in samples. It will take about 4025 hours instrument time. Next, ProteinLynx Global Server produces curated ion information. The identity driven approach only selects identified peptides. Peptides are matched across samples using their IDs. The output of this approach is a data matrix, where each element is the abundance of a peptide (row) from a sample (column). The spectral based approach, on the other hand, uses information from all ions including those with no peptide IDs. In this approach, since the vast majority of ions have no IDs, we cannot use peptide identity to match ions across samples. Therefore, we develop a clustering based method, which will be described in Chapter 5, to match similar ions using their quantitative measurements across samples. The output of this approach is a data matrix, where each column corresponds to a sample and ions in each row are regarded to be same chemical identities. With the outputs from these approaches, we can do further analyses. For example, we can use unsupervised machine learning methods, such as hierarchical clustering (Olson 1995) and topological data analysis (TDA) (Zomorodian 2011) to find subgroups of the dataset. Some supervised methods, such support vector machine (SVM) can then be used to construct the classification model of detected subgroups with optimized selection of features. Chapter 6 introduces a method of returning an optimal feature set.

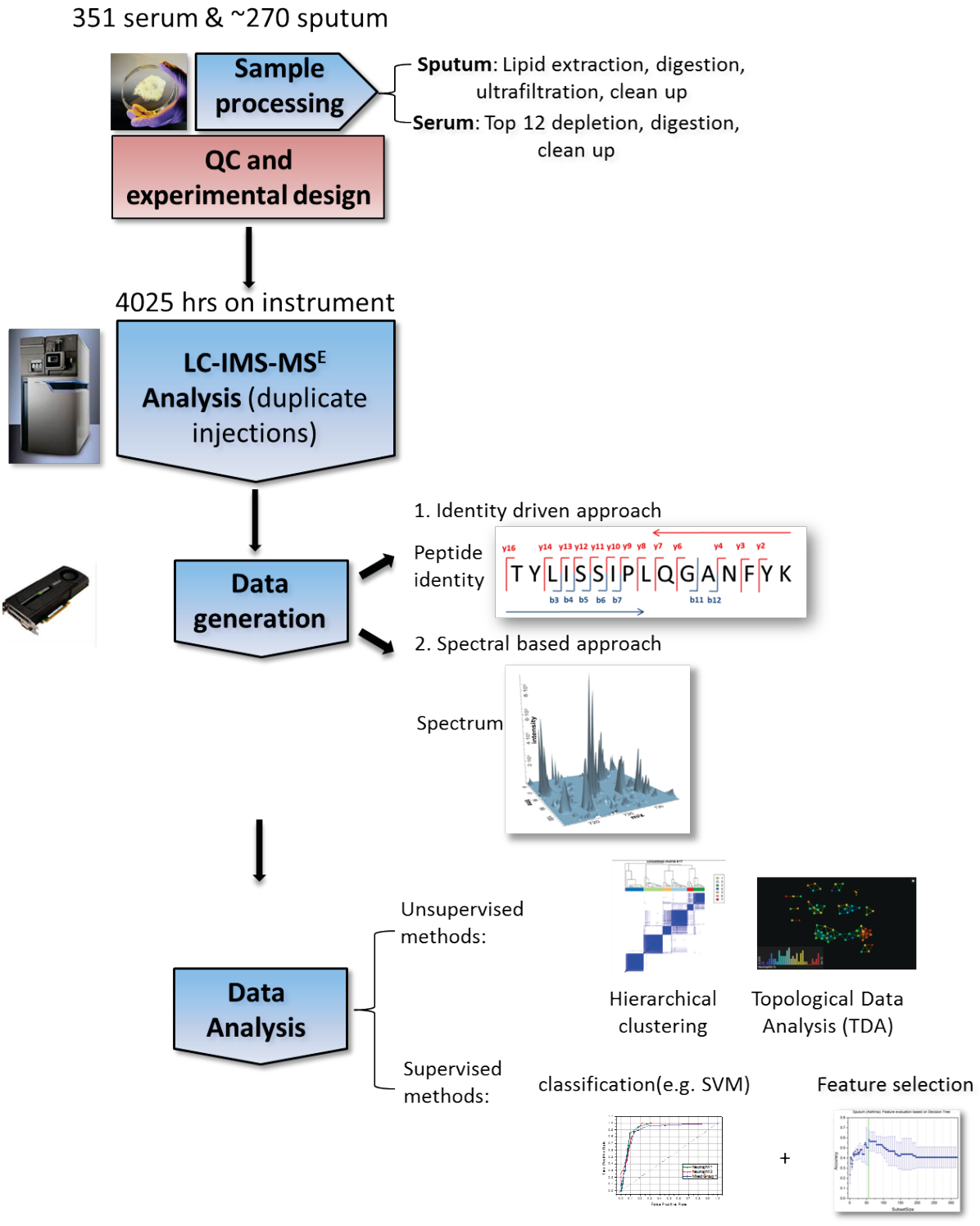


Figure 2-4 The overview of Proteomics analysis in UBIOPRED project

2.3 Moving from molecular level to pathway level

2.3.1 Mapping molecules to pathways

In UBIOPRED, as we have described in Figure 2-3, Omics datasets together with other datasets will be integrated together to generate a handprint for asthma phenotyping. We can find discriminant Omics features among different subgroups of severe asthma. These Omics features can be genes, proteins and other molecules. By mapping them to pathways, we can find which pathways are involved in phenotyping. With the understanding of these pathways, we can explain the mechanisms behind different subgroups. In silico models of biological processes can be constructed using time series measurements from longitudinal study. The whole process is shown in Figure 2-5.

This gives us an example of moving from molecular level analysis to pathway level analysis (Nikolsky et al. 2009). It is quite important to do analysis at a higher level. This is because molecules do not individually complete a biological process. Instead, they work together coherently. Owning to massive interactions among various molecules of different types, we need to study their behaviour and expression at the same time. The most straightforward way to integrate various types of molecular data is mapping molecules to pathways. By finding discriminant pathways, we can understand the underlying biological processes of disease.

2.3.2 Modelling pathways

Having detected disease related pathways, further understanding of the hidden mechanism requires modelling pathways (Wolkenhauer 2014)(Kholodenko 2006). For example, I have been working with clinicians and biologists on constructing an integrated MAPK-38 and GR pathway model to explain corticosteroid resistance of severe asthmatics (Holehouse et al. 2012). Corticosteroids (CS) are essential stress hormones that regulate many physiological processes including immune function and cell proliferation. Because of their anti-inflammatory and immunosuppressive properties, they are also used as treatments for asthma. CSs first bind to their receptor (GR) leading to nuclear translocation and binding of activated GR to DNA leads to regulation of gene expression by transactivation (Ito et al. 2006). Activated GR can also interact with other transcription factors to suppress the proinflammatory cytokine transcription through transrepression. However, in inflammatory diseases such as asthma, the effect of CS as an anti-inflammatory is impaired. Clinicians have demonstrated reduced suppression of pro-inflammatory cytokines by dexamethasone, linked

to augmented activation of p38 MAPK (Bhavsar et al. 2010; Hew et al. 2006). Therefore, it is necessary to study interactions of p38 MAPK pathway with GR-induced signalling pathway.

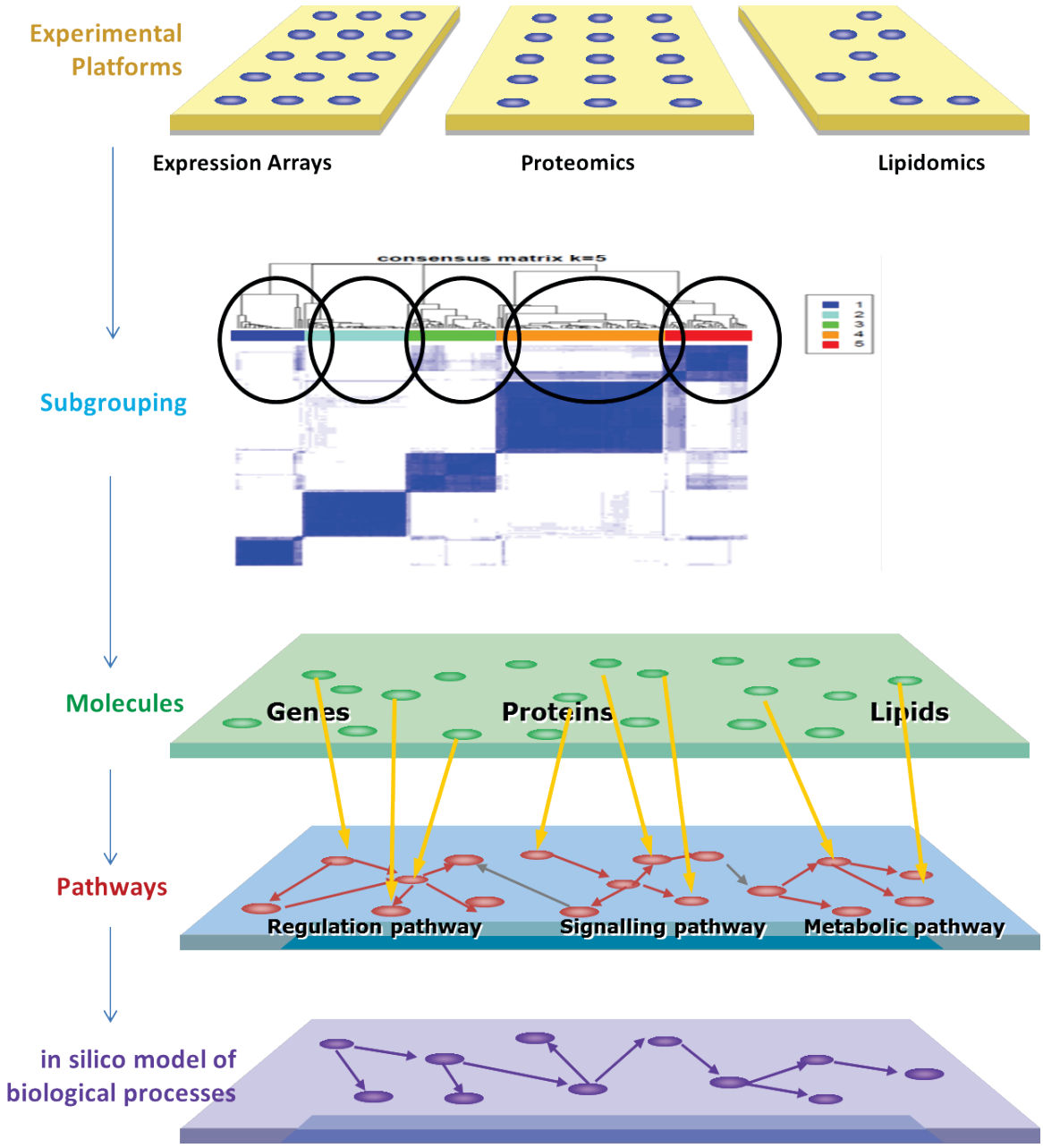


Figure 2-5 Process of obtaining pathway models for understanding sub-phenotypes.

One way to understand the mechanism of corticosteroid responsiveness in inflammatory diseases is to build a mechanistic model. To date, only mechanistic models of isolated pathways have been published whilst a complex model of multiple interconnected pathways has not yet been reported. A computational model of the LPS-induced p38 pathway derived from literature and available online pathway databases has been reported (Hendriks et al. 2008). We use this p38 model to construct a novel mechanistic model of the GR pathway

based on the known biological reactions and proposed an initial crosstalk model between these pathways (see Figure 2-6). A number of positions along the two pathways are taken as nodes where crosstalk could occur and included TGF β kinase-1 (TAK1), MAPK phosphatase-1 (MKP-1) and phosho-p38 itself. However, this crosstalk model is not yet validated by wet-lab experimental observations.

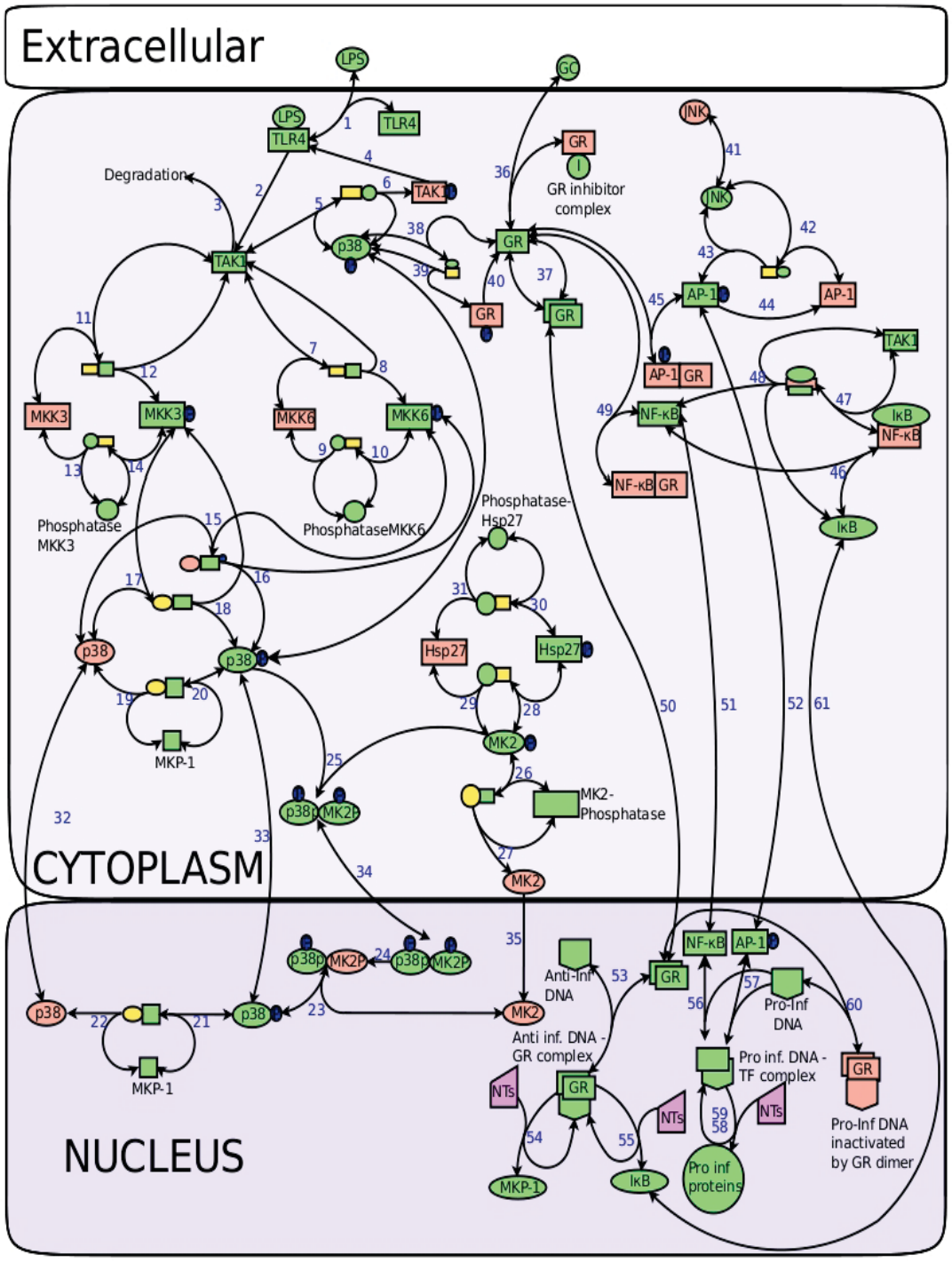


Figure 2-6 The integrated pathway of p38 MAPK and GR from (Holehouse et al. 2012)

The main issue in constructing pathway models is determining a massive amount of model parameters (e.g., kinetic rates and initial concentrations) with limited time series measurements. Like the work in (Hendriks et al. 2008), it is difficult to get a unique solution of parameter set. Although we can simplify the pathway model by reducing the number of elements, the simplified model at a gross granularity will ignore some interactions. Thus, it is better to keep a detailed model and investigate a computational method of parameter estimation. In Chapter 8, we propose an idea of using sparse signal recovery methods, which can potentially contribute to solving this problem.

2.4 Incorporating imaging data

2.4.1 Roles of imaging data in medical practice

Imaging data can also be used to construct a personalized model for disease diagnosis. It can characterize individual structural and functional properties of a tissue. There are various data types, such as magnetic resonance imaging (MRI) (Hollingworth et al. 2000), computed tomography (CT) (Herman 2009), positron emission tomography (PET) (Bailey et al. 2005) and ultra sound. Information contained in imaging data can be important in personalizing patients. This is because imaging data can provide many features of patients, such as anatomy, tissue distributions and material properties (Sensen & Hallgrímsson 2009).

Therefore, predictive models incorporating imaging data have tremendous potential in medical practice (Khan et al. 2005; Killiany et al. 2000). Adequate computational methods are required to help doctors to interpret the imaging data in a way that is consistent with the underlying pathological status. With these new investigation methods, the way of disease treatment will evolve. We can anticipate that in the near future, models based on specific imaging data obtained from individual patients will work together with models of the effects of medical devices and pharmacological therapies, which allows the development of a new discipline of predictive imaging. The integrated model will help clinicians to well interpret patients' states and design necessary interventions in silico. Therefore, patients will receive the 'best available' combination of treatments based on their own conditions.

2.4.2 Using functional MRI in brain disease study

Computational analysis is already playing an important role in providing methods to quantify and fuse together data produced using different imaging techniques. Here we focus on introducing image-based methods for brain activity investigation in diseases of brain disorders (Alivisatos et al. 2012).

There are mainly two branches of methods to enable the mapping activities of human brain: one is mapping the underlying brain electrical activity; the other one is mapping local physiological or metabolic consequence of brain electrical activity. Among methods in the former branch, functional magnetic resonance imaging is a popular non-invasive technique. Based on the fact that brain area consumes more oxygen when it becomes more active, fMRI detects changes of brain area by finding the changes in blood oxygenation and flow (Matthews & Jezzard 2004). Blood oxygenation level dependent (BOLD) fMRI is a powerful method allows localisation to volumes of the order of a few cubic millimetres.

The change in the fMRI signal from neuronal activity is called the hemodynamic response (HRF). The HRF lags stimuli by 3 to 4 seconds and lasts for approximately 10-15 seconds (Strother 2006). To investigate the relationship between brain regions and external stimuli, two types of experimental designs are generally considered: block design (Shan et al. 2014) and even-related design (Abdulrahman & Henson 2016). The main difference between these two designs is the way of presenting stimulus. In the block design, a stimulus for an experimental condition is continuously presented over an extended period of time relative to the hemodynamic response function (HRF) (typically 20-30s). Event-related design, on the other hand, presents discrete stimulus events with durations and inter-stimulus intervals typically short compared to HRF.

As the BOLD signals of activation are relatively weak, noises during data acquisition (e.g., variance from movement, subject, scanner artefacts, and other uncontrollables) must be maximally controlled and removed. Therefore, a series of pre-processing steps must be done before carrying out statistical tests, or any other analyses. Some popular software, such as Statistical Parametric Mapping (SPM) (Penny et al. 2006) and FSL (Jenkinson et al. 2012), have been developed to support various fMRI analyses including pre-processing.

Let us take the FSL pre-processing pipeline as an example (Smith et al. 2004). FSL starts with image reconstruction and brain extraction. After the image in proper format has been created and gross artefacts have been removed, the following step is motion correction, which is to remove the influence of head movements. This step is quite important as it is always inevitable to have head motion during experiment. Movement of head results in the movement of neurons under a voxel. A trivial movement that is 1% of voxel size can induce 1% signal change, which may be larger than the number of actual change in voxel activity. Motion correction in FSL is done by applying a rigid-body transform to the volume, by

shifting and rotating the whole volume data to account for motion. The transformed volume is realigned to the reference volume (in FSL it is the middle one) to minimize variance (see Figure 2-7). The next step is slice-timing correction, which is to bring all slices to the same time point reference. This is because almost all fMRI scanning takes different slices in a single brain volume separately at slightly different time points. In FSL, this step is done by introducing an additional regressor based on the temporal derivative of the expected HRF in the general linear model (GLM) (Sladky et al. 2011) (see Figure 2-7). Spatial filtering is the following step aiming at increasing the signal to noise ratio. It basically averages the intensities of neighbouring voxels to produce a smooth spatial map of intensity. This step is done by a 3D convolution with a Gaussian filter. Along with filtering in spatial domain, FSL also do temporal filtering for removing frequencies of no interest from the signal. There are various sources of temporal noise, such as temporal drift from scanner and physiological cycles (cardiac, respiratory). High pass filter lets high frequencies through, while low pass filer removes high frequencies. FSL recommends using high pass filter only, especially for event-related designs. In the meanwhile, FSL uses prewhitening to estimate autocorrelation (the intrinsic smoothness in each voxel's time series (Smith et al. 2007). Unless prewhitening has been done, the time series analysis is inefficient and may be statistically invalid. The next step in FSL is global intensity normalisation to keep overall signal mean constant across sessions. To allow the comparison between individuals, the last step is registration. All functional images are aligned to the structural image of a higher resolution. The overview of preprocessing steps is shown in Figure 2-7.

After fMRI has been preprocessed, various analyses can be carried out to answer different research questions. One dataset I have been working on is taken from a recent study by (Tomassini et al. 2012). This study is based on the hypothesis that functional impairments of brain plasticity in multiple sclerosis patients can be reduced with practice. fMRI is used to find voxels associated with improvements in visuomotor performance. 23 MS patients and 12 healthy volunteers are studied with fMRI during a visuo-motor tracking task performed with the right hand. After 2 weeks of daily practise, the study is repeated. General linear model is used to detect significantly changed voxels between two sessions. The process of this study is depicted in Figure 2-8.

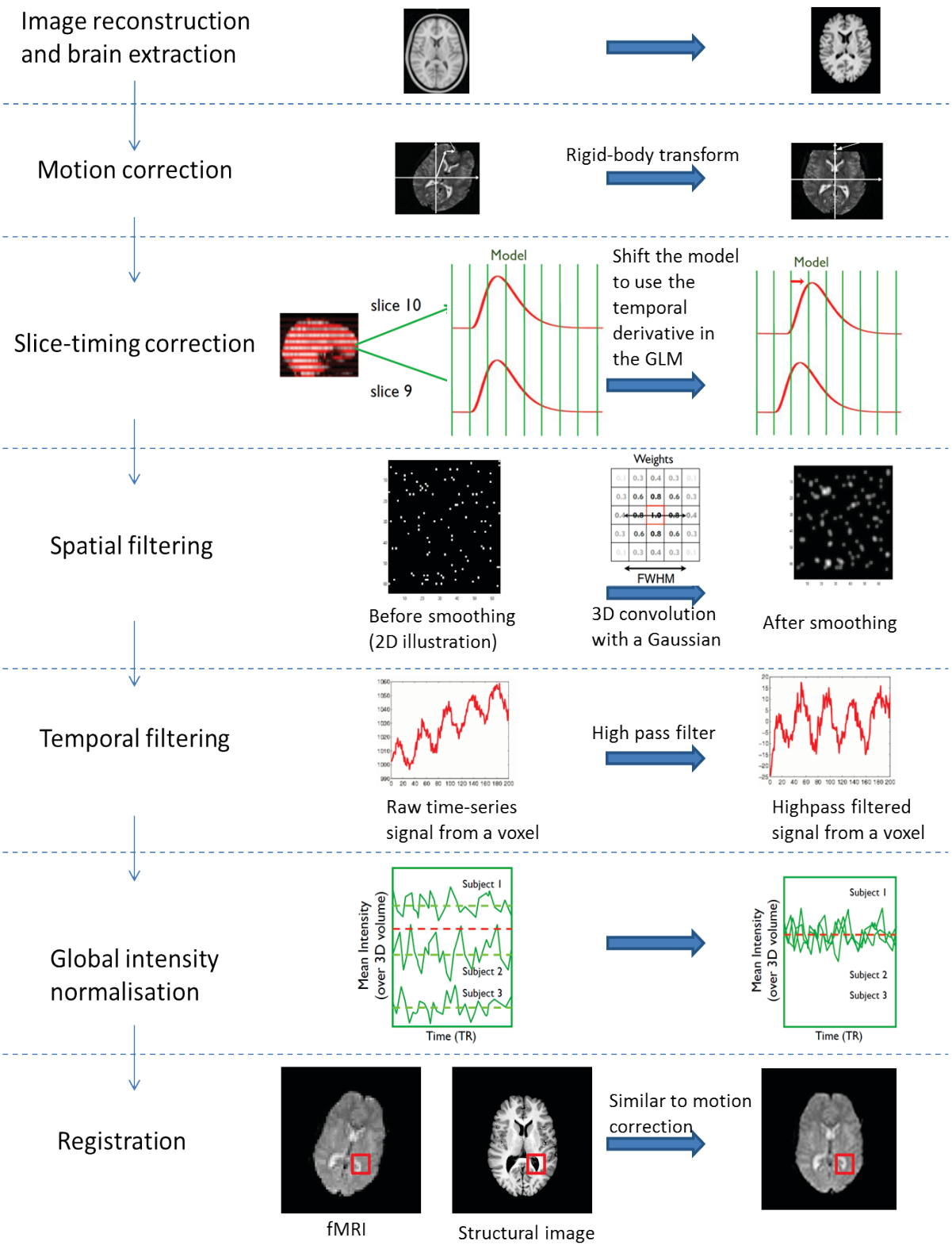


Figure 2-7 The overview of FSL preprocessing steps. Examples of inputs and outputs for 7 main steps are shown, which are image reconstruction & brain extraction, motion correction, slice-timing correction, spatial filtering, temporal filtering, global intensity normalisation and Registration. Some subplots are from http://fsl.fmrib.ox.ac.uk/fslcourse/.

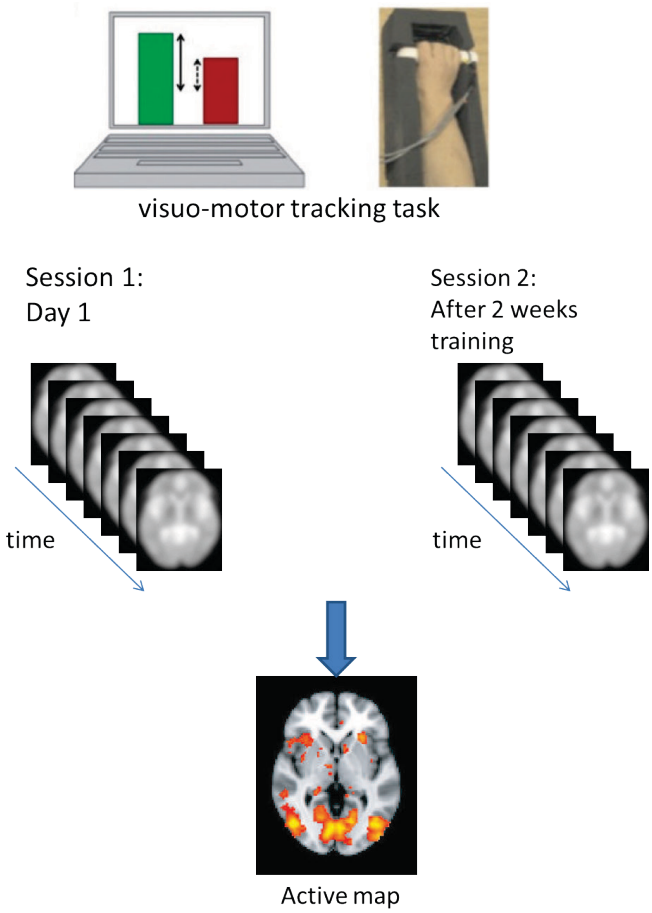


Figure 2-8 The overview of the study in (Tomassini et al. 2012). Participants are asked to track the vertical movements of a computer-controlled bar by altering the height of an adjacent bar with their right hands. The tracking error is used to show the performance. During the task, functional MRI measures brain activity signals. The general linear model method then generates active map, where significantly changed voxels between sessions are highlighted.

The analysis has been done in (Tomassini et al. 2012) is the voxel-wise univariate analysis to relate individual voxels to task. Along with this type of analysis, there are various other types, where Figure 2-9 shows their examples (from (Turk-Browne 2013)). Univariate analysis is a classical and dominant approach to discover discrete clusters of voxels related to the stimuli. In Figure 2-9, an example of univariate analysis is given. The 'Model' shows the pattern of two stimuli introduced to people and 'Measure' tells which part of brain is measured. In this example, the amplitude of BOLD signals of the highlighted brain region evoked by the stimuli is recorded over time. With the application of general linear model, the strength of association of the BOLD signal with two stimuli are estimated and shown in 'Result'. Following this approach, we can find which stimulus can evoke a stronger brain activity of

the investigated region. We have proposed our own analysis method to carry out this analysis in Chapter 4.

Different from univariate analysis where voxels are studied independently, multivariate analysis (MVPA) decodes patterns of activation across voxels relying on machine learning methods (T. Davis et al. 2014). The example shown in Figure 2-9 uses voxels in a specific brain region to construct a classification model. The classifier can predict which stimulus is the introduced. MVPA is of clear value when the overall activation in a region is weak but the pattern over voxels is informative. Chapter 6 introduces our effort on improving the performance of MVPA.

MVPA helps to capture an important characteristic of brain function: brain regions are working together, having local and long-range interactions. To study the interaction between voxels, functional connectivity studies the temporal correlation in BOLD activity between voxels. A common application of functional connectivity is shown in Figure 2-9, which is resting connectivity for examining intrinsic correlations while participants rest. Resting connectivity typically tests the temporal correlation of one or more seed regions with the remainder of the brain during rest. This approach uses selected seeds rather than all voxels to avoid intense computation of full voxelwise correlation matrix. With the increased ability of high-performance computing, full (resting) connectivity can be generated. The analysis of full correlation matrix can give insights into brain networks. By treating each voxel as a node and all correlations between two nodes above a threshold as edges, we can obtain a network graph. Network measurements, including number of edges for a node, minimum number of edges between nodes and proportion of shortest path passing through a node, can informative characterise a network topology. Chapter 7 introduces the method of efficiently and effectively obtaining full connectivity.

Currently, full connectivity is mainly investigated during rest (e.g., (Nicholson et al. 2015)). This is because without the introduction of stimuli, connectivity is not biased towards activation, placing emphasis on pairwise relationships instead. However, if our goal is to understand cognitive processes in the brain, we need to study connectivity during tasks (e.g., (Poppe et al. 2015)). In Figure 2-9, an illustration of task-based connectivity is shown. It examines how these correlations differ by two different cognitive states. Task-based connectivity is quite important to understand how brain systems interact with others. For

example, when we make a decision between options, interactions between the striatum and hippocampus can help to create preference (Wimmer & Shohamy 2012).

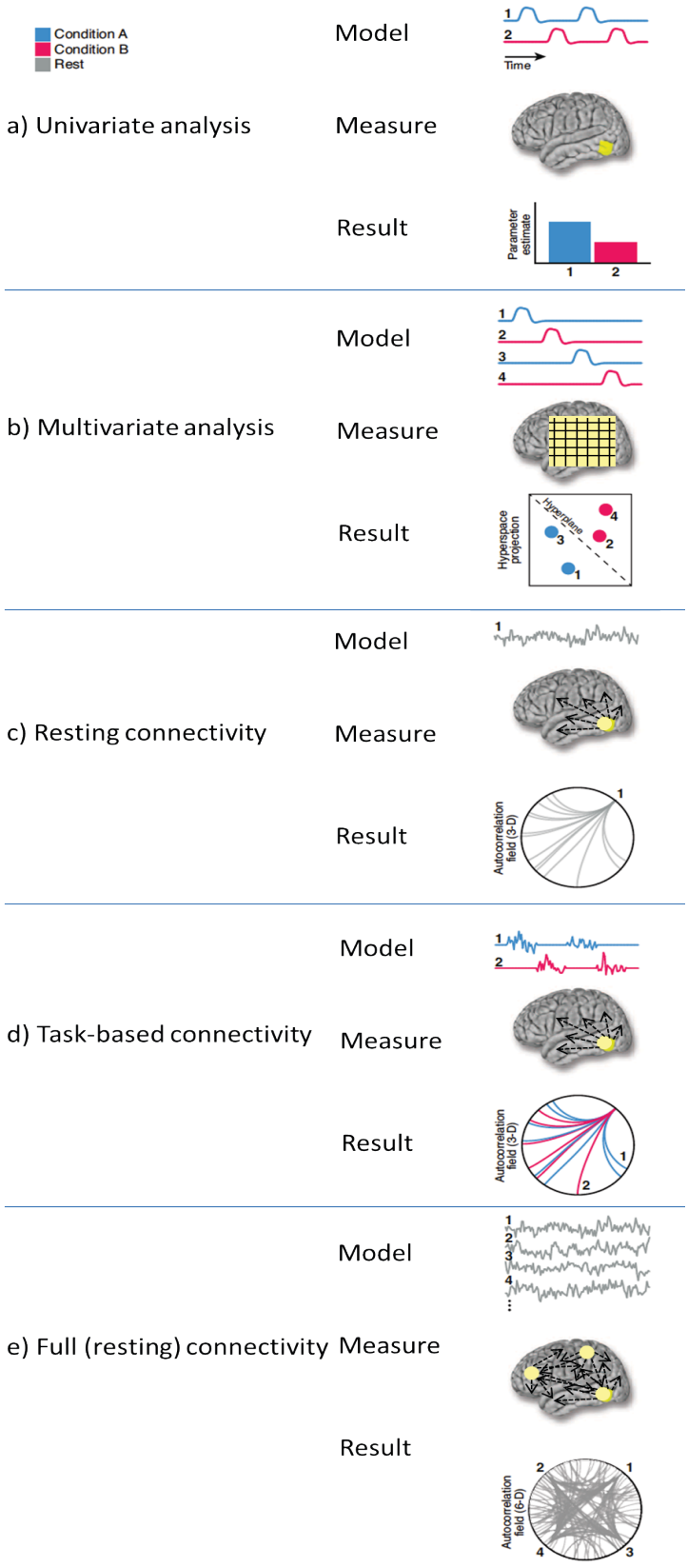


Figure 2-9 Examples of various fMRI analyses (from (Turk-Browne 2013)). In all subplots, 'Measure' tells which brain regions are measured; 'Results' gives examples of analysis results. a) univariate analysis is to detect whether individual voxels are related to experimental conditions, where 'Model' shows the patterns of two conditions; b) multivariate analysis trains classifiers based on the patterns of activations across voxels, where 'Model' shows four events from two conditions that are introduced over time and 'Results' shows the results of classification for finding the type of events; c) resting connectivity is to detect the temporal correlation of one or more seeds with the remainder of the brain during rest, where 'Model' shows the time series BOLD signal of the seed at rest; d) task-based connectivity check how correlations differ under different conditions, where 'Model' shows the time series BOLD signals of the seed under two conditions; e) full connectivity examines all pairwise correlations mostly at rest, where 'Model' shows BOLD signals of voxels at rest.

Functional connectivity can answer the question: what are the interactions between the elements of a given neuronal system. It detects the temporal correlation between spatially remote neurophysiological events. Effective connectivity can also answer the above question in a different aspect (Friston et al. 2013). The influence that the elements of a neuronal system exert over another is modelled in the study of effective connectivity. The most popular method for inferring a reasonably realistic neuronal system model is dynamic causal modelling (DCM) (Friston et al. 2003). DCM consists of two distinct levels: hidden level represents a simple model of neural dynamics in a system of coupled brain regions; observable level describes changes of neural states using HRF. The conceptual overview of DCM is shown in Figure 2-10 (from (Friston et al. 2003) and (Ashburner et al. 2013)). Figure 2-10 shows that the neural dynamics are modulated by the external inputs in two ways: 1) the input directly exerts its effects through direct synaptic responses in the target area (e.g., c_1); 2) the input changes synaptic responses in the target area responses in response to inputs from another area (e.g., b_{23}). Integrating the state equation returns predicted neural dynamics $\boldsymbol{z}$. The dynamics $\boldsymbol{z}$ are passed to the haemodynamic model λ to generate simulated BOLD signal $\boldsymbol{y}$. The parameters in the bilinear state equations and haemodynamic model are adjusted to minimize the difference between predicted and measured signals. The estimated parameters in the state equation indicate the effective connectivity.

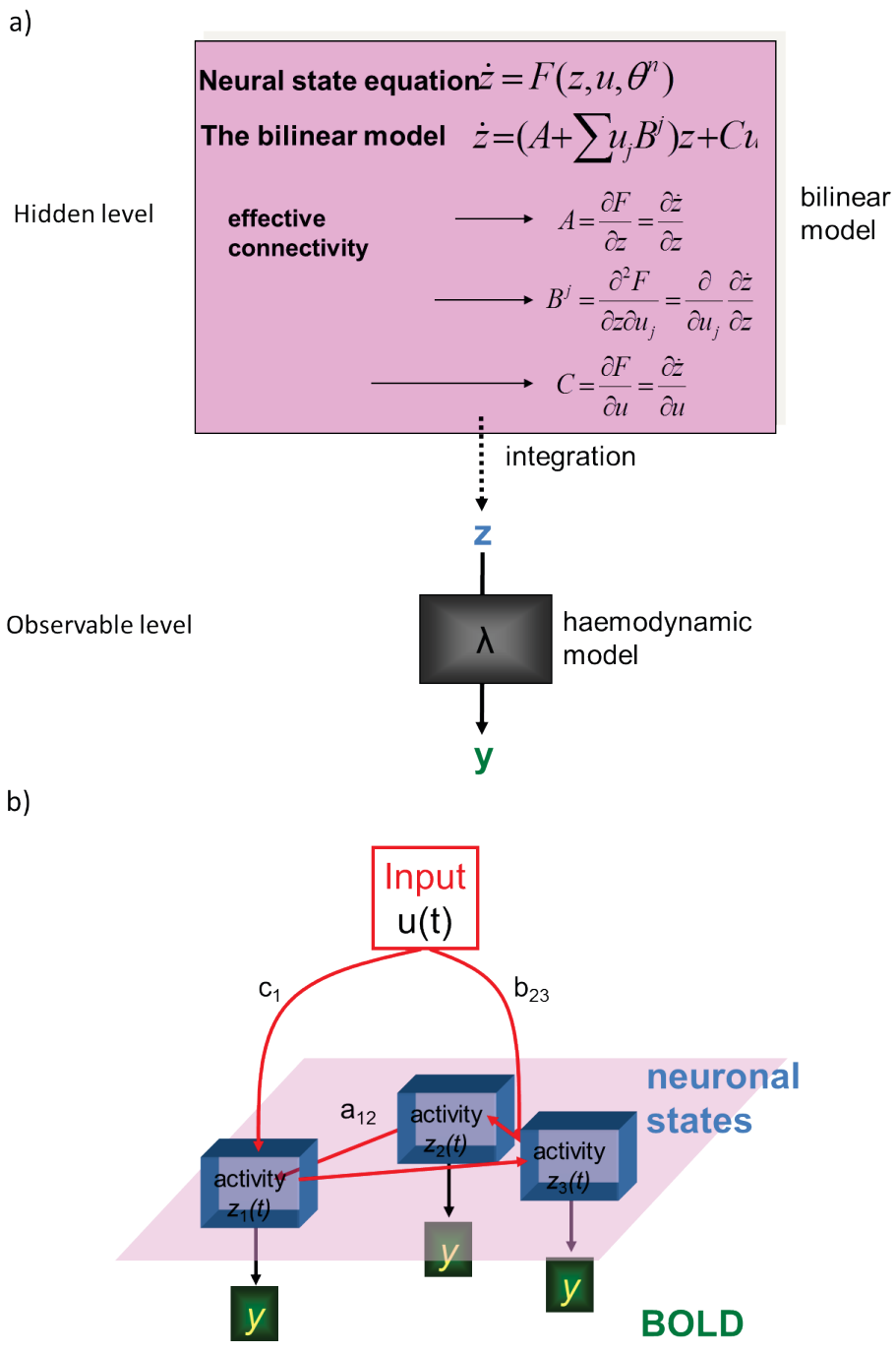


Figure 2-10 Schematic summary of the conceptual basis of DCM (from (Friston et al. 2003) and (Ashburner et al. 2013)). a) Hidden level uses bilinear state equation (pink box) to model a neural system. The state dynamics are passed to the haemodynamic model to generate stimulated BOLD signal. The graphic illustration of a) is shown in b).

2.5 Similarities of computational methods for different data types

Although the whole analysis pipeline varies for different data types, some steps in the pipelines are similar. For example, we find that there are huge similarities in analysing

molecular and fMRI data. Figure 2-11 shows that some preprocessing steps in Proteomics/Lipidomics MS and fMRI analyses are for similar purposes. For example, the starting point of MS data preprocessing is background subtraction (Herbert & Johnstone 2010). Spectra have elevated baseline, resulting from background noises. A curve fitting method, for instance, can subtract background by detecting local minima and regressing out the background noises by fitting a curve to minima (Ruckstuhl et al. 2001). In fMRI analysis, the raw images contain voxels outside brains. These voxels for non-brain tissues can be eliminated that have highly variable contrast and geometry (e.g., scalp and marrow) (Smith 2004). The background subtraction and brain extraction steps are all for removing interference of noises.

In Figure 2-11, we also compare MS data matching with motion correction & registration. In MS analysis, to enable comparison among samples, matching algorithm is developed to group similar ions across spectra using their measurements in mass to charge ratio, drift time and retention time. In fMRI analysis, motion correction and registration transforms images to a reference, making the later analysis feasible (Jenkinson et al. 2002). Although the computational methods behind these steps are different, their purposes are all for matching data across samples/time points.

Locally weighted scatterplot smoothing is frequently used in MS analysis to remove local variations caused by noises (Veselkov et al. 2011). The result of smoothing is that neighbouring ions in spectra has similar intensity. In fMRI analysis, spatial smoothing is carried out to make locally neighbouring voxels have similar strength of signals (Molloy et al. 2014). The signal to noise ratio can be increased by spatial smoothing. 3D convolution with a Gaussian function is the most common method. The purposes of these two steps are apparently the same.

The last step of preprocessing in both MS and fMRI is normalisation. The mean centring method can be used to keep overall signal mean constant across samples.

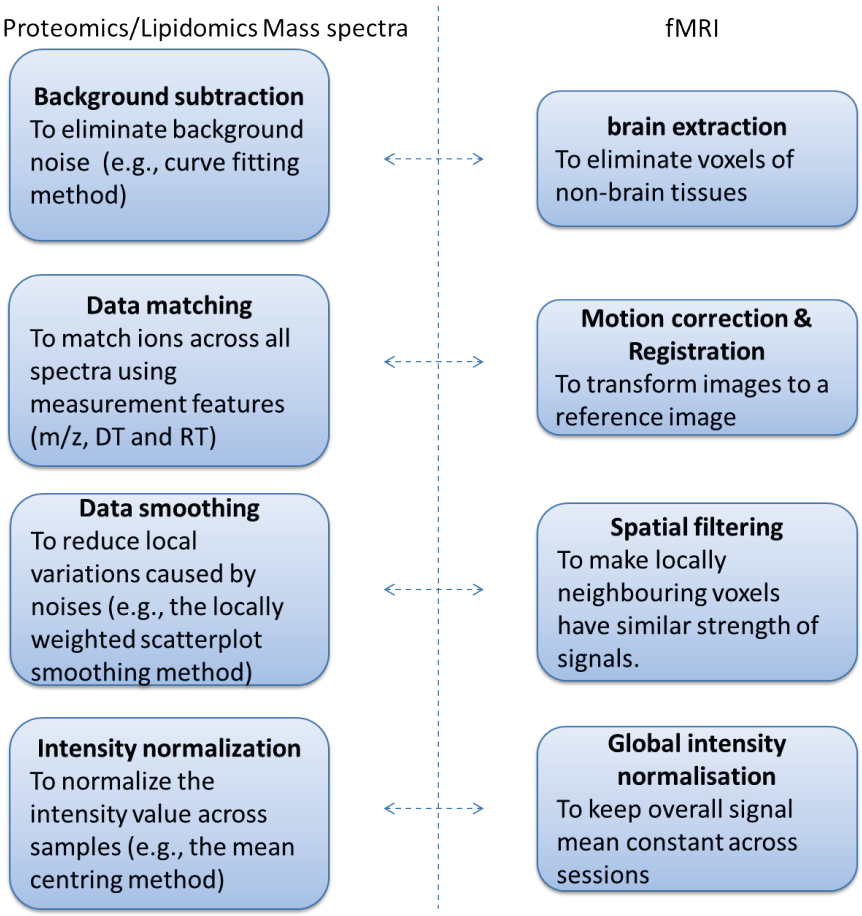


Figure 2-11 Data preprocessing steps in Proteomics/Lipidomics Mass spectrometry and fMRI for similar purposes

We also compare some analysis methods for molecular and fMRI data as summarized in Figure 2-12. Analyses of molecular data are put into the scope of Bioinformatics (Luscombe et al. 2001). The most straightforward study in Bioinformatics is selecting features, which can for example discriminate different disease states or associated with a phenotypical variable. Statistical tests and linear modelling (to be discussed in Chapter 4) can be used. A similar study in fMRI is univariate analysis, where general linear model along with statistical test can be used to detect voxels associated with stimuli. These two studies are answering similar questions. The methods can be basically shared by these two studies. In Chapter 4, we apply linear modelling method to univariate analysis in fMRI.

As we have discussed, MVPA in fMRI applies various machine learning methods to build predictive model for distinct stimuli. Construction of classifier is also a hot topic in Bioinformatics. Machine learning methods can be used to predict different disease states

using molecular data in Bioinformatics or voxels in fMRI. Chapter 6 applies the method which originally used in Bioinformatics to do MVPA.

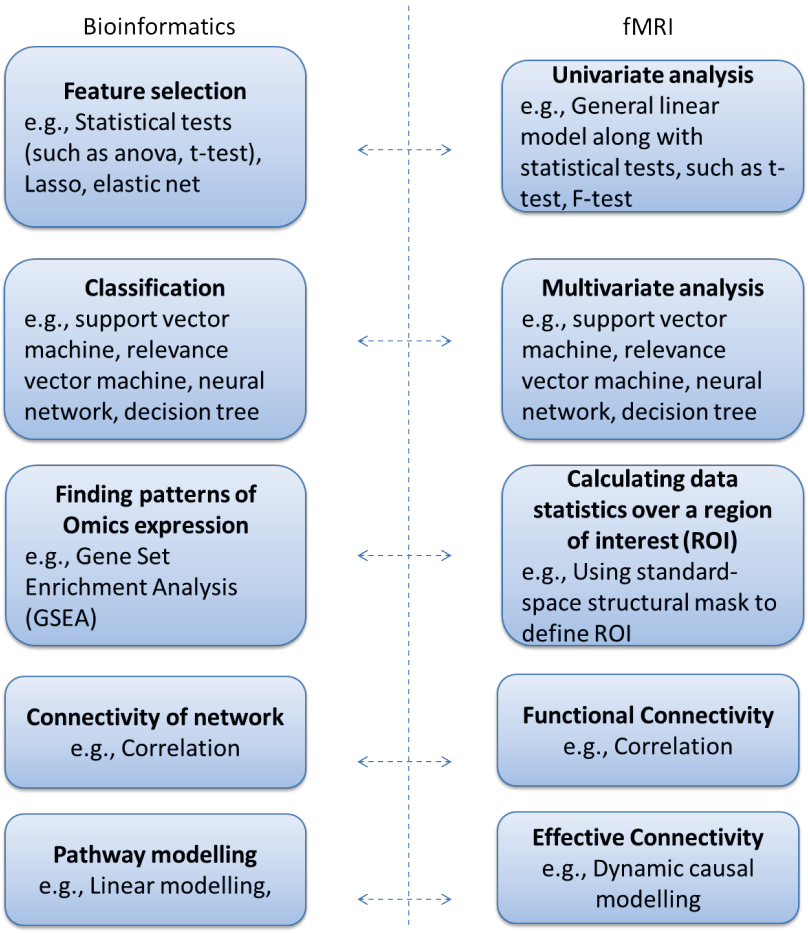


Figure 2-12 Comparison of some analysis methods in Bioinformatics and fMRI analysis.

If we are not only interested in finding features at voxel level in fMRI, we can use standard-space structural mask to calculate statistics over a region of interest (ROI). Similarly, in Bioinformatics, after features have been selected, we can further investigate which pathways contain these features (Kaever et al. 2014). For both studies, we can understand features at a high level, helping to interpret findings.

As we have discussed above, functional connectivity detects the temporal correlation between brain regions. Correlation, mutual information, transfer entropy and Granger causality can be used to find the dependence (Friston 2011). These methods can also be used in Bioinformatics. For example, with the time-series measurements of proteins in a pathway, correlation method can generate the connectivity of network to show the dependence of proteins. Chapter 7 introduces our work in functional connectivity study.

Pathway modelling in Bioinformatics can explain biological processes at systematic level; while the effective connectivity study can use DCM to construct the model of brain region interaction. Both studies model biological/neurological systems using a set of equations with unknown parameters. Thus inference methods are necessary for both studies. In chapter 8, we focus on the way to infer the pathway model.

2.6 Summary of analysis methods

To answer complicated questions, computational approaches are intensively investigated. Figure 2-13 derived from (Wolkenhauer 2014) shows the landscape of analysis methods. It is worth noting that which method should be selected is largely dependent on the type of questions asked and the dataset provided. The most straightforward question is detecting the significance in quantitative experiment. For example, in drug discovery, it would be interesting to find significant differences in molecular profiles (e.g., Proteomics, Lipidomics and Genomics) occurred before and after treatments. In this case, simple significance tests are fair enough to answer this question. If we are also interested in finding covariation of two variables, correlation analysis can be further introduced. However, in most research, statistical analysis is just the first analysis step. It is usually required to do more analyses to get a comprehensive understanding of dataset. Thus, we can move to a higher level of analysis methods, where machine learning methods are applied frequently. For example, clustering method can be used to investigate data pattern and detect subgroups of dataset. To summarize the quantitative dataset, we can fit data into mathematical models to do prediction and classification. However, those mathematical models only explain the dataset; they cannot help us understand complex mechanism behind phenomenon. A lot of work has been done to study inner interactions among components by constructing networks (e.g., protein-protein interaction network, gene regulatory network and functional connectivity). This is inspired by the fact that biological function is the emergent property of multiple collaborated components in living systems. To construct the network, experiments should be able to collect all information about components (such as molecules). Thus a more sophisticated experimented should be designed based on the request of generating a more complicated model. If we intend to explain dynamic features underlying observations, mechanistically modelling methods should be used. Sufficiently rich quantitative time course experiments should be carried out to build those explanatory models. Various parameter estimation methods are invented to meet this requirement.

The following chapter will introduce different analysis methods in answering questions with different level of complexity. We will start with the approach in detecting association and move to dynamic modelling. The investigated methods will answer a large part of questions in medical sciences. It will not be able to cover all possible methods but it will show those methods that I have been encountered during my PhD.

2.7 Conclusion

This chapter discusses about handling medical study in the scope of data science. It starts with giving a vision of what the future medicine look like and suggests the way to achieve it. Rather than introducing standard treatment to all patients, personalized medicine will be used for individuals in the near future. To achieve the personalized medicine, a comprehensive profile of human body in different aspects is needed. Various techniques will be applied to generate high throughput data for quantifying human physiological state. By highlighting the importance of using quantitative data for personalized medicine, we realize the necessity of developing computational methods. Then we discuss methods for processing molecular data and modelling pathways. The analysis of fMRI imaging data is also introduced. Although various data types require different processing pipelines, there are some similarities in individual steps of pipelines. Thus, some computational methods can be shared used for data from diverse sources. In the following chapters, we will introduce some analysis methods that can be potentially used for various data.

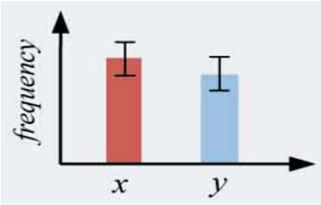
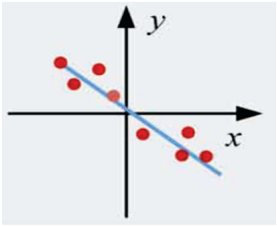
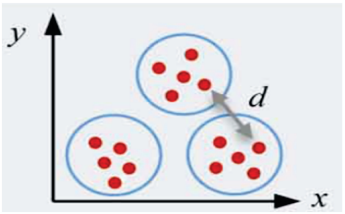
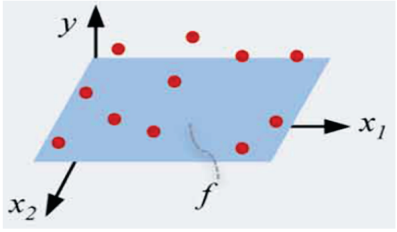
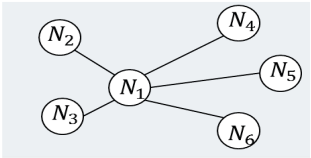
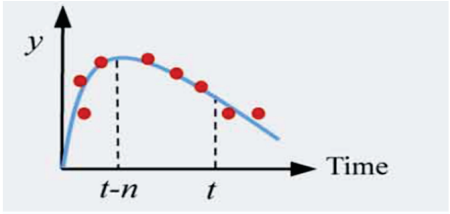
Research type	Example research questions
Finding significance	Is there a difference between two groups 
Analysing dependence	Does a small/large value of x coincide with a small/large value of y 
Identifying groups	Can the data be grouped on the basis of features x and y 
Fitting data	What is the predicted value of y given x_1 and x_2 
Creating connectivity network	Is N_2 connected to N_1 
Building dynamic model	What is the dynamic model explaining the change of y along time 

Figure 2-13 A summary of analysis methods for answering research questions from simple to complex (from (Wolkenhauer 2014)).

3 Questions and methods in datafied life

3.1 Introduction

Two categories of studies that are frequently investigated in datafied life are discussed throughout this thesis. One is investigating relationships between variants and responses, while the other category is exploring the influence of current values of variants on their future values. Suppose $\boldsymbol{Y}$ is the response variable and $\boldsymbol{X}$ contains a number of N explanatory variables. The first category of studies aims at finding the relationship between $\boldsymbol{Y}$ and $\boldsymbol{X}$, which can be presented as

$$\boldsymbol{Y} = f(\boldsymbol{X}) \tag{3.1}$$

where $f(.)$ is the underlying model mapping $\boldsymbol{X}$ to $\boldsymbol{Y}$.

The second category of studies is more interested in the time evolution model of dynamic system. The state changes for discrete time are modelled as:

$$\Delta \boldsymbol{x}_t = h(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \dots, \boldsymbol{x}_{t-\tau}) \tag{3.2}$$

where $h(.)$ is the dynamic model and τ is the maximum lagged value. Similarly, for continuous time the state changes are represented as:

$$\frac{d\boldsymbol{x}_t}{dt} = h(\boldsymbol{x}_{t-T_1}, \boldsymbol{x}_{t-T_2}, \dots, \boldsymbol{x}_{t-T_\tau}) \tag{3.3}$$

where $\{T_1, \dots, T_\tau\}$ are time delays.

In this thesis, three research questions related to the first category are investigated, which are detecting associations, identifying groups and constructing classifiers. Moreover, two questions related to the second category are also investigated, including deriving connectivity and building dynamic models. The following parts of this chapter discuss all these five questions as well as their analysis methods.

3.2 Detecting associations

One often asked research question is which variables are associated with the response. For instance, Genome-wide association study (GWAS) examines which genetic variants are associated with a trait (e.g., (Garcia-Closas et al. n.d.)). Another example is that, in task based fMRI analysis, univariate analysis investigates which voxels are related to the task. The association study is actually selecting which explanatory variables in $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N]$ are related to the response variable $\boldsymbol{Y}$.

3.2.1 1-to-1 association

The nth variable $\boldsymbol{x}_n$, its dependence with the response $\boldsymbol{Y}$ is examined without considering the influence from other variables in $\boldsymbol{X}$. In this way, association detection turns into the problem of finding pairwise dependence.

There are many methods of detecting dependence between two variables, including the Pearson correlation method (Pearson 2006), the Spearman's rank correlation method (Myers et al. 2010) and the mutual information method (Peng et al. 2005). The Pearson's product-moment coefficient between two random variables $\boldsymbol{x}_n$ and $\boldsymbol{Y}$ is:

$$\rho_{\boldsymbol{x}_n,\boldsymbol{Y}} = \frac{E[(\boldsymbol{x}_n - \mu_{\boldsymbol{x}_n})(\boldsymbol{Y} - \mu_{\boldsymbol{Y}})]}{\sigma_{\boldsymbol{x}_n}\sigma_{\boldsymbol{Y}}} \tag{3.4}$$

where $\mu_{\boldsymbol{x}_n}$ and $\mu_{\boldsymbol{Y}}$ are expected means, $\sigma_{\boldsymbol{x}_n}$ and $\sigma_{\boldsymbol{Y}}$ are standard deviations of $\boldsymbol{x}_n$ and $\boldsymbol{Y}$. The Spearman's rank correlation coefficient can be used to do non-parametric measurement of dependence:

$$\rho_{\boldsymbol{x}_n,\boldsymbol{Y}} = 1 - \frac{6\sum d_i^2}{P(P^2-1)} \tag{3.5}$$

where $d_i = x_i - y_i$ is the difference between ranks, P is the sample size, and x_i and y_i are ranks of the ith sample of $\boldsymbol{x}_n$ and $\boldsymbol{Y}$. Unlike the Pearson's product-moment coefficient and Spearman's rank correlation coefficient methods, mutual information is more general and determines the joint distribution of two random variables. There is a recently new method called, maximal information coefficient (MIC) (Reshef et al. 2011), can be used to capture a wide range of dependences both functional or not. The fundamental idea of MIC is that if a relationship exists between two variables $\boldsymbol{x}_n$ and $\boldsymbol{Y}$, then a grid can be drawn on the scatterplot that partitions the data to encapsulate that relationship. By any $n_{\boldsymbol{x}_n}$-by-$n_{\boldsymbol{Y}}$ grid, the MIC value is calculated by finding the largest possible mutual information, say $\max\{I_{n_{\boldsymbol{x}_n}\times n_{\boldsymbol{Y}}}\}$. To achieve a fair comparison between grids of different solution, the mutual information scores are normalized as $\max\left\{I_{n_{\boldsymbol{x}_n}\times n_{\boldsymbol{Y}}}\right\} / \log\min\{n_{\boldsymbol{x}_n}, n_{\boldsymbol{Y}}\}$. The optimal grid can be found by detecting the one with the largest normalized mutual information score. This method is quite powerful to detect relationships from complex dataset independent of their form.

3.2.2 1-to-N association

When variables in $\boldsymbol{X}$ are related, treating them independently and detecting each of their association with $\boldsymbol{Y}$ ignores the effects of interactions among variables in $\boldsymbol{X}$. Multiple variables may be jointly associated with the response. When we want to detect the association between the nth variable $\boldsymbol{x}_n$ and $\boldsymbol{Y}$, it is necessary to consider the effects of other variables. Therefore, instead of detecting association in the 1-to-1 manner, many methods follow the 1-to-N manner. That is, coherently detecting association between the response variable $\boldsymbol{Y}$ and all explanatory variables in $\boldsymbol{X}$.

A popular branch of the 1-to-N association detection approaches is constructing a linear model to model the linear relationship between the response variable $\boldsymbol{Y}$ and explanatory variables $\boldsymbol{X}$. The model is written as (Freedman 2005)

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.6}$$

where $\boldsymbol{Y}$ is a $P \times 1$ vector of observable random variables, $\boldsymbol{X}$ is a $P \times N$ design matrix (i.e. matrix of explanatory variables), $\boldsymbol{\beta}$ is an $N \times 1$ vector containing parameters to be estimated and $\boldsymbol{\epsilon}$ is an $N \times 1$ vector containing noises with the assumption that $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, where σ^2 is the variance of noise.

3.2.2.1 Ordinary least squares regression

An exemplar regression based method to estimate $\boldsymbol{\beta}$ is ordinary least squares regression (OLS), which aims at minimizing the difference between observations and predictions. This conventional approach conditions on$P >$ N. Let $\widehat{\boldsymbol{\beta}}$ denotes the estimated $\boldsymbol{\beta}$, which can be obtained by minimizing the objective function:

$$\boldsymbol{J} = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{3.7}$$

By equating the first derivative of $\boldsymbol{J}$ with respect to $\boldsymbol{\beta}$ to be zero, we can get $\widehat{\boldsymbol{\beta}}$ as:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \tag{3.8}$$

From Equation 3.8 and 3.6, we can get the expectation of $\widehat{\boldsymbol{\beta}}$ as

$$\begin{aligned} \mathbb{E}(\widehat{\boldsymbol{\beta}}|\boldsymbol{X}) &= \mathbb{E}\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon})\} \\ &= \mathbb{E}\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\} \\ &= \boldsymbol{\beta} + \mathbb{E}\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\} \end{aligned} \tag{3.9}$$

As $\boldsymbol{X}$ and $\boldsymbol{\epsilon}$ are independent, $\mathbb{E}\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\}$ is zero. We can get

$$\mathbb{E}(\widehat{\boldsymbol{\beta}}|\boldsymbol{X}) = \boldsymbol{\beta} \tag{3.10}$$

The covariance of $\widehat{\boldsymbol{\beta}}$ can be obtained from

$$\begin{aligned} cov\big(\widehat{\boldsymbol{\beta}}\big|\boldsymbol{X}\big) &= cov\{(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\epsilon})\} \\ &= cov\{\boldsymbol{\beta}+(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\} \\ &= \mathbb{E}\big\{[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}][(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}]'\big\} \\ &= \mathbb{E}\big\{[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\epsilon}\boldsymbol{\epsilon}'\,\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}]'\big\} \\ &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'cov(\boldsymbol{\epsilon})\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\ &= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\ &= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \end{aligned} \tag{3.11}$$

Usually, σ^2 is unknown and is estimated from

$$\hat{\sigma}^2 = \frac{1}{P-N}\sum_{i=1}^{P} e_i^2 \tag{3.12}$$

where $\boldsymbol{e} = \boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}$.

To find which element in the $N \times 1$ vector $\boldsymbol{\beta}$ is significant, statistic tests, typically t-test, can be used based on OLS. To test whether the nth component β_n is non-zero, t-statistic can be used:

$$t = \left|\frac{\hat{\beta}_n}{\widehat{SE}}\right| \tag{3.13}$$

where $\widehat{SE} = \hat{\sigma}\sqrt{(\boldsymbol{X}'\boldsymbol{X})^{-1}_{nn}}$. When $P \gg N$, t-test becomes z-test, where t-statistic is close to $N(0,1)$. Otherwise, t-statistic follows Student's t-distribution with $P-N$ degrees of freedom:

$$t \sim N(0,1)/\sqrt{d^{-1}\chi_d^2} \tag{3.14}$$

where $d = P - N$ and χ_d^2 follows the chi-squared distribution with d degrees of freedom.

To check whether the last n_0 of $\boldsymbol{\beta}$ are significant, F-test is usually used. Suppose we have a model with $\beta_i = 0$ for all $i = N - n_0 + 1, \dots, N$. Let a $(N-n_0) \times 1$ vector $\hat{\beta}^{(s)}$ be the OLS estimate for this model. The F-statistic is

$$F = \frac{\left(\left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}\right\|^2 - \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(s)}\right\|^2\right)/n_0}{\|\boldsymbol{e}\|^2/(P-N)} \tag{3.15}$$

With independent error, we get

$$\left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}\right\|^2 - \left\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(s)}\right\|^2 \sim \sigma^2\chi^2_{n_0}, \qquad \|\boldsymbol{e}\|^2 \sim \sigma^2\chi^2_d \tag{3.16}$$

where $d = P - N$. F-statistic follows Fisher's F-distribution, with n_0 degrees of freedom in the numerator and $P - N$ in the denominator.

3.2.2.2 A constrained version of OLS: Lasso

As we have introduced, OLS minimizes residual sums of squares (RSS), yielding an unbiased estimator $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$. Although it is simple and unbiased, but it is based on the assumption that $P \geq N$. To meet a satisfactory level of prediction accuracy and stability when $P < N$, a Lasso (least absolute shrinkage and selection operator) estimator is proposed as a constrained version of OLS (Tibshirani 1994; Chen et al. 1998):

$$\widehat{\boldsymbol{\beta}} = \mathbf{argmin}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{3.17}$$

where λ is a tuning parameter and $\|\cdot\|_1$ stands for the vector ℓ_1-norm. The ℓ_1-norm penalty term controls the sparsity of estimation. As λ balances the estimation error and the degree of sparsity, increasing λ makes more and more parameters varnish to zero. The value of λ can be determined by cross-validation. There are a considerable amount of computational methods to solve this optimization problem, such as quadratic programming or convex optimization methods (S. R. Becker et al. 2011; Boyd 2010; S. Becker et al. 2011; Beck & Teboulle 2009; Friedman et al. 2007; King et al. 2010).

One straightforward way to select which variables in $\boldsymbol{X}$ are associated with the response variable $\boldsymbol{Y}$ is directly finding whose corresponding values in $\boldsymbol{\beta}$ is non-zero. The usual constructs like p-values and confidence intervals of Lasso estimator are still being investigated. There are some attempts on p-value estimation, including methods based on resampling (Meinshausen et al. 2008; Wasserman & Roeder 2009; Minnier et al. 2011) and methods of providing statistic of Lasso coefficients (Lockhart et al. 2014).

3.2.2.3 Least Angle Regression

Least Angle Regression method (LARS) proposed in (Efron et al. 2004) sequentially selects which factors should be included in the model. However, instead of giving a vector solution of $\boldsymbol{\beta}$, LARS returns a curve denoting the solution for each value of ℓ_1-norm of the parameter vector. LARS generally works as follows: 1) assign all coefficients to zero at the beginning, and find the factor that is most correlated to $\boldsymbol{Y}$, say $\boldsymbol{x}_{n_1}$; 2) take the largest step in the direction of $\boldsymbol{x}_{n_1}$ until another factor $\boldsymbol{x}_{n_2}$ has as much correlation as $\boldsymbol{x}_{n_1}$ with the current

residual; 3) proceed in a direction equiangular between $\boldsymbol{x}_{n_1}$ and $\boldsymbol{x}_{n_2}$ until a third variable $\boldsymbol{x}_{n_3}$ has as much correlation with the residual; 4) continue until all factors are in the model.

The formal description of LARS algorithm is as follows (Efron et al. 2004). Suppose $\widehat{\boldsymbol{\mu}}_{\mathrm{A}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ is the current LARS estimates and $\hat{\boldsymbol{c}} = \boldsymbol{X}'(\boldsymbol{Y} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}})$ is the correlation between factors and residual. The active set A contains the set of indices n whose $\hat{c}_n$ equals to the largest current correlation $\hat{C} = \max_n\{|\hat{c}_n|\}$. Let s_n denote the sign of $\hat{c}_n$ and compute:

$$\begin{aligned} \boldsymbol{g}_A &= \boldsymbol{X}_A{}'\boldsymbol{X}_A \\ A_A &= (\mathbf{1}_A{}'\boldsymbol{g}_A{}^{-1}\mathbf{1}_A)^{-1/2} \\ \boldsymbol{w}_A &= A_A\boldsymbol{g}_A{}^{-1}\mathbf{1}_A \end{aligned} \tag{3.18}$$

where $\mathbf{1}_A$ is a vector of 1's equals the size of A and $\boldsymbol{X}_A = (\dots s_n\boldsymbol{x}_n \dots)_{n\in\boldsymbol{A}}$.

Then the equiangular vector

$$\boldsymbol{u}_A = \boldsymbol{X}_A\boldsymbol{w}_A \text{ with } \|\boldsymbol{u}_A\|^2 = 1 \tag{3.19}$$

subtends equal angles, less than 90°, with each columns of $\boldsymbol{X}_A$, i.e.

$$\boldsymbol{X}_A{}'\boldsymbol{u}_A = A_A\mathbf{1}_{\boldsymbol{A}} \tag{3.20}$$

Also compute the correlations between each predictor with the equiangular vector $\boldsymbol{u}_A$

$$\boldsymbol{a} = \boldsymbol{X}'\boldsymbol{u}_A \tag{3.21}$$

Then the next step of LARS updates $\widehat{\boldsymbol{\mu}}_{\mathrm{A}}$ by

$$\widehat{\boldsymbol{\mu}}_{\mathrm{A}_+} = \widehat{\boldsymbol{\mu}}_{\mathrm{A}} + \hat{\gamma}\boldsymbol{u}_A \tag{3.22}$$

where the step $\hat{\gamma}$ forward the $\boldsymbol{u}_{\boldsymbol{A}}$ direction is

$$\hat{\gamma} = \min_{n\notin A}{}^{+}\{\frac{\hat{C} - \hat{c}_n}{A_A - a_n}, \frac{\hat{C} + \hat{c}_n}{A_A + a_n}\} \tag{3.23}$$

$\min^+$ is the minimum of positive elements within each choice of n. Equation 3.23 can be interpreted as follows. Let us define

$$\boldsymbol{\mu}(\gamma) = \widehat{\boldsymbol{\mu}}_{\mathrm{A}} + \gamma\boldsymbol{u}_A \tag{3.24}$$

For $\gamma > 0$, the current correlations between the nth predictor with the residual is

$$\begin{aligned} c_n(\gamma) &= \boldsymbol{x}_n'\big(\boldsymbol{Y} - \boldsymbol{\mu}(\gamma)\big) \\ &= \boldsymbol{x}_n'(\boldsymbol{Y} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}} - \gamma\boldsymbol{u}_A) \\ &= \boldsymbol{x}_n'(\boldsymbol{Y} - \widehat{\boldsymbol{\mu}}_{\mathrm{A}}) - \gamma\boldsymbol{x}_n'\boldsymbol{u}_A \end{aligned} \tag{3.25}$$

$$= \hat{c}_n - \gamma a_n$$

For $n \in A$, we get

$$|\hat{c}_n| = \hat{C} \tag{3.26}$$

and

$$a_n = A_A \tag{3.27}$$

Then we get

$$|c_n(\gamma)| = \hat{C} - \gamma A_A \tag{3.28}$$

For $n \notin A$, equating Equation 3.28 with Equation 3.25 gives that

$$\begin{cases} \gamma = \dfrac{\hat{C} - \hat{c}_n}{A_A - a_n} & if\ c_n(\gamma) > 0 \\ \gamma = \dfrac{\hat{C} + \hat{c}_n}{A_A + a_n} & if\ c_n(\gamma) < 0 \end{cases} \tag{3.29}$$

Therefore, $\hat{\gamma}$ is the smallest positive value of γ such that some new index joins the active set.

(Efron et al. 2004) has also shown that LARS is closely related to Lasso. Just with one modification, LARS can give the entire path of Lasso solutions as λ varies from 0 to infinity. The modification is: if the value of a factor crosses zero, then this factor will be removed from the active set of predictors and the equiangular will be recomputed. Different from original LARS process, the active set selected in the modified version is not guaranteed to be monotonically growing.

3.2.2.4 Sparse Bayesian learning

Instead of giving regularization penalty as $\lambda\|\boldsymbol{\beta}\|_1$ in Lasso, the Sparse Bayesian learning method (Tipping 2001) specifies a prior on $\boldsymbol{\beta}$ as:

$$p(\boldsymbol{\beta}|\alpha_1, \dots, \alpha_N) = \prod_{n=1}^{N} (2\pi)^{-\frac{1}{2}} \alpha_n^{\frac{1}{2}} \exp\{-\frac{\alpha_n}{2}\beta_n^2\} \tag{3.30}$$

where each hyperparameter α_n independently controls the variance of each weight β_n. This form of prior tends to reduce non-zero weights. If we give a prior to α_n, that is $p(\alpha_n)$, then we can get the full prior of $\boldsymbol{\beta}$ as:

$$p(\boldsymbol{\beta}) = \prod_{n=1}^{N} p(\beta_n) = \prod_{n=1}^{N} \int p(\beta_n|\alpha_n) p(\alpha_n) \mathrm{d}\alpha_n \tag{3.31}$$

When $p(\alpha_n)$ follows Gamma distribution, $p(\beta_n)$ becomes a Student-t distribution. Thus the prior distribution of $\boldsymbol{\beta}$ has the feature of sparsity. Figure 3-1 from (Tipping 2006) shows examples of different priors of $\boldsymbol{\beta} = \{\beta_1, \beta_2\}$. Figure 3-1 a) shows a Gaussian prior; Figure 3-1 b) shows the marginal prior $p(\beta_1, \beta_2)$ for the single hyperparameter model; Figure 3-1 c) shows the marginal prior $p(\beta_1, \beta_2)$ for multiple independent hyperparameter models defined in Equation 3.30. It is clear that the 3rd distribution is most sparse and has a sharp peak at zero. Following descriptions in (Tipping 2001), we assume uniform hyperpriors for the following introduction.

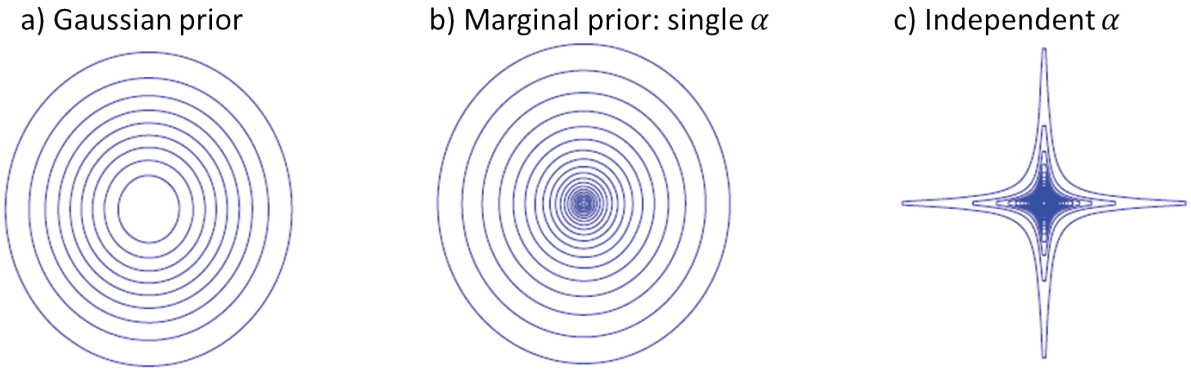


Figure 3-1 Contour plots of different prior distributions over two parameters (Tipping 2006).

The general model for sparse Bayesian learning is shown in (Tipping 2006) as

$$\boldsymbol{Y} = \sum_{n=1}^{N} \beta_n \phi_n(\boldsymbol{X}) + \boldsymbol{\epsilon} \tag{3.32}$$

where the output is the linear weighted sum of N basis functions $\boldsymbol{\Phi} = (\phi_1(\boldsymbol{X}), \phi_2(\boldsymbol{X}), \dots, \phi_N(\boldsymbol{X}))$. The basis functions, for example, can be chosen to be the radial basis function (Sahin & Bay 1998). Equation 3.32 can be easily turned into Equation 3.6 for investigating the association problem by setting $\boldsymbol{\Phi} = \boldsymbol{X}$. For generality, we keep using $\boldsymbol{\Phi}$ for notation.

As the noise is assumed to follow Gaussian distribution $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, the likelihood of $\boldsymbol{Y} = (y_1, y_2, \dots y_P)'$ is of the form:

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-P/2} \exp\left[-\frac{||\boldsymbol{Y} - \boldsymbol{\Phi\beta}||^2}{2\sigma^2}\right] \tag{3.33}$$

As both the likelihood and prior follow Gaussian distribution, the posterior is consequently also Gaussian. Let $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_N\}$. We can obtain the posterior distribution of $\boldsymbol{\beta}$ via Bayes' rule:

$$p(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}|\boldsymbol{\alpha})}{p(\boldsymbol{Y}|\boldsymbol{\alpha}, \sigma^2)} \tag{3.34}$$

in the form of $p(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\alpha}, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}'\mathbf{t} \\ \boldsymbol{\Sigma} &= (\sigma^{-2}\boldsymbol{\Phi}'\boldsymbol{\Phi} + \mathbf{A})^{-1} \end{aligned} \tag{3.35}$$

where $\mathbf{A} = \mathrm{diag}(\alpha_1, \ldots, \alpha_N)$.

The ultimate goal in inference is to estimate the full posterior distribution of $\boldsymbol{\beta}$ over variables, that is $p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$. However, $p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$ cannot be computed analytically (Tipping 2001).There are many ways to approximate this distribution such as Type-II maximum likelihood and Variational techniques (Bishop & Tipping 2000). Here we, introduce the Type-II maximum likelihood method.

Following the product rule of probability, $p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$ can be represented as:

$$p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y}) = p(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y}) \tag{3.36}$$

where the first term can be inferred by Equation 3.35 and the second term is approximated in the following way. According to Bayes' rule, we get

$$p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)}{p(\boldsymbol{Y})} \tag{3.37}$$

Here, $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$ is crudely approximated by a δ-function, $\delta(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP})$, where $\boldsymbol{\alpha}_{MP}$ and σ^2_{MP} are the most probable values that maximise $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$. The full posterior distribution of $\boldsymbol{\beta}$ can be therefore obtained by

$$\begin{aligned} p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y}) &\approx p(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\alpha}, \sigma^2)\delta(\boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) \\ &= p(\boldsymbol{\beta}|\boldsymbol{Y}, \boldsymbol{\alpha}_{MP}, \sigma^2_{MP}) \end{aligned} \tag{3.38}$$

For the case of uniform hyperpriors, maximising $p(\boldsymbol{\alpha}, \sigma^2|\boldsymbol{Y})$ with respecting to $\boldsymbol{\alpha}$ and σ^2 is equivalent to maximising $p(\boldsymbol{Y}|\boldsymbol{\alpha}, \sigma^2)$. The marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{\alpha}, \sigma^2)$ can be represented by an integration over $\boldsymbol{\beta}$:

$$p(\boldsymbol{Y}|\boldsymbol{\alpha},\sigma^2) = \int p(\boldsymbol{Y}|\boldsymbol{\beta},\sigma^2)p(\boldsymbol{\beta}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\beta}$$
$$= (2\pi)^{-P/2}|\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}'|^{-1/2}\exp\{-\frac{1}{2}\boldsymbol{Y}'(\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}')^{-1}\boldsymbol{Y}\} \quad 3.39$$

To obtain α_{MP} and σ^2_{MP} that maximise $p(\boldsymbol{Y}|\boldsymbol{\alpha},\sigma^2)$, we differentiate $\log(p(\boldsymbol{Y}|\boldsymbol{\alpha},\sigma^2))$ with respect to $\boldsymbol{\alpha}$ and σ^2. The iterative re-estimation formulae are:

$$\alpha_n^{new} = \frac{\gamma_n}{\mu_n^2} \quad 3.40$$

and

$$(\sigma^2)^{new} = \frac{\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{N - \sum_{n=1}^{N}\gamma_n} \quad 3.41$$

where $\gamma_n = 1 - \alpha_n\boldsymbol{\Sigma}_{n,n}$ and $\boldsymbol{\Sigma}_{n,n}$ is the element in the n^{th} row and n^{th} column of the covariance matrix $\boldsymbol{\Sigma}$ (MacKay 1992).

The iterative process for inference is:

1. Set initial values to all α_n and σ^2
2. Compute $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using Equation 3.35
3. Computer all γ_n and then re-estimate α_n using Equation 3.40
4. Compute σ^2 using Equation 3.41
5. Go back to step 2 until Convergence

3.2.3 Our interest: M-to-N association

Up till now, we have discussed the association problem for the case of only one response variable. A more general case is that multiple response variables are being investigated together. Thus, $\boldsymbol{Y}$ is in the dimension of $P \times M$, where M is the number of response variables. The parameter matrix $\boldsymbol{\beta}$ is an $N \times M$ matrix, showing the association of the corresponding explanatory variables and response variables. In this case, association is in the M-to-N mode. The M-to-N association is investigated in many studies. For instance, in GWAS, $\boldsymbol{Y}$ can be expression levels of M genes and $\boldsymbol{X}$ contains N genetic variants. Some extra constraints are added, which can, for example, represent the correlation of responses (Curtis et al. 2012).

In chapter 4, we investigate M-to-N association problem in fMRI analysis. An extension of Lasso is used to detect active voxels associated with tasks. Smooth constraint, which forces

spatially neighbouring voxels having similar values in $\boldsymbol{\beta}$, is introduced in the association model.

3.3 Identifying groups

To fully understand the pattern of large scale quantitative data it is also necessary to group similar samples that data points in a group are more similar to each other than those in other groups. The grouping analysis is of significance in datafied life research. For example, in asthma phenotype identification, various grouping methods such as hierarchical clustering (Hastie & Tibshirani 2009) and k-means clustering (Hartigan & Wong 1979) are used to discover novel distinct clinical phenotypes of asthma (Wu et al. 2014; Moore et al. 2010). In breast cancer subtyping study, unsupervised clustering methods are also used to find subgroups of different Omics profiling datasets, such as DNA methylation and mRNA expression datasets (Koboldt et al. 2012).

In grouping analysis, variables in $\boldsymbol{X}$ contains information from individual samples. The output of grouping analysis is the group indicator $\boldsymbol{Y}$ assigned to each sample. The model for mapping $\boldsymbol{X}$ to $\boldsymbol{Y}$ is learned from $\boldsymbol{X}$. Identifying groups is an unsupervised learning approach that only uses unlabelled data. In contrast to grouping analysis, classification, which also uses $\boldsymbol{X}$ to find group indicator $\boldsymbol{Y}$, is a supervised approach. This is because classification uses labelled data to learn the mapping model. We will discuss classification in the next subsection.

Here we generally discuss three ways to identify groups. One is based on clustering methods, such as k-means clustering, hierarchical clustering and mixture models (Jensen 1997). Another one is applying the signal separation techniques, among which principle component analysis (PCA) will be briefly discussed. The third one is topological data analysis (TDA) (Zomorodian 2011).

3.3.1 Clustering analysis

The clustering analysis can fall into two types taking different kinds of input: feature-based clustering and similarity-based clustering (P. Murphy 1991). Feature-based clustering takes feature matrix as the input and is applicable to raw noisy dataset. Commonly, finite mixture models, such as Mixture of Gaussians Model and infinite mixture models, such as Dirichlet process mixture model are used (Bishop 2006). The basic idea of using mixture model is first fitting the mixture model with dataset and then computing the posterior probability of the data point whether it belongs to a cluster. The similarity-based clustering method, on the

other hand, requires distance matrix as the input and facilitates the domain-specific similarity. This is the reason why we choose this kind of method to find similar identities in quantitative Proteomics analysis in Chapter 5. Here, we focus our discussion on similarity-based methods.

The similarity matrix is the measure of distance pairs of data points. Suppose $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two data vectors spanned in R^N space. Some common attribute similarity functions are shown in Table 3-1. Euclidian distance is used quite often and strongly emphasizes on large differences as the distance is squared. The city block distance takes off the square. If $\boldsymbol{x}_i$ is a time series of real valued data, then correlation coefficient is commonly used to measure the similarity. For ordinary variables, such as {low, medium, high}, before the distance calculation, a conversion into real valued numbers (such as 1/3, 2/3 and 1) should be done first. If the $\boldsymbol{x}_i$ contains categorical variable, we can apply Hamming distance. In short, the choice of distance function is basically dependent on the type of dataset and the requirement of similarity detection.

Table 3-1 Different distance functions in similarity matrix calculation

Name	Description
Euclidean distance	$\sum_{n=1}^{N}(x_{i,n}-x_{j,n})^2$
City block distance	$\sum_{n=1}^{N}\lvert x_{i,n}-x_{j,n}\rvert$
Correlation distance	Correlation coefficient between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
Hamming distance	$\sum_{n=1}^{N}\mathrm{I}(x_{i,n}\neq x_{j,n})$

If we would like to explore how clusters are nested inside each other, the hierarchical clustering method can be adopted. This approach includes two ways of clustering: bottom-up (agglomerative) clustering and top-down (divisive) clustering. We focus on discussing the first way where most similar groups are integrated at each step. The algorithm of agglomerative clustering is explained as follows (P. Murphy 1991):

1. Suppose there are P data points. Initialize P clusters $\{C_p\}_{p=1}^{P}$, where C_p contains the pth data point.

2. Initialize a set of clusters available for merging: $S=\{C_1,\ldots,C_P\}$.

3. Pick two most similar clusters to merge: $(i, j) = \arg\min_{C_i, C_j \in S} D_{i,j}$. $D_{i,j}$ is the distance between cluster C_i and C_j.

4. Create a new cluster $C_l = C_i \cup C_j$.

4. Mark C_i and C_j as unavailable: $= S \backslash \{C_i , C_j\}$.

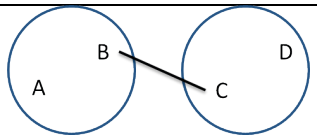
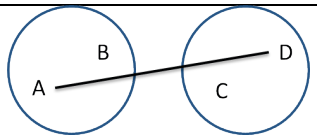
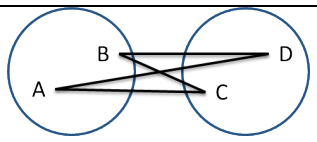
5. Mark C_l as available, $S = S \cup \{C_l\}$.

6. For each $C_p \in S$, update similarity matrix $D_{p,l}$.

7. Go back to step 3 until no more clusters are available for merging.

The distance between clusters is determined by the linkage criteria, where some popular ones are shown in Table 3-2, where a and b represent data points from cluster C_i and C_j and $d_{a,b}$ represents their distance calculated by the methods in Table 3-1. Complete linkage calculates the distance between clusters as the distance between two data points (one in each cluster) that are farthest apart from each other. Single linkage, on the other hand, determine distance simply by the smallest distance between two points (one in each cluster). Average linkage balances single linkage and complete, where the average distance between points is used. It avoids generating large and tight compact clusters.

Table 3-2 Typical linkage functions

Name	Formula	Illustration
Single Linkage	$\min_{a \in C_i, b \in C_j} d_{a,b}$	
Complete Linkage	$\max_{a \in C_i, b \in C_j} d_{a,b}$	
Average Linkage	$\frac{1}{\lvert C_i \rvert \lvert C_j \rvert} \sum_{a \in C_i} \sum_{a \in C_j} d_{a,b}$	

In Bioinformatics, the clustering methods can be used to group similar samples and also similar features. For example, a gene expression dataset collected from multiple patients can be represented by a matrix, in which rows can represent genes and columns represent patients. Clustering by columns (patients) can find groups of patients. If we would also like to find

groups of rows (genes) in the meantime, we can apply biclustering (Madeira & Oliveira 2004). There are many biclustering methods, such as Cheng and Church's algorithm (Cheng & Church 2000), Coupled Two-way clustering (Getz et al. 2000), the iterative signature algorithm (Bergmann et al. 2003) and the SAMBA algorithm (Statistical-Algorithmic Method for Bicluster Analysis) (Tanay et al. 2004). A comprehensive review of those methods can be found in (Tanay 2005). These methods have been widely used in medical research (Oghabian et al. 2014; Cha et al. 2015; Hussain & Ramazan 2016; Williams & Halappanavar 2015).

3.3.2 PCA

Principal component analysis (PCA) is a frequently used signal separation technique to discover potential subgroups of the dataset (Jolliffe & Cadima 2016). It uses an orthogonal transformation to converts observations of correlated variables into linearly uncorrelated ones (i.e. principal components). The largest variance of dataset is captured by the first principal components and the succeeding component has the largest variance possible under the constraint that it is orthogonal to the preceding components.

The transformation helps us to observe clusters of data points. The data points can be shown in a scatterplot, where the coordinates are principal components. PCA is quite useful to provide an intuitive understand of the dataset (e.g., (Desdouits et al. 2015; Alonso-Gutierrez et al. 2015; Zhang et al. 2015)). If we find some clear groups or separations of data points in the scatterplot, we can divide the whole dataset into groups. Then we can investigate whether the data points within a group identified by PCA share some similar characteristic. For example, to detect batch effect, we can use Chi-square test to check whether the separation of dataset observed by PCA is associated with batch factors.

3.3.3 TDA

In medical research, with the emergence of new experimental techniques, large quantitative datasets are generated. For example, millions of ions can be generated by the MS^E machine in Proteomics data analysis. Finding the underlying structure (e.g., subgroups) of high-dimensional dataset is quite challenging. A new approach, topological data analysis (TDA) (Zomorodian 2011), has been proposed to extract the hidden pattern from high throughput dataset (Nicolau et al. 2011). In (Hinks et al. 2015), the full clinical and pathobiological dataset are analysed by TDA to find subgroups of asthmatics. TDA represents high dimensional dataset by a 3-dimensional network. Each node in the network contains patients

similar to each other in multiple dimensions. The presence of a link between two nodes shows that these two nodes share some common patients. TDA does not hard-cluster data points into disjoint groups and provides a convenient way to visualise and explore the whole dataset in low dimensional domain.

The basic idea of TDA is that topological methods work as a geometric approach to shape recognition within data (Lum et al. 2013). This is quite useful, as the pattern of the dataset helps to find meaningful groups. Moreover, the shape of the complex dataset in a multidimensional space drives the analysis by exploring the parallelism of various machine learning methods (Rucco et al. 2014). The three key concepts of TDA are independence of coordinate systems, insensitivity to deformation and compressed representation. The coordinate free means that the topological construction is independent on the coordinate system chosen, but only relies on the distance function. This feature facilitates comparing and integrating various data types collected from different platform. The second characteristic, invariant under small deformation, means that topologically a circle, an eclipse and the boundary of a hexagon are identical because by stretching those all three shapes are loop. This property makes TDA discover data pattern regardless of deformations. The third feature emphasises that TDA gives a compressed representation of the data shape. A shape is identified by a finite combinatorial object. Some information of the pattern maybe lost but the important feature, such as the presence of a loop, will be retained. TDA claims to detect patterns of dataset more comprehensively than clustering and PCA methods, while clustering methods generates unrelated groups and PCA produces unstructured scatterplots.

3.3.4 Our interest: hierarchical clustering for large scale dataset

Hierarchical clustering is widely used in many studies for grouping similar items together. One application is grouping similar chemical identities across different samples which have similar quantitative measurements (e.g., mass to charge ratio). As high throughput machines are increasingly used to generate quantitative dataset, one challenge is how to make clustering work on large scale dataset. Processing large dataset requires large memory and processing power. Especially, hierarchical clustering needs to generate large distance matrixes for large datasets. In chapter 5, we will introduce a binning process to facilitate the application of hierarchical clustering to group similar identities for large scale datasets.

3.4 Constructing classifiers

A way to extract information from quantitative datasets is fitting the data into a mathematical model which has predictive power. For example, we can apply machine learning methods to construct classifiers using the training dataset, which contains observations whose category membership is known. Then the classifier can identify which of a set of categories that a new coming observation falls into. Predictive model can be a useful tool in disease diagnoses (e.g., (J. Davis et al. 2014; Eckardt et al. 2015; Wisittipanit et al. 2015; Möller et al. 2015)).

In grouping analysis the resulting groups are the matter of interest, while the predictive performance of the predictors is of interest in prediction. For example, after we have applied grouping methods to find subgroups of severe asthma using the training dataset, we can then construct a classification model to mathematically summarize the characteristic of each subgroup. When a new sample arrives, we can use the classification model to predict whether it is collected from a healthy people or a patient with a specific subtype of asthma (Park et al. 2015).

As we have mentioned, the quantitative dataset may have large dimension (i.e. large feature space) which is much larger than the number of samples. In some projects, rather than using the whole feature space, some methods such as statistic tests are used in advance to select features. Rather than discussing individual classification methods in detail, we would like to discuss the selection of classification methods and feature set.

3.4.1 Selection of classification methods

Classification aims to learn a function f that maps from input dataset $\boldsymbol{X}$ to output categories $\boldsymbol{Y}$, where elements in $\boldsymbol{Y}$ can be $\{1, \dots, C\}$ with C being the number of categories. When $C = 2$, the classifier is binary; when $C > 2$, we call it multiclass classification (P. Murphy 1991). Classification methods will estimate f using a labelled training dataset (the category value $\boldsymbol{Y}$ for training dataset is known) and use the estimated function $\hat{f}$ to make predictions on the new dataset (the category value for the new dataset is unknown).

There are various classification algorithms to derive $\hat{f}$, including linear classifiers (e.g., Fisher's linear discriminant (Fisher 1936), Logistic regression (Cox 1958), Naïve Bayes classifier (Rish 2001)), Support Vector Machine (Cortes & Vapnik 1995), decision trees (Quinlan 1987), Neural networks (Bishop 1995), Relevance vector machine (Tipping 2001) and deep learning (LeCun et al. 2015). Selecting which classification method to construct the

classifier for the input datasets is a hot topic. It is never easy to say which method can always perform better than the others. Thus, it is worthwhile to try different classification algorithms to see which one has the best performance for your dataset.

The most straightforward way to select which classification method we should choose to construct classifier is exhaustively trying different methods. There are various model selection methods, such as Akaike information criterion (Aho et al. 2014), Bayesian information criterion (Schwarz 1978; Dutta et al. 2015), Bayes factor (Toni & Stumpf 2010; Hug et al. 2015). Among them, cross-validation (CV) is used quite often for selecting classification models. We can evaluate the performance of every method via cross-validation (CV) and check which method returns the best performance (e.g., (Yang, Guo, et al. 2012)).

Cross-validation partitions the training dataset into complementary subsets, performs the analysis on one subset (i.e., training dataset), and validate the analysis on the other subset (i.e. validation set). To avoid bias evaluation, multiple rounds of cross-validation are performed with different ways of partitions. The validation results are averaged over the rounds. There are many types of cross-validation, such as leave-one-out cross-validation (LOOCV) and K-fold cross-validation. The LOOCV removes each sample (the validation set) in turn from the training set and uses the remaining samples as the training set to build the classification model. Then the removed sample is classified by the model. Repeat this procedure multiple times until all samples have been acted as the validation set. In k-fold cross-validation, the whole dataset is randomly partitioned into k equal size subsets. For each run, a single subset is retained as the validation set while the other $k - 1$ subsets are regarded as training set. The process repeats k times (folds) with each subset used exactly once as the validation data.

The common statistical measures of performance are shown in Figure 3-2 (from (Wikipedia n.d.)). Sensitivity measures the percentage of real positives that are correctly identified, where an example of positive samples could be patients. Specificity, on the other hand, measures the ratio of negatives which are correctly found. A negative sample could be a healthy person. To make a balance between sensitivity and specificity, a receiver operating characteristic (ROC) can be used. The ROC curve is a graphical plot of the true positive rate (i.e. sensitivity) vs. false positive rate (1-specificity) created by various threshold settings (Fawcett 2006).

Total population	Condition positive	Condition negative	
Test outcome positive	True positive	False positive (Type I error)	
Test outcome negative	False negative (Type II error)	True negative	
	True positive rate (Sensitivity) $= \frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate $= \frac{\sum \text{Flase positive}}{\sum \text{Condition negative}}$	Accuracy $= \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
	False negative rate $= \frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (Specificity) $= \frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	

Figure 3-2 Summary of statistical measurements of performance (from (Wikipedia n.d.)).

3.4.2 Construction of classifiers

The selection of classification methods are based on their performance estimated via cross-validation. However, the settings of classifier that will ultimately be used to predict new coming samples cannot be fully determined by CV. That is, we can roughly determine which machine learning method to use, but the detailed settings (e.g., parameters) of the classifier using the chosen method is not defined. We can give a quick explanation. Let us take the k-fold CV as an example. In each fold of CV, a classifier is constructed using $k-1$ subsets. Thus k classifiers will be constructed in the whole process. As each classifier takes different dataset as input, they are not identical. Therefore, CV methods cannot give a single classifier that can be directly used for future prediction.

Here, we would like to introduce a pipeline proposed in (Proutski 2010) for the construction of a classifier for the prediction on an independent dataset. As it is shown in Figure 3-3, the optimal settings of classifier are determined via CV. The whole process is as follows:

1. The whole dataset is split into two pieces: a complete training dataset and an independent dataset.
2. Then we apply the cross-validation method. In each CV fold, we proceed as:
    a. For each CV training dataset, we first construct the classifier using all features.
    b. Test and record its performance using area under ROC (AUROC) on the CV testing dataset.

c. Then we rank features by feature selection methods. Features with a higher rank are supposed to have larger influence on the classification performance.
d. We then remove a certain number features and use the remaining features to construct the classifier.
e. Go back to step b until the number of remaining features in the classifier reaches the lowest limit.

3. We will get a plot, like the example shown in Figure 3-3. The plot shows the ranges of AUROC values obtained under different number of features. Select the optimal number of features based on this plot.
4. Afterwards, we use the complete training dataset to construct the classifier with the optimal number of features.
5. Test the performance of the classifier on the independent dataset.

The whole procedure creates a classifier through intensive CV computations. The settings of classifier are determined based on the selection of optimal number of features. This is just an example pipeline to build a single classifier for new predictions. In Chapter 6, we will propose our own way to find a single optimal feature set, which can be used to construct a model for future prediction.

3.4.3 Our interest: the balance of stability and predictive performance of features

Features selected in the classifier are of significance. For example, in Bioinformatics, those features are regarded as biomarkers. It is necessary to discover stable biomarkers that can be consistently identified under small perturbations of the dataset. There are increasing literature discussing the stability of feature selection (Abeel et al. 2009; Zucknick et al. 2008; Ahmed et al. 2011; Alexander & Lange 2011; Rondina et al. 2014). The general way to assess the stability is as follows: 1) subsample the whole datasets; 2) construct classifier for each subset; 3) select features for each classifier; 4) assess the agreement among the resulting sets of selections. As the stability of the feature selection itself is not meaningful, it should be always assessed together with predictive performance.

A method to balance the predictive performance and stability of feature selection has been proposed in (Kirk et al. 2013). We will adopt this method, which is originally developed for biomarker discovery in Bioinformatics, to do robust prediction for fMRI data in Chapter 6. Our aim is using fMRI signals to predict brain activities and in the meanwhile obtaining

which brain regions are related to activities. Our interest is applying an analysis method which has been developed in other disciplines to a new field.

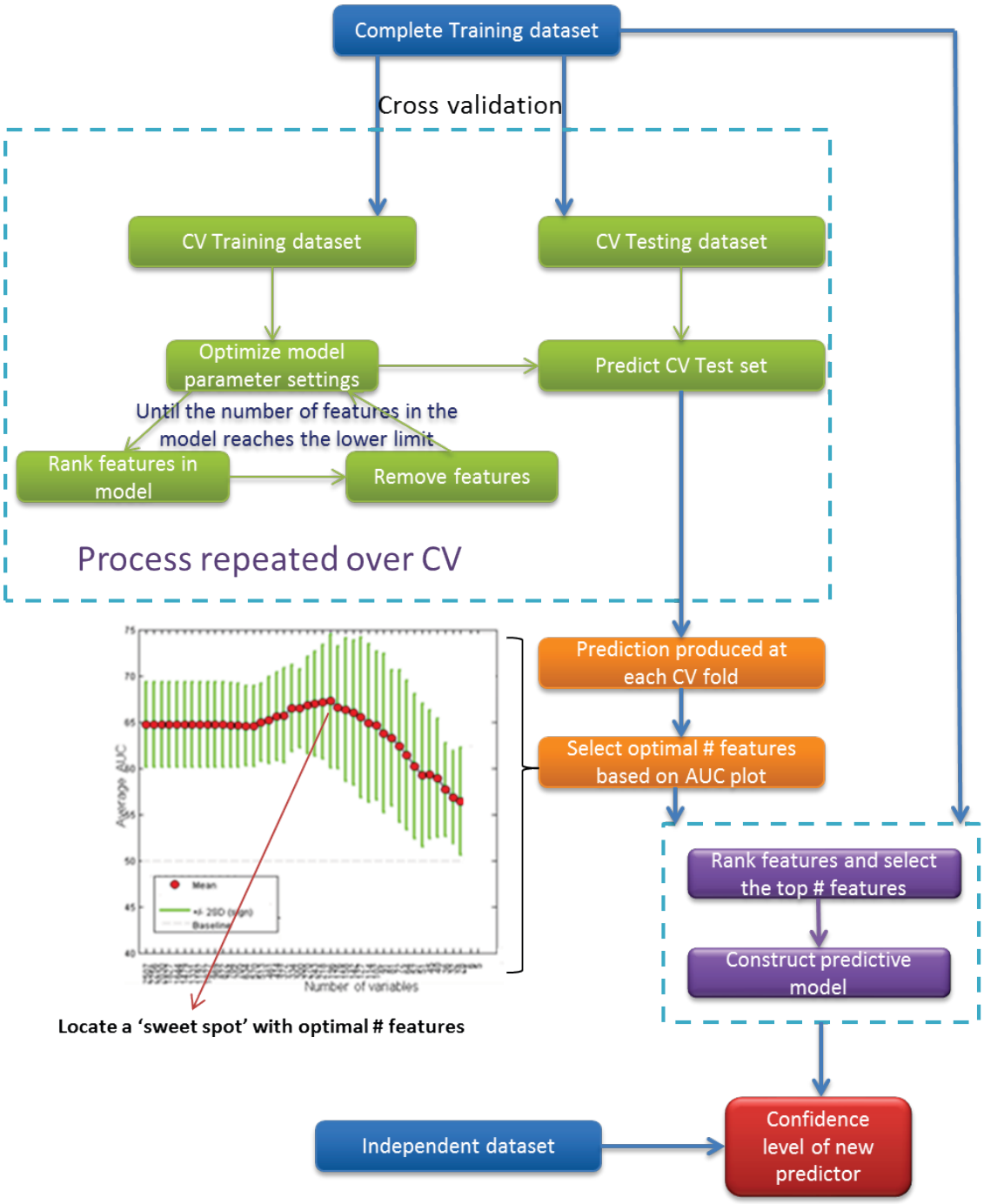


Figure 3-3 The process of constructing a classifier for prediction on an independent dataset via cross validation

3.5 Deriving connectivity

Now let us move on to the second category of questions. We are interested in investigating dynamic systems, whose components are interacting with each other resulting in time

evolution of systems. The previous state of variables, which represents system components, influences the current state. The underlying dynamic function is of our interest to be inferred. In this subsection, we focus on investigating methods to find which components are interacted in a system, while the next subsection will discuss approaches to infer the mathematical models of dynamic systems.

The interaction of components forms networks. Networks, as defined in (Newman 2010), is a collection of nodes joined together in pairs by links. Here, we regard the connectivity as the collection of links. Deriving connectivity is a hot topic in many studies. For example, there are lots of efforts on constructing gene regulatory network using quantitative gene expression data, where its connectivity contains links between nodes showing interactions between genes (Allen et al. 2012; Parikshak et al. 2015; Le Novère 2015; Li et al. 2015). Similarly, functional connectivity is frequently investigated in fMRI analysis to explore relationships among different functional regions of a brain (e.g., (Luo et al. 2014; Chen et al. 2016)). Methods of deriving connectivity falls into two categories: one only focuses on inferring skeleton, while the other category also cares about the direction of link.

3.5.1 Methods of inferring skeleton

Skeleton can be simply inferred by computing pairwise correlation, mutual information, or other similarity metric between each pair of nodes. However, the resulting skeleton contains a massive amount of indirect links owing to transitive effects of correlations (Barzel & Barabási 2013). Therefore, many studies focus on inferring direct links that identify true pairwise interactions (Stephen M. Smith et al. 2013). For example, partial correlation is used to characterize conditional dependences (de la Fuente et al. 2004; Hemelrijk 1990; Veiga et al. 2007; Zuo et al. 2014); network deconvolution methods formulate the inference problem in the framework of graph-theory (Barzel & Barabási 2013; Feizi et al. 2013). Here we focus on discussing the partial correlation method that is closely related to the work in Chapter 7.

Partial correlation is to calculate the dependence between two P-dimensional variables, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, conditioned on a set of M controlling variables $\boldsymbol{Z}$. Then the partial correlation between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, written as $\rho_{\boldsymbol{x}_1,\boldsymbol{x}_2\cdot\boldsymbol{Z}}$, is the correlation between the residuals $R_{\boldsymbol{x}_1}$ and $R_{\boldsymbol{x}_2}$, which are resulted from the linear regression of $\boldsymbol{x}_1$ with $\boldsymbol{Z}$ and $\boldsymbol{x}_2$ with $\boldsymbol{Z}$, respectively. The formula to calculate partial correlation is:

$$\rho_{x_1,x_2\cdot \boldsymbol{Z}} = \frac{E[\left(R_{x_1} - \mu_{R_{x_1}}\right)\left(R_{x_2} - \mu_{R_{x_2}}\right)]}{\sigma_{R_{x_1}}\sigma_{R_{x_2}}} \tag{3.42}$$

The problem of calculating partial correlation can also been handled by estimating the inverse of the covariance matrix. As it has been shown in (Hastie et al. 2009; Koller & Friedmand 2009; Meinshausen & Bühlmann 2006; Peng et al. 2009), off-diagonal elements of the inverse covariance matrix indicates linear dependences of nodes. If an off-diagonal element is zero, the corresponding two nodes are conditionally independent. Suppose we would like to estimate the partial correlation between any two nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of a set of $\boldsymbol{V}$, conditioned on $\boldsymbol{V} \setminus \{\boldsymbol{x}_i, \boldsymbol{x}_j\}$. If the covariance matrix $\boldsymbol{\Sigma}$, with the (i,j)th entry of $\Sigma_{i,j} = cov(\boldsymbol{x}_i, \boldsymbol{x}_j)$, is invertible, we can get $\boldsymbol{\Omega} = (v_{i,j}) = \boldsymbol{\Sigma}^{-1}$. Then we have

$$\rho_{\boldsymbol{x}_i,\boldsymbol{x}_j\cdot \boldsymbol{V}\setminus\{\boldsymbol{x}_i,\boldsymbol{x}_j\}} = -\frac{v_{i,j}}{\sqrt{v_{i,i}v_{j,j}}} \tag{3.43}$$

As the example shown in Figure 3-4, partial correlation can remove some indirect links but sometimes it may also wrongly discover some non-existence links.

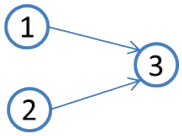
	Network A	Network B
Network structure		
Network dynamic equations	$\begin{cases} x_1(t) = u_1(t) + noise \\ x_2(t) = x_1(t) + noise \\ x_3(t) = x_2(t) + noise \end{cases}$	$\begin{cases} x_1(t) = u_1(t) + noise \\ x_2(t) = u_2(t) + noise \\ x_3(t) = 0.5x_1(t) + 0.5x_2(t) + noise \end{cases}$
Full Correlation	$\begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.1 & 0.7 \\ 0.1 & 1.0 & 0.6 \\ 0.7 & 0.6 & 1.0 \end{bmatrix}$
Partial correlation	$\begin{bmatrix} 1.0 & 0.7 & 0.1 \\ 0.7 & 1.0 & 0.7 \\ 0.1 & 0.7 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \end{bmatrix}$

Figure 3-4 Examples of using partial correlation to infer direct links. $u_1(t)$ and $u_2(t)$ are input signals with the correlation coefficient of -0.035. The noises are randomly sampled from a Gaussian distribution with mean 0 and standard deviation 0.01. Two 3-node networks are used for illustration. The results of partial correlation are compared with the results of full correlation. Full correlation can also be calculated using Equation 3.42 by simply setting

$\boldsymbol{Z} = \emptyset$. For Network A, partial correlation has a better performance than full correlation in indirect link reduction. However, partial correlation cannot successful detect indirect links in Network B.

Moreover, when the dimension of variable is smaller than the number of variables in $\boldsymbol{Z}$, that is $P < M$, the calculation of inverse covariance is quite challenging. Regularization methods, called regularized inverse covariance (ICOV), are then developed to solve this problem by adding sparse constraints (Friedman et al. 2008; Varoquaux et al. 2010):

$$\boldsymbol{\Omega} = \mathbf{argmin}_{\boldsymbol{\Omega}>0}\mathrm{tr}\left(\boldsymbol{\Omega}\boldsymbol{\Sigma}_{\mathrm{sample}}\right) - \log\det\boldsymbol{\Omega} + \lambda\|\boldsymbol{\Omega}\|_1 \qquad 3.44$$

Where $\boldsymbol{\Sigma}_{\mathrm{sample}}$ is the sample covariance matrix, and $\|\cdot\|_1$ is the element-wise ℓ_1 norm of coefficients in the matrix $\boldsymbol{\Omega}$. Optimization methods can be used to solve this equation. ICOV is a parameter dependent methods and the value of the regularization parameter greatly influences the results.

3.5.2 Methods of inferring skeleton and direction

Along with the skeleton, there are many methods inferring the direction of links as well. For example, graphic model based methods detect direct information flow of networks (Weigt et al. 2009; Jordan & Wainwright 2008); the LiNGAM (linear, Non-Gaussian, Acyclic causal Models) (Shimizu 2006) and its variations (Hyvarinen & Smith 2013) estimate connectivity via independent component analysis; Granger causality (Granger 1969) infers dynamic relationships among variables using multivariate vector autoregressive modelling; Patel's conditional dependence (Patel et al. 2006) arrives at a measure of connectivity by simply looking at imbalance between conditional probabilities; generalised synchronisation evaluates synchrony by analysing the independence between signals in a state space reconstructed domain (Dauwels et al. 2010).

The above methods along with the methods mentioned in the previous subsections, including mutual information, partial correlation and ICOV have been discussed and compared in (Smith et al. 2011). The top 3 methods with the best performance found in (Smith et al. 2011) are partial correlation, ICOV and the Bayes net methods. Partial correlation and ICOV have already been discussed in the above subsection. An introduction of Bayes net methods will be given in Chapter 7. Bayes net methods are based on the assumption that the true causal model forms a directed acyclic graph. There are many searching methods for causal graphs available such as SGS (Pearl 2000), PC (Pearl 2000), TPDA (Cheng et al. 2001), MMHC

(Tsamardinos et al. 2006) and TPMB (Wang & Chan 2012). All these methods first infer the skeleton and then infer the direction of links. Our method in Chapter 7 is based on the PC algorithm.

3.5.3 Our interest: a parameter free inference method

Our interest is developing a parameter free method that can accurately infer the skeleton of network. The top 3 connectivity inference methods found in (Smith et al. 2011) are not all parameter free. For example, the performance of ICOV is influenced by the penalty parameter; the PC method based on Bayes net is dependent on the chosen of the significant level. In Chapter 7, we propose an elastic PC method which can approximate to the real skeleton of network given no time constraint. This method is applied to fMRI data to generate functional connectivity. With a time constraint, the elastic method can maximally approximate to the skeleton of network by using the calculations from previous steps.

3.6 Building dynamic models

To explain the time evolution of a system, researchers introduce dynamic models which include a set of equations describing a function of time and the state of the system (Boccara 2010). For instance, dynamic models of biological pathways describe the time evolution of intracellular molecules in response to stimuli and give a detailed insight into pathway dynamics (Hecker et al. 2009). Particularly, in drug discovery for diseases, such as cancer, diabetes and asthma, dynamic models of pathways allow for a better understanding of biological processes of drug effects leading to more efficient treatments (Ribeiro & Pinto 2009). The inference of dynamic models is finding $h(.)$ that describes the time evolution of systems. The following parts first discuss three different classes of inference problem and then introduce various inference methods.

3.6.1 Three main problem classes

As it has been discussed in (Villaverde & Banga 2014), from the identification point of view, the model inference problems can fall into three main classes: full network inference, network selection and parameter estimation .

Full network inference explores a full network that fits into the observed data. In this class, the model structure and parameters are both unknown. Inference methods need to explore the whole spaces to find appropriate model structure (e.g., kinetic laws) and parameters with the given quantitative data (e.g., time concentrations). This class of problems is quite general and challenge.

Network selection targets at finding the most probable model matching to the observed data from a set of candidate models. Given quantitative measurements and an existing dynamic model with possible modifications, this class of problems detect structural modifications and parameters that fit the data. Compared to full network inference, network selection requires more prior knowledge of model structure.

When the model structure has been predefined, the main task to construct model is inferring parameters. Let us take the biological pathway model as an example. A biological pathway system is a collection of interacting molecules (species). For building a reliable model, kinetic parameters are needed. Some parameters, such as kinetic rates and initial concentrations, cannot be measured directly through biological experiments. Moreover, as pathways are just the abstraction of real biological processes, some kinetic rates do not have corresponding biological interpretations. Thus, it is necessary to estimate unknown parameters from the observed experiment measurements. Compared to the other two classes, this class of problems is the least challenge.

3.6.2 Inference methods

Currently, there are a lot of inference methods available, such as traditional optimization methods (Mendes & Kell 1998), alternating regression (Chou et al. 2006), simulating annealing (Gonzalez et al. 2007), multiple shooting (Peifer & Timmer 2007), particle swarm (Schwaab et al. 2008), spline techniques (Zhan & Yeung 2011), recursive estimators (e.g., Kalman filters) (Baker 2010; Simon & Simon 2003; Sun et al. 2008; Wang et al. 2009) and Bayes' inference (Jaakkola & Jordan 2000; Granqvist et al. 2011; Kirk et al. 2008). The study in (Villaverde & Banga 2014) has reviewed different perspectives for inferring dynamic systems, grouping them in eight areas: inverse problems, optimisation, systems and control theory, chemical reaction network theory, Bayesian statistics, physics, information theory and machine learning. The following parts discuss three most predominant perspectives, which are optimisation, systems & control theory and Bayesian statistics.

3.6.2.1 Perspective from optimisation

The inference problem can be handled in the framework of optimisation, where a cost function specifies the distance between predicted values and the real data. Convex optimization methods (Boyd & Vandenberghe 2004; Bertsekas & Scientific 2015) return the solution of unknown in system models by seeking the minimum of the cost function, which work well if the problem is convex. If the problem is not convex, optimisation algorithms

may reach to a local minimum. Different settings of starting point in the parameter space will result in different solutions. There are some attempts to solve this non-convex problem. For example, we can assign different starting points and repeat the inference many times. However, this multi-start fashion cannot guarantee a good solution. Therefore, global optimisation (Gounaris & Floudas 2009; Strongin & Sergeyev 2013) for non-convex problem is of demand. One challenge of using global optimisation methods is the computational cost increases dramatically with the problem sizes (Villaverde & Banga 2014).

3.6.2.2 Perspective from systems and control theory

The parameter estimation can be handled in the framework of control theory by using state observers (Sun et al. 2008). State observers, using the technique of state extension, transforms the parameters into states by suitably expanding the system (Fey et al. 2008). One typical parameter estimator is the Kalman filter (Welch & Bishop 2006). It is a recursive estimator, where at each step the filter refines the previous estimation by incorporating new information from experimental data. The method proposed in (Lillacci & Khammash 2010) applies the extended Kalman filter to a large parameter space to handle sparse and noisy data. This approach is based on the combination of an Extended Kalman Filter (EKF) algorithm, a statistical accuracy test and a moment matching procedure. One significant feature of this method is its ability to use information about the statistics of the measurement noise to validate and refine the estimation results (Lillacci & Khammash 2010). If the biological model is quite nonlinear, then Unscented Kalman Filter (UKF) is used to estimate the approximations of mean and covariance relying on small deterministic sets of appropriately chosen points (Quach et al. 2007; Capinski & Polanski 2016). However, the performance of Kalman filters is influenced by the settings of initial parameter values.

3.6.2.3 Perspective from Bayesian statistics

Another significant branch of parameter estimation algorithms is based on Bayesian inference where Bayes' rule is used to update the probability estimation of parameters by computing their likelihood (Galagali & Marzouk 2015; Sparacino & De Nicolao 2001). For complex probability models, computing the likelihood is time consuming and intractable. Therefore, Approximate Bayesian Computation (ABC) has been developed to avoid the direct evaluation of likelihood (Beaumont et al. 2002).

The basic ABC parameter estimation scheme is based on Monte Carlo rejection sampling, whereby sets of initial parameter values, such as concentrations and rate constants, which fall

within the appropriate ranges are generated randomly. Then the likelihood function, approximated by distance, is used to determine the acceptance of sampled parameters. Although in principle, this approach is simple, the parameter space to be explored is very large. Thus some efficient, less exhaustive, parameter searching algorithms should be used. The Markov Chain Monte Carlo (MCMC) based ABC algorithm is thus proposed where parameter search is performed via a proposal density function (Sisson et al. 2007). However, when the proposal density function is poorly chosen, the efficiency of this scheme is largely affected because the sampler may easily get stuck in a region where the chance of acceptance is low.

Therefore, the ABC Sequential Monte Carlo (SMC) method is developed to iterate towards the target posterior distribution via a series of intermediate distributions (Toni et al. 2009). Figure 3-5 selected from (Yang et al. 2014) shows the intermediate distributions of 4 variables obtained in the ABC SMC process at different iterations. It gives us an intuitive impression that ABC SMC sequentially narrows down the distributions and gradually approaches to the target one.

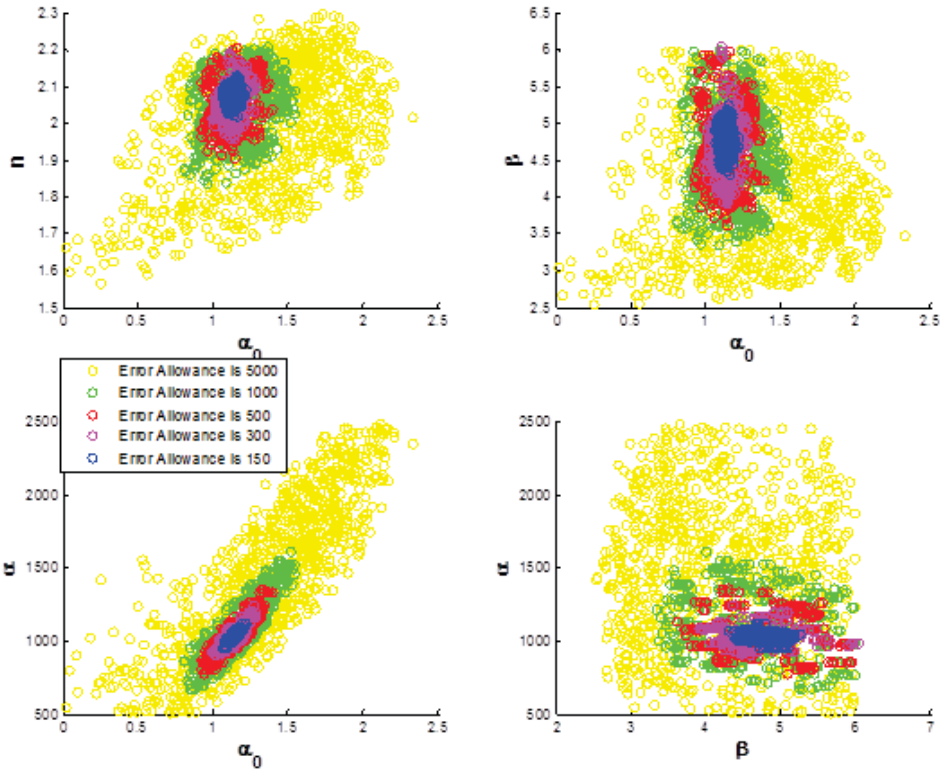


Figure 3-5 The outputs of the adaptive weighting ABC SMC method as scatter plots. This figure is selected from our previous work in (Yang et al. 2014).There are four variables of interest, whose distributions narrows down with smaller error allowance values.

One concern of Bayesian approximation approach is its scalability. A recent contribution in (Hug et al. 2013) have discussed the computational issues of applying Bayesian inference methods to high-dimensional parameter spaces and presented a multi-chain sampling method to infer a model with up to 100 parameters. This study gives us an example of using Bayesian methods to handle large-scale models.

3.6.3 Our interest: large scale network model inference under limited measurements

As it has been pointed out in (Villaverde & Banga 2014), well-posed inference problem should have the following properties: existence, uniqueness and stability. Existence means a solution exists, while uniqueness requires the solution is unique. Stability refers that the model is table under perturbation of noises. When the problem is ill-posed, inference methods cannot work well.

However, we cannot always guarantee to have a well-posed problem when we investigate models for large scale networks. It is common that we do not have much prior knowledge of the model that interactions among components are not known. Moreover, we may not know what exactly components are involved in the system. For example, there are vast kinds of proteins and genes in a cellular system. It is difficult to specify which proteins and genes play parts in the system when a specific stimulus is introduced. It is impossible to measuring all components independently. One typical issue in large scale network inference is that the observation is limited. With limited measurements, the inference problem becomes ill-posed as it is difficult to get a unique solution. In Chapter 8, we propose a possible way to do large scale network model inference with limited measurements.

3.7 Conclusion

In this chapter, we have discussed two main categories of questions in datafied life and reviewed analysis methods for answering different questions. Specifically, five questions and their related methods have been discussed, which are detecting associations (chapter 4), identifying groups (chapter 5), constructing classifiers (chapter 6), deriving connectivity (chapter 7) and building dynamic models (chapter 8). This chapter gives an overview of popular research approaches and focuses on introducing methods related to my own study.

4 Detecting associations: a Lasso based approach

4.1 Motivation

As having discussed in section 3.2, one typical research question is detecting variant and response association from expression measurements. There are many analysis methods available. One common branch of methods is based on linear modelling. For example, in fMRI analysis, general linear model (GLM) is used predominantly to discover voxels significantly associated with experimental conditions (more descriptions of GLM can be found in the following section). Another example is that Genome-wide association study (GWAS) commonly uses linear regression model to analyse quantitative traits which show continuous variation (such as weight or height) (Bush & Moore 2012). An example GWA study, published in the journal Nature Genetics, is led by the Genetic investigation of anthropometric traits (GIANT) consortium, focusing on finding loci (places in chromosome where genes are located) associated with body mass index (BMI), height as well as clinical classes of obesity (Berndt et al. 2013; Locke et al. 2015; Shungin et al. 2015; Wood et al. 2014). As obesity is the major factor of many chronic diseases, this study is important to public healthcare. Here we focus on discussing the association detection method based on linear modelling approach.

The typical form of the linear model is shown in Figure 4-1. Figure 4-1 a) shows the matrix representation of a simple linear model, while subplot b) gives us an extension form for the case of multiple responses. A response variable can be an independent measurement. Let us take the linear model in GWAS as an example. $\boldsymbol{Y} \in \mathbb{R}^{T \times J}$ can be a gene expression matrix where expression levels of J genes are measured for all T subjects. The desgin matrix $\boldsymbol{D} \in \mathbb{R}^{T \times K}$ can be a genotype matrix for T subjects, where each row represents the allele states of an individual at K loci. The (k, j) th element of regression coefficient matrix $\boldsymbol{X} \in \mathbb{R}^{K \times J}$ denotes the strength of kth SNP associated with jth gene. Elements $\boldsymbol{X}$ with larger values indicate a stronger association between the corresponding SNPs and genes. Statistic tests can further represent the significance of association by p-values.

As Lasso (Tibshirani 1994) has the advantage of selecting the most informative predictor (e.g., SNP) for each response (e.g., gene expression or clinical trait) and eliminating false positives, it is widely used in association mapping by introducing sparse constraints. In Lasso, the problem of estimating $\boldsymbol{X}$ can be formulated as an optimisation problem defined by

$$\widehat{\boldsymbol{X}} = \mathbf{argmin}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_1 \quad 4.1$$

where the first term represents a penalty based on prediction error and the second term is a sparsity constraint. The operator $\|\cdot\|_F$ is the Frobenius norm calculating the square root of the sum of the absolute squares of the matrix; and the operator $\|\cdot\|_1$ is the ℓ_1 norm. Elements of $\boldsymbol{X}$ indicate the strength of the corresponding predictors and responses associated.

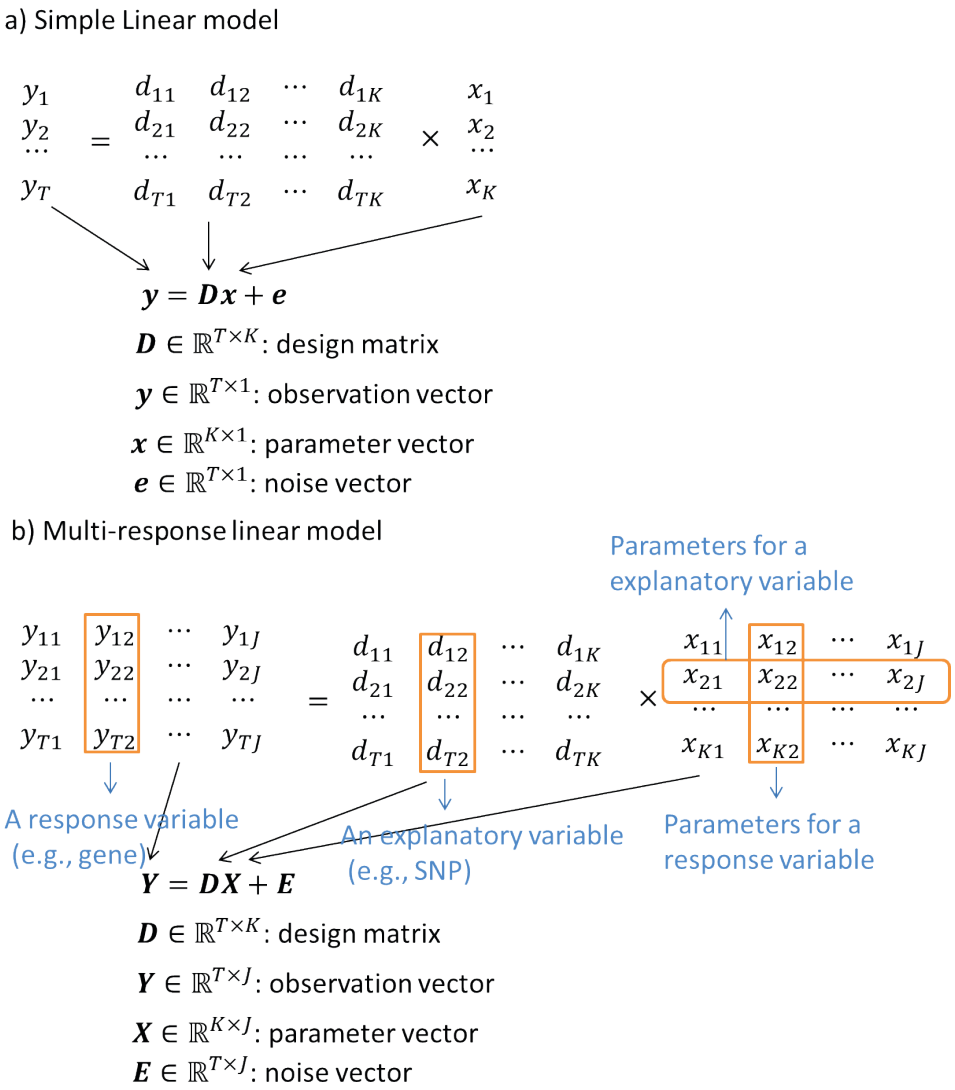


Figure 4-1 The illustration of linear models. a) shows the representation of a linear model; b) shows the representation of multiple-response (or multivariate) linear model, where each column of $\boldsymbol{Y}$ contains observations of a task.

Alternative forms of Lasso can also be applied to detect significant association by introducing extra constraints. For example, in GWAS, the work in (Curtis et al. 2012) assumes that highly

correlated genes tend to be influenced by a common subset of SNPs. This assumption is explicitly expressed as the second regularization term in the following expression:

$$\widehat{\boldsymbol{X}} = \mathbf{argmin}\|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{X}\|_1 + \gamma \sum_{\{i,j\}\in E_G} \sum_k (x_{ki} - sign(\rho_{ij})x_{kj}) \qquad 4.2$$

The third term encourages highly correlated genes (connected by an edge in E_G which is a graph containing indirect links among genes) to be associated with the same SNPs. ρ_{ij} is the correlation value between gene i and j; $sign(\rho_{ij})$ is the sign of ρ_{ij}, controlling the pattern of fusion applied to the association strengths. Further extensions of Lasso can be used if we make any other assumptions (or constraints).

In this chapter, we will introduce a Lasso based method to find significant voxels in task-based fMRI analysis. Traditional method of detecting significantly associated voxels is applying GLM to independently investigate individual voxels. For fMRI analysis, $\boldsymbol{y} \in \mathbb{R}^{T\times 1}$ in Figure 4-1 a) denotes time-series measurements of one voxel, while the design matrix $\boldsymbol{D} \in \mathbb{R}^{T\times K}$ contains K stimuli introduced over time. $\boldsymbol{x} \in \mathbb{R}^{K\times 1}$ indicates the association between stimuli and voxels, where the level of elements shows the strength of association. Selecting significant voxels independently ignores the correlations among voxels. fMRI images are spatially smooth, such that spatially neighbouring voxels are correlated. Consider the smoothness, we choose the model in Figure 4-1 b). Then $\boldsymbol{Y} \in \mathbb{R}^{T\times J}$ contains time-series measurements of all voxels; the (k, j)th element of $\boldsymbol{X} \in \mathbb{R}^{K\times J}$ contains the strength of the kth stimulus associated with the jth voxel. As this problem is similar with GWAS, we solve it in a Lasso style.

The value of elements in $\boldsymbol{X}$ can tell the strength of association. It is not a direct indicator of significance as p-values from a statistic test. In this chapter, we propose an indicator to represent relative significance of voxels. In the following sections, we will first discuss the problem of significant voxels detection in fMRI and then introduce our method in detail.

4.2 Background

Functional magnetic resonance imaging (fMRI) is a powerful tool to study the functional activity of brain by providing high quality of the location of activities (Smith et al. 2001). One typical fMRI analysis focuses on finding significant voxels activated by stimulations, where general linear model (GLM) is the most widely used approach (Friston et al. 1995). The time-series measurement of each voxel is decomposed into a linear combination of

explanatory variables (BOLD signals of stimulus) and error terms. The explanatory variables are calculated by convoluting the stimulus functions and haemodynamic response functions (HRF). Significant voxels are selected independently according to their corresponding P-values. The whole process of this analysis is shown in Figure 4-2.

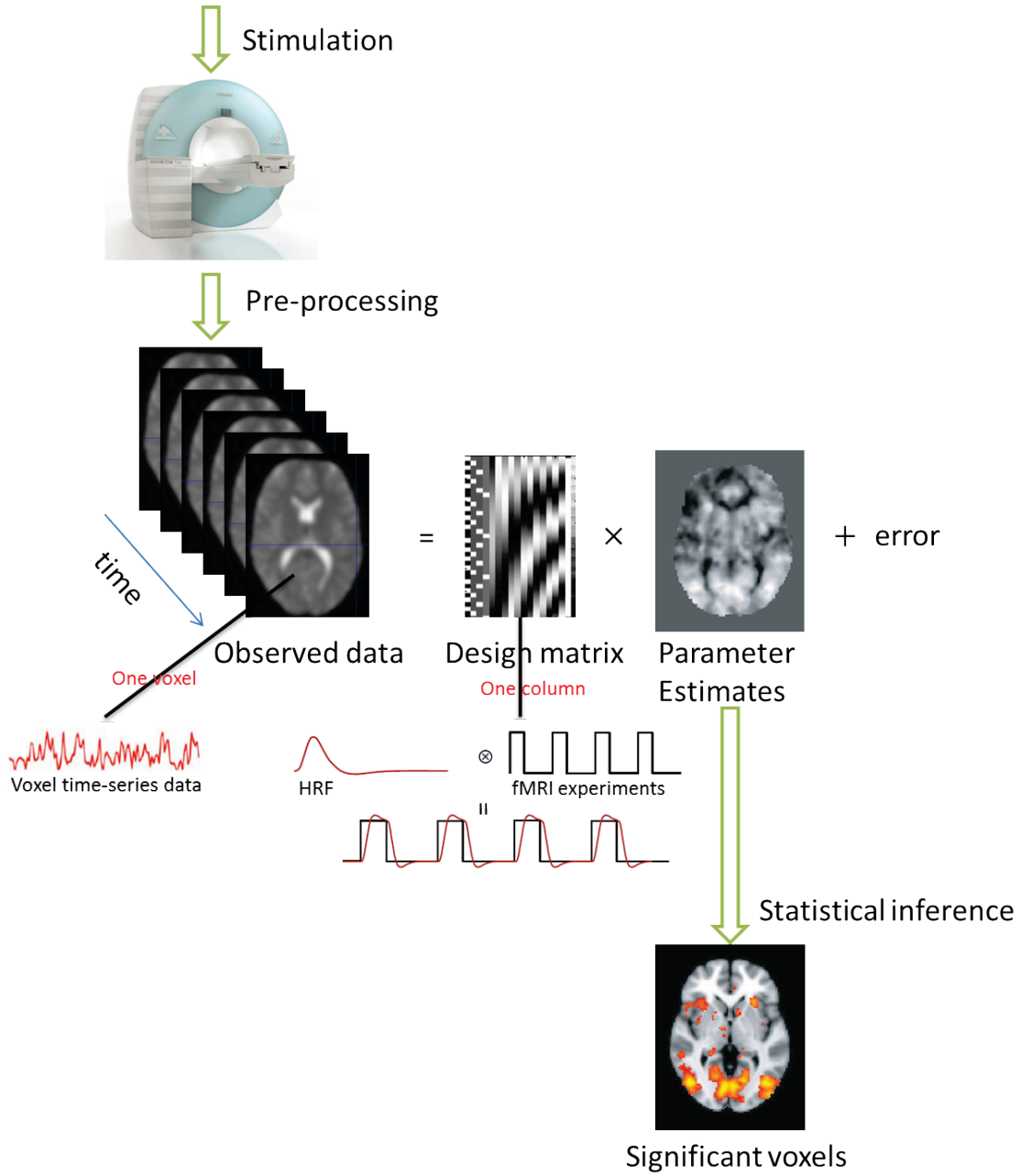


Figure 4-2 The fMRI analysis for significant voxel detection using GLM. fMRI measurements are first preprocessed and then fitted into a general linear model of design matrix. Each column of design matrix is obtained by convoluting one fMRI experiment design sequence (containing 1's and 0's, where 1 indicates the presence of stimulus and 0

indicates the absence) with the HRF. Then the significance of the parameter estimates from GLM is inferred.

The performance of the GLM method is greatly influenced by the noise level. When there are large noises, significant voxels are likely to be missed. In some studies, we cannot maintain noises at low level. For instance, there is an increasing interest in using task-based fMRI to detect brain activation with differences in condition, often by contrasts of serial imaging sessions over time. Additional sources of noise can be introduced into contrasts by factors such as day-to-day variation in the haemodynamic response (Bosnell et al. 2008; Bosnell et al. 2011) or subject movement (Liao et al. 2006). Noise reduction of fMRI data either demands longer signal averaging times or averaging signal changes over larger numbers of subjects. For applications to clinical populations, e.g., in assessing learning (Floyer-Lea & Matthews 2005; Floyer-Lea & Matthews 2004) or drug effects (Rabiner et al. 2011; Cader et al. 2006; Völlm et al. 2004), neither option may be practical. Therefore, it is necessary to develop a significant voxel detection method robust to noises.

To improve sensitivity and accuracy of signal detection from highly noisy fMRI data, potentially important information, such as spatial smoothness and connectivity constraints, therefore should be incorporated. Inspiring by the fact that active voxels are typically found adjacent to other active voxels in a cluster, several cluster-level inference methods have been proposed. These methods are commonly implemented in two steps: first, a threshold is applied to the statistic image and find groups of contiguous voxels (clusters); second, calculate the significance level of each cluster (Poldrack & Nichols 2011). A recent advance in this cluster-level inference is the threshold free cluster enhancement (TFCE) method that takes all threshold values and integrates over them to derive the cluster-level significance (Smith & Nichols 2009). However, even with this framework, spatial information is used as a second processing step after single voxel inference.

In order to make direct use of spatial constraints, a regularized general linear model implemented in the Bayesian framework by specifying spatial priors using Sparse Spatial Basis Functions was proposed (Penny et al. 2005). This approach uses Posterior Probability Maps (PPM) to characterize regional specific effects (Friston et al. 1995). As an exact Bayesian inference process is typically computationally intensive and frequently intractable, approximation approaches, such as variational Bayes method (Attias 1999) and the type-II maximum likelihood method (Berger 1985), are used to estimate the posterior distributions.

In this chapter, we propose an alternative index, the critical regularization value (CR-value), which detects significant variation in voxels by taking both the magnitude of the voxel-wise signal change and spatial smoothness into account. A voxel that survives with a stronger sparse constraint has a greater CR value indicating greater significance. To implement this idea, we solve GLM models using regularized optimization with various values of a penalty parameter λ. Fewer voxels will survive with larger values of the parameter λ. If the voxel A is selected while the voxel B is not for a given value of the parameter λ, we can claim that the voxel A is relatively more significant than the voxel B. The CR-value of a voxel is defined as the largest value of parameter λ to ensure this voxel being selected. Thus, we can construct a variational map using the CR-values of all voxels. Every CR-value takes into account spatial relationships between voxels because the regularized optimization includes all voxels in the brain image. We also extend formulation of the CR-value for multi-subject analysis (Argyriou et al. 2008). We demonstrate the performance of our method using both the simulated data and a real fMRI dataset from a study of motor learning reported previously (Tomassini et al., 2012).

4.3 Method

In this section, fMRI data are modelled using the general linear model. Then, we define our significant indictor, critical regularization value, for both single-subject and multi-subject analyses.

4.3.1 Single-subject analysis

For single-subject analysis, each fMRI dataset is fitted into a linear model as follows:

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X} + \boldsymbol{E} \tag{4.3}$$

The $T \times N$ matrix $\boldsymbol{Y}$ represents the fMRI dataset consisting of T time points at N voxels. These datasets are explained by K unknown effects included in the $T \times K$ design matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \dots, \boldsymbol{d}_K]$. The $K \times N$ matrix $\boldsymbol{X}$ contains the regression coefficients to be inferred. The element in the kth row and nth column of $\boldsymbol{X}$ indicates the association between the kth effect and the nth voxel. The $T \times N$ matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \dots, \boldsymbol{e}_N]$ represents the residuals. The objective is to find out voxels significantly associated with these effects. Traditional GLM approach does not consider the relationships among voxels, thus the association of individual voxel is inferred independently, turning Equation 4.3 into N independent equations:

$$\boldsymbol{y}_n = \boldsymbol{D}\boldsymbol{x}_n + \boldsymbol{e}_n \qquad \text{for } n \in \{1, \dots, N\} \tag{4.4}$$

where $\boldsymbol{y}_n = [y_{1,n}, \ldots, y_{t,n}, \ldots, y_{T,n}]'$ and $\boldsymbol{x}_n = [x_{1,n}, \ldots, x_{k,n}, \ldots, x_{K,n}]'$. $y_{t,n}$ is the element of $\boldsymbol{Y}$ on the tth row and nth column; $x_{k,n}$ is the element of $\boldsymbol{X}$ on the kth row and nth column.

As we have discussed, spatially neighbouring voxels are correlated and their regression coefficients for the same effect should be similar. There are many ways to define neighbouring voxels. Here, we define that two voxels sharing a common surface are neighbours. Therefore, each voxel has at most 6 neighbours as shown in Figure 4-3.

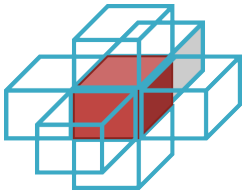


Figure 4-3 An example of neighbouring voxels. The voxel in red shares a surface with 6 other voxels. These 6 voxels are its neighbours.

Let $\boldsymbol{d}_k \in R^{T\times 1}$ be the kth effect and $\overline{\boldsymbol{x}}_k = [x_{k,1}, \ldots, x_{k,n}, \ldots, x_{k,N}]'$ contain its corresponding coefficient. To make the coefficients of neighbouring voxels in $\overline{\boldsymbol{x}}_k$ similar, we introduce a transformation matrix $\boldsymbol{\Gamma}$ that $\|\boldsymbol{\Gamma}\overline{\boldsymbol{x}}_k\|^2$ counts the total square differences between neighbours. The matrix $\boldsymbol{\Gamma}$ is constructed in a way that each row represents a neighbourhood relationship between two voxels. For example, we have only 4 voxels. If we get a row of $\boldsymbol{\Gamma}$ as $[1,0,0,-1]$, it indicates that the 1st and the 4th voxels are neighbours. Multiplying this row with $\overline{\boldsymbol{x}}_k$, we obtain $(x_{k,1} - x_{k,4})$ measuring the difference between the coefficients of these two voxels. The process of generating $\boldsymbol{\Gamma} \in \mathrm{R}^{S\times N}$ for 3D image is shown in Algorithm 4.1.

Algorithm 4.1	The process of generating the transformation matrix $\boldsymbol{\Gamma}$ for adding smooth constraint
Input	Sizes of 3D images: $\mathrm{size}_x, \mathrm{size}_y, \mathrm{size}_z$
Output	The transformation matrix $\boldsymbol{\Gamma}$ for smoothing
1	$\mathrm{N} \leftarrow \mathrm{size}_x \times \mathrm{size}_y \times \mathrm{size}_z$ // Total number of voxels
2	$\boldsymbol{\Gamma} \leftarrow []$ // Initialize $\boldsymbol{\Gamma}$ to be empty
3	**for** $n \leftarrow 1$ **to** N
	// get the 3D location of the nth voxel
4	$[\mathrm{x}_n, \mathrm{y}_n, z_n] \leftarrow \mathrm{ind2sub}([\mathrm{size}_x, \mathrm{size}_y, \mathrm{size}_z], n)$
	// check whether this voxel has neighbours in 3D

6 **if** $x_n + 1 \leq \text{size}_x$

// get the index of voxel whose 3D location is $[x_n + 1, \text{y}_n, z_n]$

7 $n_1 \leftarrow \text{sub2ind}([\text{size}_x, \text{size}_y, \text{size}_z], x_n + 1, \text{y}_n, z_n)$

8 $vec \leftarrow \text{zeros}(1, N)$

9 $vec(n) \leftarrow 1$

10 $vec(n_1) \leftarrow -1$

11 $\boldsymbol{\Gamma} \leftarrow [\boldsymbol{\Gamma}; vec]$

12 **end**

13 **if** $y_n + 1 \leq \text{size}_y$

// get the index of voxel whose 3D location is $[x_n, \text{y}_n + 1, z_n]$

14 $n_2 \leftarrow \text{sub2ind}([\text{size}_x, \text{size}_y, \text{size}_z], x_n, \text{y}_n + 1, z_n)$

15 $vec \leftarrow \text{zeros}(1, N)$

16 $vec(n) \leftarrow 1$

17 $vec(n_2) \leftarrow -1$

18 $\boldsymbol{\Gamma} \leftarrow [\boldsymbol{\Gamma}; vec]$

19 **end**

20 **if** $z_n + 1 \leq \text{size}_z$

// get the index of voxel whose 3D location is $[x_n, \text{y}_n, z_n + 1]$

21 $n_3 \leftarrow \text{sub2ind}([\text{size}_x, \text{size}_y, \text{size}_z], x_n, \text{y}_n, z_n + 1)$

22 $vec \leftarrow \text{zeros}(1, N)$

23 $vec(n) \leftarrow 1$

24 $vec(n_3) \leftarrow -1$

25 $\boldsymbol{\Gamma} \leftarrow [\boldsymbol{\Gamma}; vec]$

26 **end**

Ideally, $\boldsymbol{X}$ should be chosen to minimize the sum of coefficient differences among neighbouring voxels (i.e. $\|\boldsymbol{\Gamma}\overline{\boldsymbol{x}}_k\|^2$ for all k), while maintaining the prediction residuals $\|\boldsymbol{DX} - \boldsymbol{Y}\|_F^2$ as low as possible. As the number of voxels significantly associated with an effect is usually small, we can introduce a sparse constraint to $\boldsymbol{X}$. The $\boldsymbol{X}$ inference problem can be therefore formulated as

$$\boldsymbol{X} = \arg\min_{\boldsymbol{X}} \left(\frac{1}{2} \|\boldsymbol{DX} - \boldsymbol{Y}\|_F^2 + \frac{\alpha^2}{2} \sum_{k=1}^{K} \|\boldsymbol{\Gamma}\overline{\boldsymbol{x}}_k\|^2 + \lambda \sum_{k=1}^{K} \|\overline{\boldsymbol{x}}_k\|_1 \right) \quad 4.5$$

where $\|\cdot\|_1$ is the ℓ_1 norm; α is smoothing parameter controls the smoothness of the regression coefficients; and λ is the regularization parameter controls the sparseness of the coefficients. Equation 4.5 is in the similar format of Lasso (Tibshirani 1994). The only difference is that matrix $\boldsymbol{Y}$ and $\boldsymbol{X}$ in Lasso are vectors. Therefore, we can first reformulate Equation 4.3 and Equation 4.5 and then we can directly use the existing optimization method for Lasso to solve $\boldsymbol{X}$. Equation 4.3 can be turned into:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix}_{NT\times 1}}_{\mathbb{Y}} = \underbrace{\begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} \end{bmatrix}_{NT\times NK}}_{\mathbb{D}} \underbrace{\begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{K,1} \\ \vdots \\ x_{1,n} \\ x_{2,n} \\ \vdots \\ x_{K,n} \\ \vdots \\ x_{1,N} \\ x_{2,N} \\ \vdots \\ x_{K,N} \end{bmatrix}_{NK\times 1}}_{\mathbb{X}} + \underbrace{\begin{bmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_n \\ \vdots \\ \boldsymbol{e}_N \end{bmatrix}_{NT\times 1}}_{\mathbb{E}} \quad 4.6$$

Then the optimization problem can be reformulated as:

$$\mathbb{X} = \arg\min_{\mathbb{X}} \left(\frac{1}{2}\|\mathbb{D}\mathbb{X} - \mathbb{Y}\|_F^2 + \frac{\alpha^2}{2}\|\mathbb{T}\mathbb{X}\|^2 + \lambda\|\mathbb{X}\|_1 \right) \quad 4.7$$

where the transformation matrix $\mathbb{T}$ is derived from $\boldsymbol{\Gamma}$. Let $\tau_{s,n}$ denote the element in the sth row and nth column of $\boldsymbol{\Gamma}$. Then we set

$$\mathbb{t}_s = [\tau_{s,1} \quad \boldsymbol{0}_{1\times(K-1)} \quad \tau_{s,2} \quad \boldsymbol{0}_{1\times(K-1)} \quad \cdots \quad \tau_{s,N} \quad \boldsymbol{0}_{1\times(K-1)}] \quad 4.8$$

We define the function $\delta(\mathbb{t}_s, k)$ to circularly shift the elements in $\mathbb{t}_s$ by k positions. For example, $\delta(\mathbb{t}_s, 1)$ is:

$$\delta(\mathbb{t}_s, 1) = [0 \quad \tau_{s,1} \quad \boldsymbol{0}_{1\times(K-1)} \quad \tau_{s,2} \quad \boldsymbol{0}_{1\times(K-1)} \quad \cdots \quad \tau_{s,N} \quad \boldsymbol{0}_{1\times(K-2)}] \quad 4.9$$

$\mathbb{T}$ is constructed as:

$$\mathbb{T} = \begin{bmatrix} \mathbb{t}_1 \\ \delta(\mathbb{t}_1, 1) \\ \vdots \\ \delta(\mathbb{t}_1, K-1) \\ \mathbb{t}_2 \\ \delta(\mathbb{t}_2, 1) \\ \vdots \\ \delta(\mathbb{t}_2, K-1) \\ \vdots \\ \mathbb{t}_N \\ \delta(\mathbb{t}_N, 1) \\ \vdots \\ \delta(\mathbb{t}_N, K-1) \end{bmatrix}_{KS \times NK} \tag{4.10}$$

It is not hard to see that Equation 4.7 can be regarded as a Lasso estimate with the following definitions:

$$\begin{aligned} \widetilde{\mathbb{Y}} &= \begin{bmatrix} \mathbb{Y} \\ \mathbf{0}_{KS \times 1} \end{bmatrix} \\ \widetilde{\mathbb{D}} &= \begin{bmatrix} \mathbb{D} \\ \alpha \mathbb{T} \end{bmatrix} \end{aligned} \tag{4.11}$$

As a result, we get

$$\mathbb{X} = \arg\min_{\mathbb{X}} \left(\frac{1}{2} \left\| \widetilde{\mathbb{D}}\mathbb{X} - \widetilde{\mathbb{Y}} \right\|_F^2 + \lambda |\mathbb{X}|_1 \right) \tag{4.12}$$

The optimization method used to solve this equation is from (Wright et al. 2009). In this subsection, we assume the parameter α is given and fixed. We will discuss how to define it in the later subsection.

The parameter λ determines the level of sparseness and controls the complexity of the model. All the regression coefficients can be regulated to zero if the parameter λ is sufficiently large. With the decrease of λ, the coefficients $\mathbb{X}$ will gradually become nonzero one by one. The larger the parameter λ is, the sparser the matrix $\mathbb{X}$ will be. For any or two voxels with non-zero and zero coefficients, the voxel with the non-zero coefficient will be considered more significant than the one with zero coefficient, as it will be more likely to be chosen in a sparser model with larger parameter λ.

Based on this observation, we propose an index, called the critical regularization value (CR-value), which expresses the relative significance of regression coefficients. CR-value of $x_{k,n}$ is defined as

$$c_{k,n} = \max\{\lambda | x_{k,n}(\lambda) \neq 0\} \quad 4.13$$

where $x_{k,n}(\lambda)$ denotes the optimal value of Equation 4.12 with the parameter λ. The CR-value $c_{k,n}$ for the kth effect and the nth voxel is the maximum of all possible values of the parameter λ making the corresponding coefficient $x_{k,n}$ nonzero. When we are only interested in investigating the significance level of voxels in response to the kth effect, then we can focus on comparing the CR-values which are related to the kth effect.

If $c_{k,n_1} < c_{k,n_2}$, then the coefficient x_{k,n_2} is more likely to be non-zero than x_{k,n_1}. Figure 4-4 provides a simple illustration of how CR-values are calculated. The whole process of CR-value calculation is described in Algorithm 4.2. The CR-value calculation involves solving Lasso equations. In our approach, we increase λ with fix step size and find Lasso solutions for all tested λ values. Then we determine CR-values for all voxels by checking Equation 4.13.

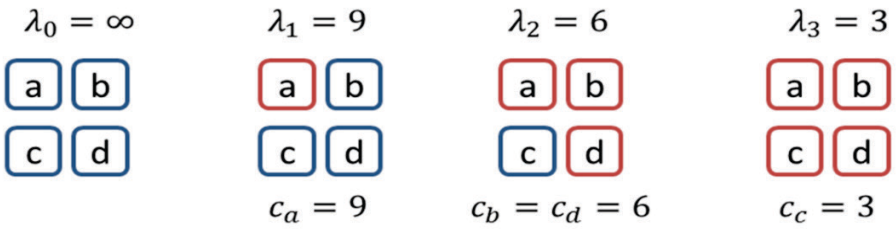


Figure 4-4 An illustration of CR-value calculation. There are 4 voxels that are denoted as a,b,c and d. If the coefficient of a voxel is zero, it is coloured in blue. Otherwise, it is coloured in red. All the coefficients are zero if the parameter λ is sufficiently large. When the parameter λ decreases to 9, the coefficient of voxel-a firstly becomes nonzero. Therefore, the CR-value of voxel-a is 9. As the parameter λ continually decreases to 6, the coefficients of voxel-b and voxel-d are nonzero for the first time. The CR-values of both voxel-b and voxel-d are 6.

Algorithm 4.2	The CR-value algorithm
Input	$\boldsymbol{D}$: design matrix
	$\boldsymbol{Y}$: fMRI signals consisting of T time points at N voxels
	$\text{size}_x, \text{size}_y, \text{size}_z$: sizes of 3D images
Output	CR-values $\{c_{k,n}\}_{k=1,..K;n=1,...,N}$
1	$N \leftarrow$ Total number of voxels
2	$K \leftarrow$ Number of effects

```
3    transform_matrix_gen(size_x, size_y, size_z)//Use Algorithm 4.1 to generate Γ
4    Form 𝕐 and 𝔻 following Equation 4.6 using D and Y
5    Construct 𝕋 following Equation 4.10
6    Get 𝕐̃ and 𝔻̃ following Equation4.11
7    Initialize λ with a small value
8    step_λ ← increment of λ in each iteration
9    do
         // use optimization methods to solve Lasso
10       𝕏(λ) ← Lasso_optimization(𝕐̃, 𝔻̃, λ)
11       save 𝕏(λ)
12       λ ← λ + step_λ
13   while 𝕏(λ) is not a zero vector
14   for n ← 1 to N  //each voxel
15       for k ← 1 to K  //each effect
16           p ← K × (n − 1) + k
17           for all λ
18               x_{k,n}(λ) ← the pth element in 𝕏(λ)
19           end
20           c_{k,n} = max{λ | x_{k,n}(λ) ≠ 0}
21       end
22   end
```

4.3.2 Multi-subject analysis

We can extend our method for multi-subject analyses to find significant voxels across multiple subjects. Consider L subjects in a study. We apply the same definitions of matrices in Section 4.3.1 and use the subscript $l \in \{1, \dots, L\}$ to denote the subject. For the lth subject, we get $\widetilde{\mathbb{D}}^l$ and $\widetilde{\mathbb{Y}}^l$ as above. The target vectors $\mathbb{X}^l \in R^{NK\times 1}$ for all l can form a matrix as

$$\bar{\mathbb{X}} = [\mathbb{X}^1, \dots, \mathbb{X}^l, \dots, \mathbb{X}^L] = \begin{bmatrix} x_{1,1}^1 & x_{1,1}^2 & \cdots & x_{1,1}^L \\ x_{2,1}^1 & x_{2,1}^2 & & x_{2,1}^L \\ \vdots & \vdots & & \vdots \\ x_{K,1}^1 & x_{K,1}^2 & & x_{K,1}^L \\ \vdots & \vdots & & \vdots \\ x_{1,n}^1 & x_{1,n}^2 & & x_{1,n}^L \\ x_{2,n}^1 & x_{2,n}^2 & & x_{2,n}^L \\ \vdots & \vdots & & \vdots \\ x_{1,N}^1 & x_{1,N}^2 & & x_{1,N}^L \\ x_{2,N}^1 & x_{2,N}^2 & & x_{2,N}^L \\ \vdots & \vdots & & \vdots \\ x_{K,N}^1 & x_{K,N}^2 & & x_{K,N}^L \end{bmatrix} \quad 4.14$$

Let $\mathbb{x}_r$ denote the rth row of $\bar{\mathbb{X}}$, corresponding to the rth feature across all subjects. Figure 4-5 depicts relations of matrices used in multi-subject analyses.

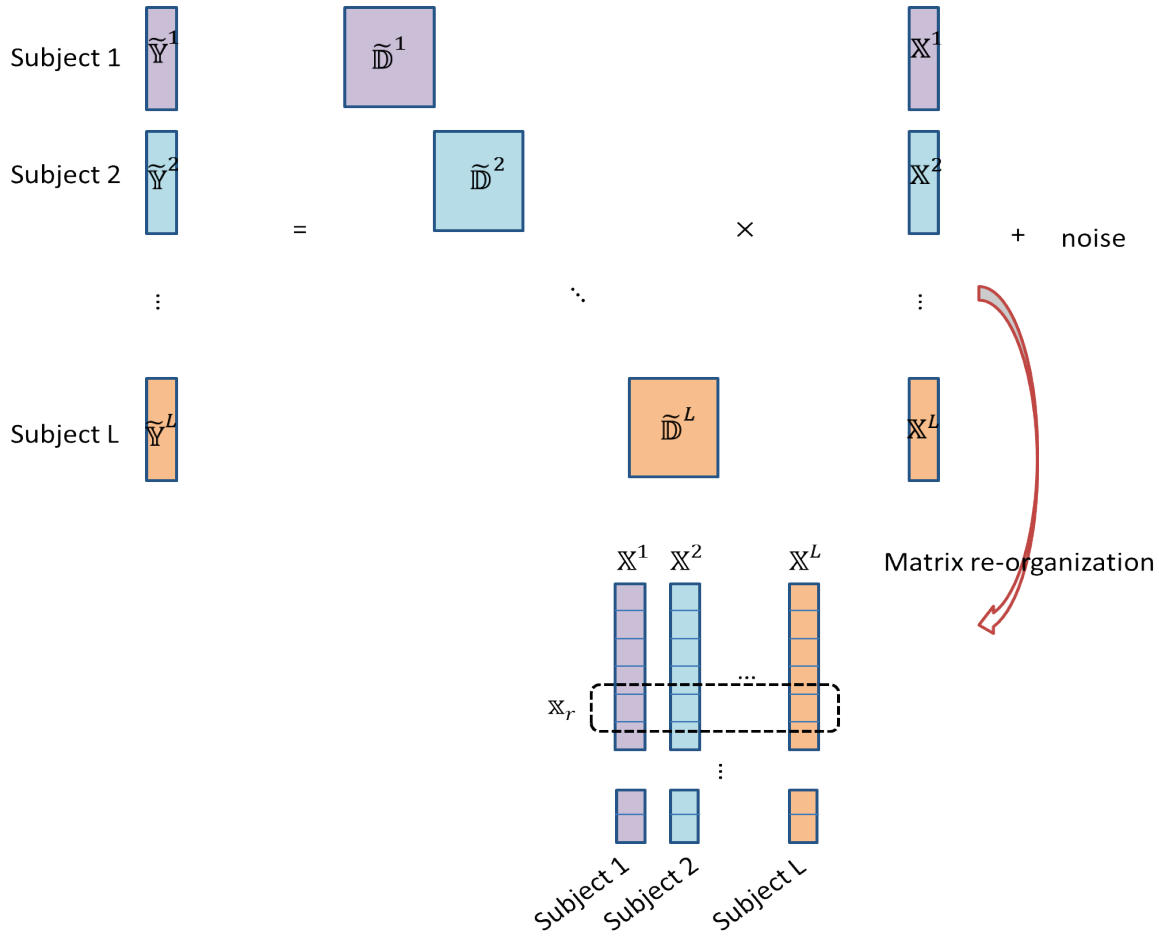


Figure 4-5 Relations of matrices used in multi-subject analysis to detect significant features across all subjects. $\{\widetilde{\mathbb{D}}^l\}_{l=1}^L$ and $\{\widetilde{\mathbb{Y}}^l\}_{l=1}^L$ are used to generate $[\mathbb{X}^1, \mathbb{X}^2, \dots, \mathbb{X}^L]$, whose rth row is represented by $\mathbb{x}_r$.

To get common features across samples, we utilize multi-task learning (Argyriou et al. 2008) by formulating the optimization problem as:

$$\overline{\mathbb{X}} = \arg\min_{\overline{\mathbb{X}}} \left(\sum_{l=1}^{L} \frac{1}{2} \left\| \widetilde{\mathbb{D}}^l \mathbb{X}^l - \widetilde{\mathbb{Y}}^l \right\|_F^2 + \lambda \|\overline{\mathbb{X}}\|_{2,1} \right) \quad 4.15$$

where $\|\overline{\mathbb{X}}\|_{2,1}$ is the regularization term penalizing the ℓ_2/ℓ_1-norm of matrix $\overline{\mathbb{X}}$. It is obtained by first calculating the ℓ_2 norms of rows $\mathbb{x}_r$ and then ℓ_1 norms of the vector $[\|\mathbb{x}_1\|_2, \dots, \|\mathbb{x}_r\|_2, \dots, \|\mathbb{x}_R\|_2]$. The magnitude of $\|\mathbb{x}_r\|_2$ indicates the importance of r th feature.

The group CR-value for multi-subject analyses is defined as:

$$c_{k,n} = \max\{\lambda | \exists l \in \{1, \dots, L\} \text{ that } x_{k,n}^l(\lambda) \neq 0\} \quad 4.16$$

Different from the single-subject analysis, the group CR-value $c_{k,n}$ is the maximum of values of the parameter λ to ensure that at least one coefficient $x_{k,n}^l$ is nonzero. As we have discussed, if we are only interested in finding the significance level of voxels for the kth effect, then we can focus on comparing the CR-values which are associated with the kth effect.

Algorithm 4.3	The group CR-value algorithm
Input	$\{\boldsymbol{D}^l\}_{l=1}^L$: design matrix for L subjects
	$\{\boldsymbol{Y}^l\}_{l=1}^L$: fMRI signals for L subjects
	$\text{size}_x, \text{size}_y, \text{size}_z$: sizes of 3D images
Output	CR-values $\{c_{k,n}\}_{k=1,..K;n=1,...,N}$
1	$N \leftarrow$ Total number of voxels
2	$K \leftarrow$ Number of effects
3	$L \leftarrow$ Number of subjects
4	transform_matrix_gen($\text{size}_x, \text{size}_y, \text{size}_z$)//Use Algorithm 4.1 to generate Γ
5	Form $\mathbb{Y}^l$ and $\mathbb{D}^l$ following Equation 4.6 using $\boldsymbol{D}^l$and $\boldsymbol{Y}^l$ for all $l \in \{1, \dots, L\}$
6	Construct $\mathbb{T}$ following Equation 4.10
7	Get $\widetilde{\mathbb{Y}}^l$ and $\widetilde{\mathbb{D}}^l$ for all $l \in \{1, \dots, L\}$ following Equation4.11
8	Initialize λ with a small value
9	$step_\lambda \leftarrow$ increment of λ in each iteration

10 **do**

// use multi-task learning methods

11 $\overline{\mathbb{X}}(\lambda) \leftarrow$ multi_task_optimization($\{\widetilde{\mathbb{Y}}^l\}_{l=1}^{L}$,$\{\widetilde{\mathbb{D}}^l\}_{l=1}^{L}$, λ)

12 save $\overline{\mathbb{X}}(\lambda)$

13 $\lambda \leftarrow \lambda + step_\lambda$

14 **while** $\overline{\mathbb{X}}(\lambda)$ is not a zero matrix

15 **for** $n \leftarrow 1$ **to** N //each voxel

16 **for** $k \leftarrow 1$ **to** K //each effect

17 $p \leftarrow K \times (n-1) + k$

18 **for** all λ

19 **for** all l

20 $x_{k,n}^{l}(\lambda) \leftarrow$ the pth row and lth column in $\overline{\mathbb{X}}(\lambda)$

21 **end**

22 $x_{k,n}(\lambda) = \sum_{l=1}^{L} |x_{k,n}^{l}(\lambda)|$

23 **end**

24 $c_{k,n} = \max\{\lambda | x_{k,n}(\lambda) \neq 0\}$

25 **end**

26 **end**

4.3.3 Permutation test

Suppose a subject is scanned at the presence of an experimental effect (which determines the design matrix). Observation of a large CR-value for a voxel is likely that this voxel has significant association with the design matrix. Permutation test provides a formal mechanism to quantify the likelihood, where the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of data (Nichols & Holmes 2001). To carry out the permutation test of the CR-value, we serially perturb the design matrix and calculate new CR-values for the voxel. The design matrix is generated by convolving the experimental condition matrix with HRF. The experimental condition matrix contains a sequence of 1's and 0's, where 1 indicates the presence of a given condition and 0 represents its absence. Convolution of experimental condition matrix with HRF can mimic the measurement of fMRI signal. This is because fMRI is based on studying the vascular response in the brain to neuronal activity, where blood oxygenation level-dependent (BOLD) contrast is used to investigate local changes blood oxygenation.

In the permutation test, we first calculate CR-values without perturbation. Then we randomly perturb the experimental condition matrix, in which the order of 1's and 0's is changed. The next step is convolving the perturbed experimental condition matrix with HRF to get the perturbed design matrix. With the perturbed design matrix, we estimate a new set of CR-values. Save this newly obtained CR-values and repeat above steps many times. Afterwards, we will obtain a set of CR-values, which form the distribution of possible CR-values under the null hypothesis that voxels are not significant associated with the design matrix. The p-value of the test for each voxel is calculated as the proportion of perturbed CR-values is greater than or equal to its CR-value without perturbation.

The above procedure is carried out in our experiment, where in our case design matrix is constructed by only one regressor. As it has been pointed out in (Winkler et al. 2014), with only one regressor, perturbing design matrix is equivalent to perturbing fMRI signal. For more general case that more than one regressors are at presence, we should first regress out regressors that are not of interest (Freedman & Lane 1983). Then perturb residual fMRI signal and add back the regressed out part of signal. With the perturbed signal, we can estimate a set of new CR-values. Repeating these steps many times, we can obtain the p-value of each voxels.

It is noteworthy to mention that so far we haven't considered the fact that serial autocorrelation in time series violates the fundamental assumption needed for permutation, which is exchangeability. Direct perturbation of time series signal does not preserve the temporal dependence structure. Therefore, it is necessary to transform the time series to a domain, where perturbation preserves statistical structure. Whitening resampling discussed in (Friman & Westin 2005) is one approach making time series uncorrelated. AR(1) model is used to capture serial correlation in fMRI time series where the observed time series is modelled by $y_{t,n} = \eta y_{t-1,n} + e_{t,n}$. The parameter η indicates the strength of the serial correlation and $e_{t,n}$ is white Gaussian noise. To estimate η, the Yule-Walker method is used. The whitened time series $\hat{e}_{t,n}$ is approximately uncorrelated. We can perturb $\hat{e}_{t,n}$ and use it to generate a new AR(1) process with estimated η. Using the newly generated signal, we can estimate a set of new CR-values.

4.3.4 Smoothing parameter determination

In the above subsection, we assume the smoothing parameter is given and fixed. Now let us discuss how to choose a smoothing parameter. Traditionally, cross-validation is commonly

used for justifying a parameter of a prediction algorithm. In Lasso, cross-validation is used to determine the regularization parameter λ. As Lasso is a regression method simultaneously performing variable selection and shrinkage, the chosen coefficients are brought closer to zero. In order to prevent relevant coefficients from being shrunk in this way, cross-validation tends to select a value of λ that is not too large. In this case, more irrelevant variables are selected via cross-validation; and the resulting model does not necessarily correspond to a "true model" able to generate individual observational data (Murphy 2012). This is one limitation of cross-validation. In this paper, we do not apply cross-validation to determine the value of smoothing parameter. The main reason is that we find the significance of coefficients by varying the regularization parameter λ. As λ is changing, we cannot adopt the cross-validation method, which aims to return an optimal parameter pair. Therefore, we stress that the smoothing parameter should be adjusted and validated using prior knowledge. The extend of smoothing should be ideally matched to the signal extent.

To analyse the smoothness of a variational map, we select a set of active voxels with P-values from permutation tests exceed a given significance level (e.g., $p<0.05$). A voxel is regarded as a surface of the active voxel set, if it is inactive and share at least one face with an active voxel. Then, we can calculate the surface area of the active voxel set, which is defined as the total number of surface voxels. The process of calculation is shown in Algorithm 4.4. The optimal smoothing parameter is chosen to be the one that resulting in the most reasonable surface area according to the prior knowledge.

Algorithm 4.4	The process of calculating surface area
Input	3D image $\boldsymbol{I}$
Output	Surface area S
1	Initialize size_x, size_y and size_z to be the sizes in 3D dimensions of $\boldsymbol{I}$
2	$Index \leftarrow find(\boldsymbol{I})$ //find index of non-zero voxels
3	$\text{N} \leftarrow \text{length}(Index)$ // Total number of non-zero voxels
4	$S \leftarrow 0$ // Initialize surface area
5	$\boldsymbol{F} \leftarrow \mathbf{0}_{\text{size}_x \times \text{size}_y \times \text{size}_z}$ // record whether voxels have been counted
6	**for** $n \leftarrow 1$ **to** N
	// get the 3D location of the nth voxel
7	$[\text{x}_n, \text{y}_n, z_n] \leftarrow \text{ind2sub}([\text{size}_x, \text{size}_y, \text{size}_z], Index(n))$
8	// check whether this voxel has surface voxels

9 **if** $x_n + 1 \leq \text{size}_x$ && $\boldsymbol{I}(\mathrm{x}_n + 1, \mathrm{y}_n, z_n) == 0$ && $\boldsymbol{F}(\mathrm{x}_n + 1, \mathrm{y}_n, z_n) == 0$

10 $\boldsymbol{F}(\mathrm{x}_n + 1, \mathrm{y}_n, z_n) = 1$

11 $S \leftarrow S + 1$

12 **end**

13 **if** $y_n + 1 \leq \text{size}_y$ && $\boldsymbol{I}(\mathrm{x}_n, \mathrm{y}_n + 1, z_n) == 0$ && $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n + 1, z_n) == 0$

14 $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n + 1, z_n) = 1$

15 $S \leftarrow S + 1$

16 **end**

17 **if** $z_n + 1 \leq \text{size}_z$ && $\boldsymbol{I}(\mathrm{x}_n, \mathrm{y}_n, z_n + 1) == 0$ && $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n, z_n + 1) == 0$

18 $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n, z_n + 1) = 1$

19 $S \leftarrow S + 1$

20 **end**

21 **if** $x_n - 1 > 0$ && $\boldsymbol{I}(\mathrm{x}_n - 1, \mathrm{y}_n, z_n) == 0$ && $\boldsymbol{F}(\mathrm{x}_n - 1, \mathrm{y}_n, z_n) == 0$

22 $\boldsymbol{F}(\mathrm{x}_n - 1, \mathrm{y}_n, z_n) = 1$

23 $S \leftarrow S + 1$

24 **end**

25 **if** $y_n - 1 > 0$ && $\boldsymbol{I}(\mathrm{x}_n, \mathrm{y}_n - 1, z_n) == 0$ && $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n - 1, z_n) == 0$

26 $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n - 1, z_n) = 1$

27 $S \leftarrow S + 1$

28 **end**

29 **if** $z_n - 1 > 0$ && $\boldsymbol{I}(\mathrm{x}_n, \mathrm{y}_n, z_n - 1) == 0$ && $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n, z_n - 1) == 0$

30 $\boldsymbol{F}(\mathrm{x}_n, \mathrm{y}_n, z_n - 1) = 1$

31 $S \leftarrow S + 1$

32 **end**

4.4 Experimental results

Here we demonstrate the performance of our method using both the simulated data and real fMRI dataset (Tomassini et al. 2012). For a single subject analysis, the simulation results highlight the prediction accuracy of our method compared to other methods. It shows that our metric to evaluate the performance (i.e. surface areas) can help us to find a promising smoothing parameter. For a multi-subject analysis, the simulation results further demonstrate that our method performs well in defining common high signal across a group. In applying our method to real fMRI collected during a visuo-motor training study, we illustrate the potential application of our method for solving new research problems.

4.4.1 Simulation analysis

4.4.1.1 Experimental design

The design matrix used throughout this paper is shown in Figure 4-6, which contains a task correlated regressor (in blue) and its corresponding derivative response (in red). We only focus on investigating the performance of discovering voxels associated with the task regressor, where the time series $\boldsymbol{Y}$ are generated by 1) only the task correlated regressor; 2) both the task correlated regressor and its derivative.

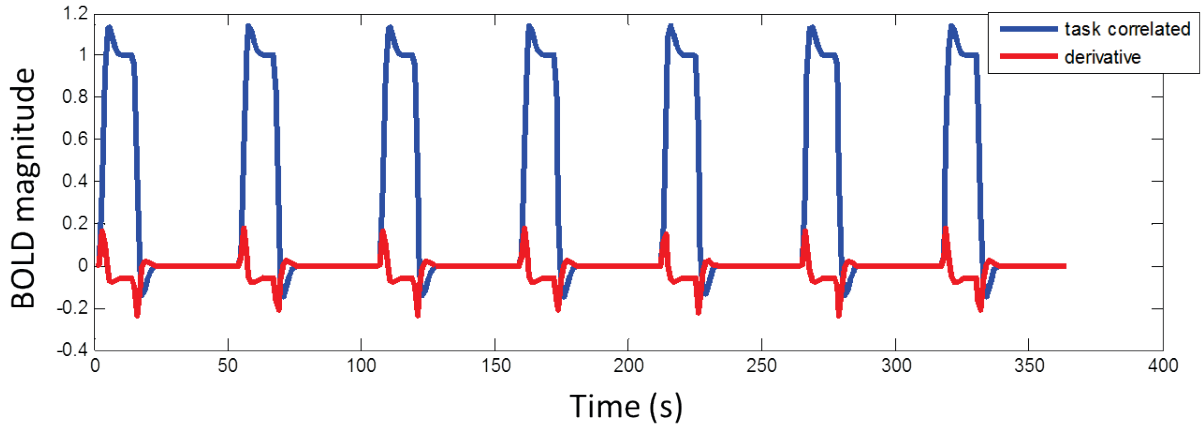


Figure 4-6 Design matrix containing two regressors. The task correlated regressor in blue is obtained by convolving a real fMRI experiment design with the canonical hemodynamic response function. The regressor in red is the derivative of the task correlated regressor.

The simulation study is carried out in two different cases that the regression coefficients for the regressors are generated differently as follows:

- **Case 1**: The spatially contiguous regression coefficients $\bar{\boldsymbol{x}}_1$ for the task correlated regressor are generated as shown in Figure 4-7(a). $\bar{\boldsymbol{x}}_1$ is represented by a 18×12 matrix with 25 non-zero pixels (in the black rectangular) grouped locally together. Each pixel has its colour ranging from blue to red and value ranging from 0 to 1 with a blue-red (low-high) colour scale. $\bar{\boldsymbol{x}}_2$ in response to the derivative regressor is generated in a similar way (not shown). The simulation experiment includes randomly generated noise from a Gaussian distribution with zero mean and standard deviation of 1.5. With the application of our method, locally grouped non-zero pixels should be found to be more significant than others.
- **Case 2**: The regression coefficients $\bar{\boldsymbol{x}}_1$ for the task correlated regressor are also represented by a 18×12 matrix, where the value at each pixel is designed to be 1. Similarly, $\bar{\boldsymbol{x}}_2$ in response to the derivative regressor has the common value of 0.5 for all pixels. In order to make individual pixels have different significance values in response to the task correlated regressor, different levels of noise are added to generate $\boldsymbol{Y}$, where noise over time at individual pixels was randomly generated from a Gaussian distribution with zero mean and standard deviation varying from 1 to 9. Intrinsically, pixels which are less noisy should have higher significance levels. Figure 4-7(b) shows the standard deviation of the time varying noise added to each pixel with a blue-red (low-high) colour scale as well. As it is shown in Figure 4-7(b), there are spatially grouped pixels in the black rectangular area with relatively low noise. Although some pixels outside the black rectangular also have similarly low noise, they are not locally clustered. Ideally, with the application of our method, which takes the smoothness constraint into consideration, the locally grouped pixels with relatively low noise should be found to be more significant than the others.

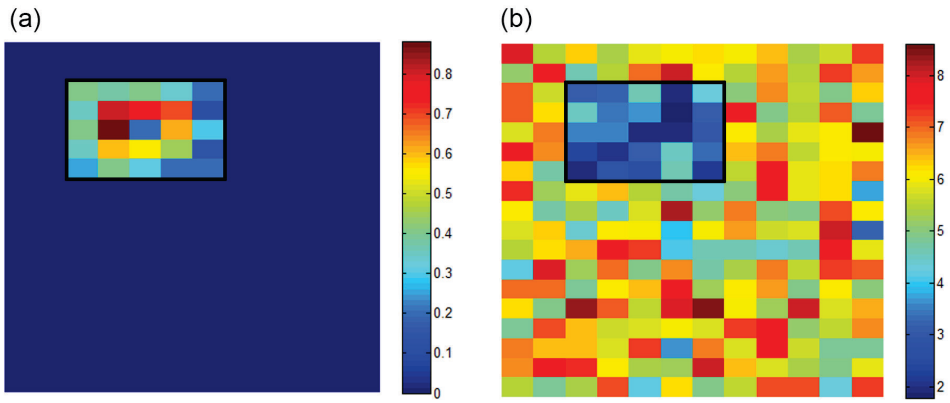


Figure 4-7 Design of simulations in two cases. (a) Case 1: Regression coefficients $\bar{x}_1$ for the task correlated regressor. $\bar{x}_1$ is represented by a 18×12 matrix with 25 non-zero pixels locally grouped together. Each pixel has a value ranging from 0 to 1 (blue-red colour scale). (b) Case 2: Standard deviation of noise added to each pixel. Time series of noise for each pixel is randomly generated from a Gaussian distribution with zero mean and a standard deviation varying from 1-9 (indicated in blue to red). Pixels in the black rectangular have relatively low noise (mean standard deviation of 3), while pixels outside this area have higher noise (mean standard deviation of 6).

In both cases, we use CR-value to find active pixels of the normalized simulation datasets with different values of the smoothing parameter. As there are 25 pixels in the black rectangular area that are designed to be more significant than the others, these have been used as a "gold standard" for specificity, sensitivity, accuracy and AUROC.

In signal subject analysis the step size for searching to get CR-values is set to be 0.1. To compare the performance of our method with others, we also show the pixel selection results using GLM P-values, TFCE-values (Smith and Nichols, 2009) and PPM-values from Bayesian approach (Penny et al., 2005). Three issues of using GLM P-values, TFCE-values and PPM-values methods should be noted. For GLM P-value results, we do not apply the false discovery rate correction or Bonferroni correction methods. This is because these corrections only help to choose significant pixels using P-values for multiple comparisons. As the rank of pixels resulting from P-values is not influenced, these methods will not change the selection of top significant pixels. For TFCE-value results, we do not turn them into P-values via permutation test as the rank of pixels will not change as well. TFCE-values are calculated following Equation 1 in (Smith and Nichols, 2009), where two parameters E and H are set to their recommended values (0.5 and 2 respectively). The spatial neighbours are

defined as pixels sharing a common edge in 2D image, corresponding to voxels sharing a common surface in 3D images. For PPM-value results, the Bayesian method in (Penny et al., 2005) is first applied to generate the posterior distribution of regression coefficients (Friston et al. 2003). Then PPM-values are obtained by calculating the probability that the coefficient is greater than 2% of the global mean.

In multi-subject analysis the step size for searching to get CR-values is set to be 1. The performance of the group CR-values is compared with the group GLM P-values. The group GLM P-values is calculated by first estimating coefficients of individual subjects. Then a statistic test generates p-values with null hypothesis that the voxels is not significant across multiple subjects.

4.4.1.2 Results

Figure 4-8, Figure 4-9, Figure 4-10 and Figure 4-11 show the single-subject analysis results under different simulation conditions: I) the simulation data is generated using the task correlated regressor as described in case 1; II) the simulation data is generated using the task correlated regressor as described in case 2; III) the simulation data is generated using both the task correlated regressor and its derivative as described in case 1; IV) the simulation data is generated using both the task correlated regressor and its derivative as described in case 2. Blue-red colour scale is used to indicate relative strength of associations for different voxels.

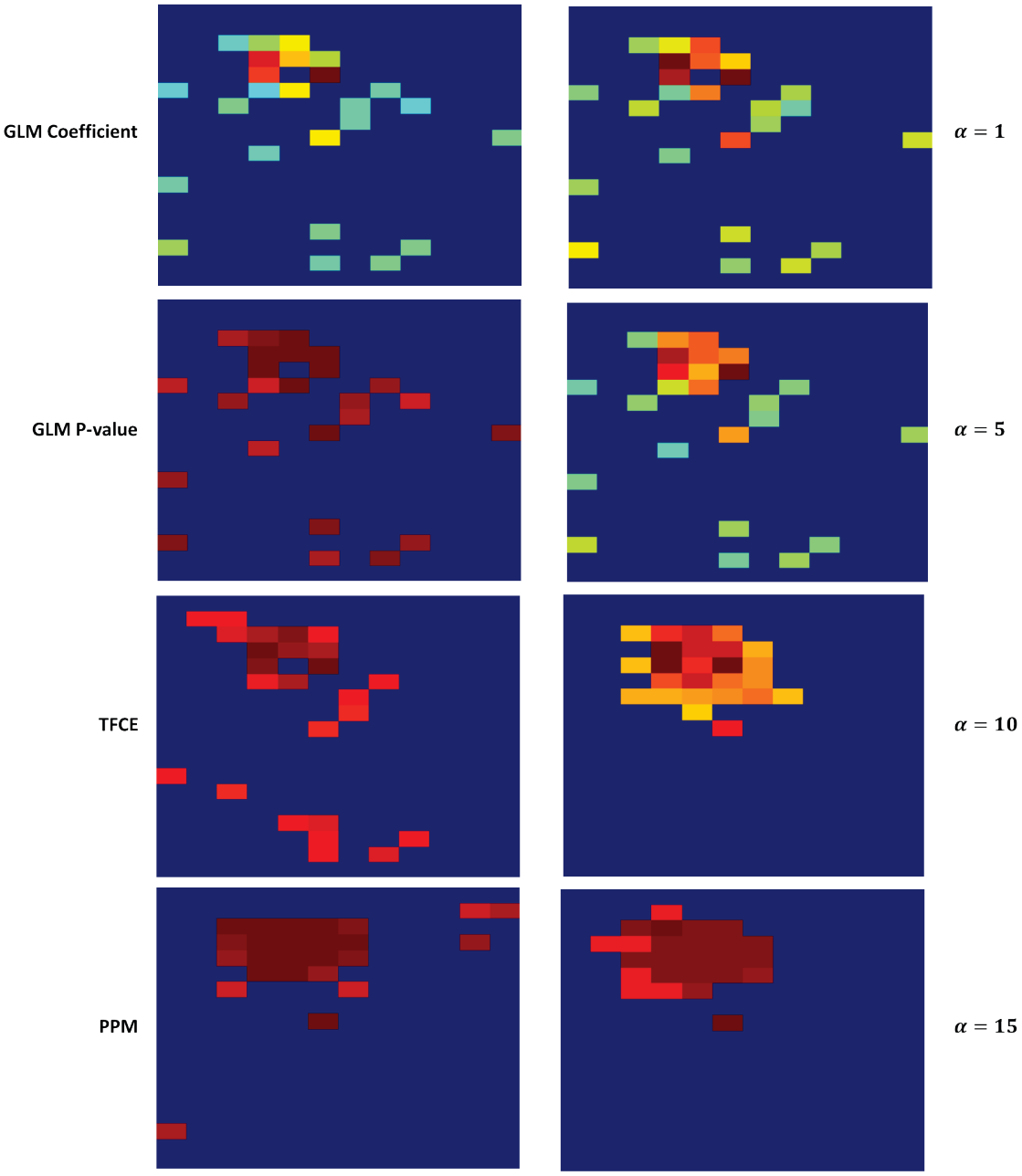


Figure 4-8 Single-subject analysis for Case1 using the task correlated regressor.

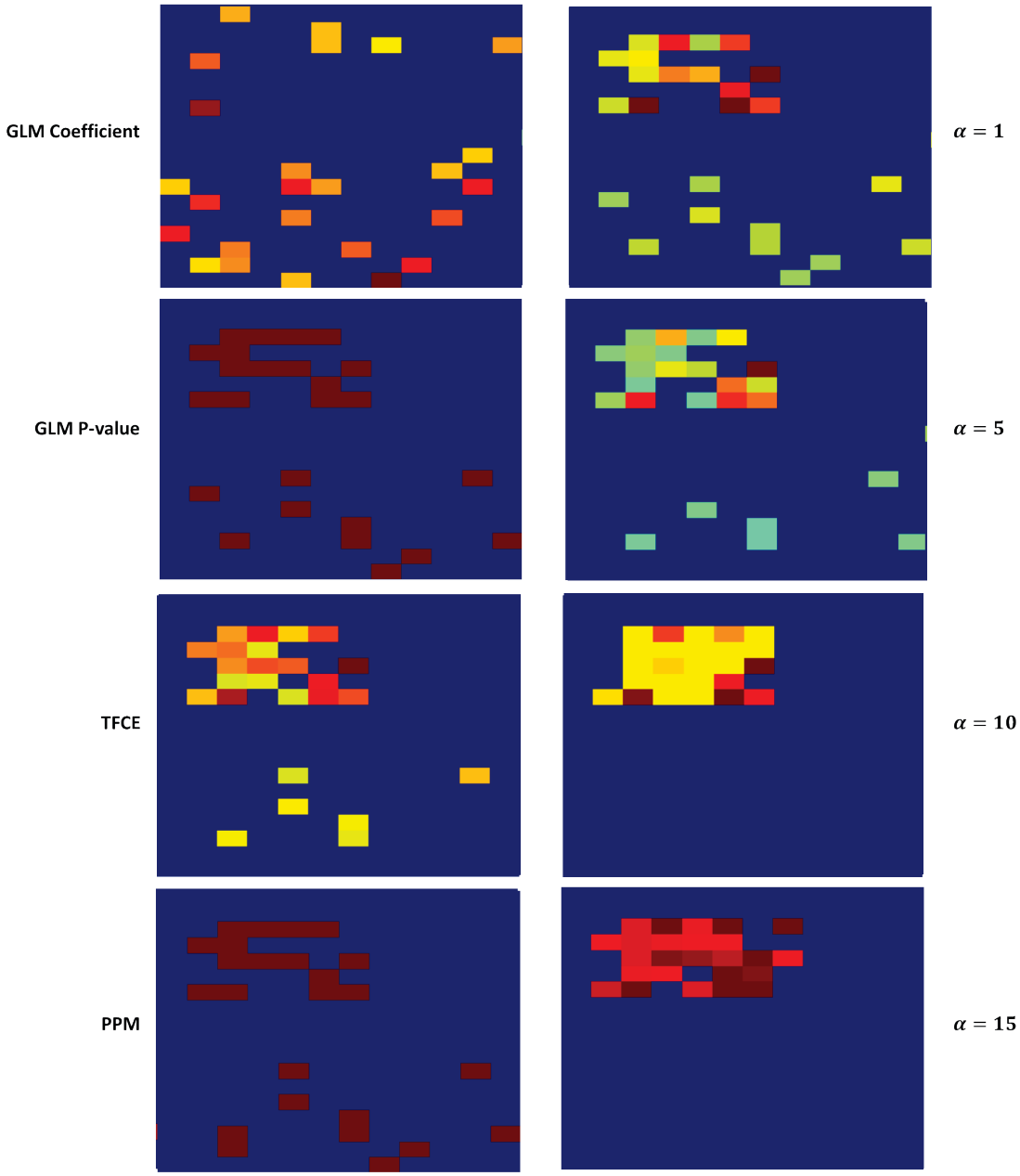


Figure 4-9 Single-subject analysis for Case2 using the task correlated regressor

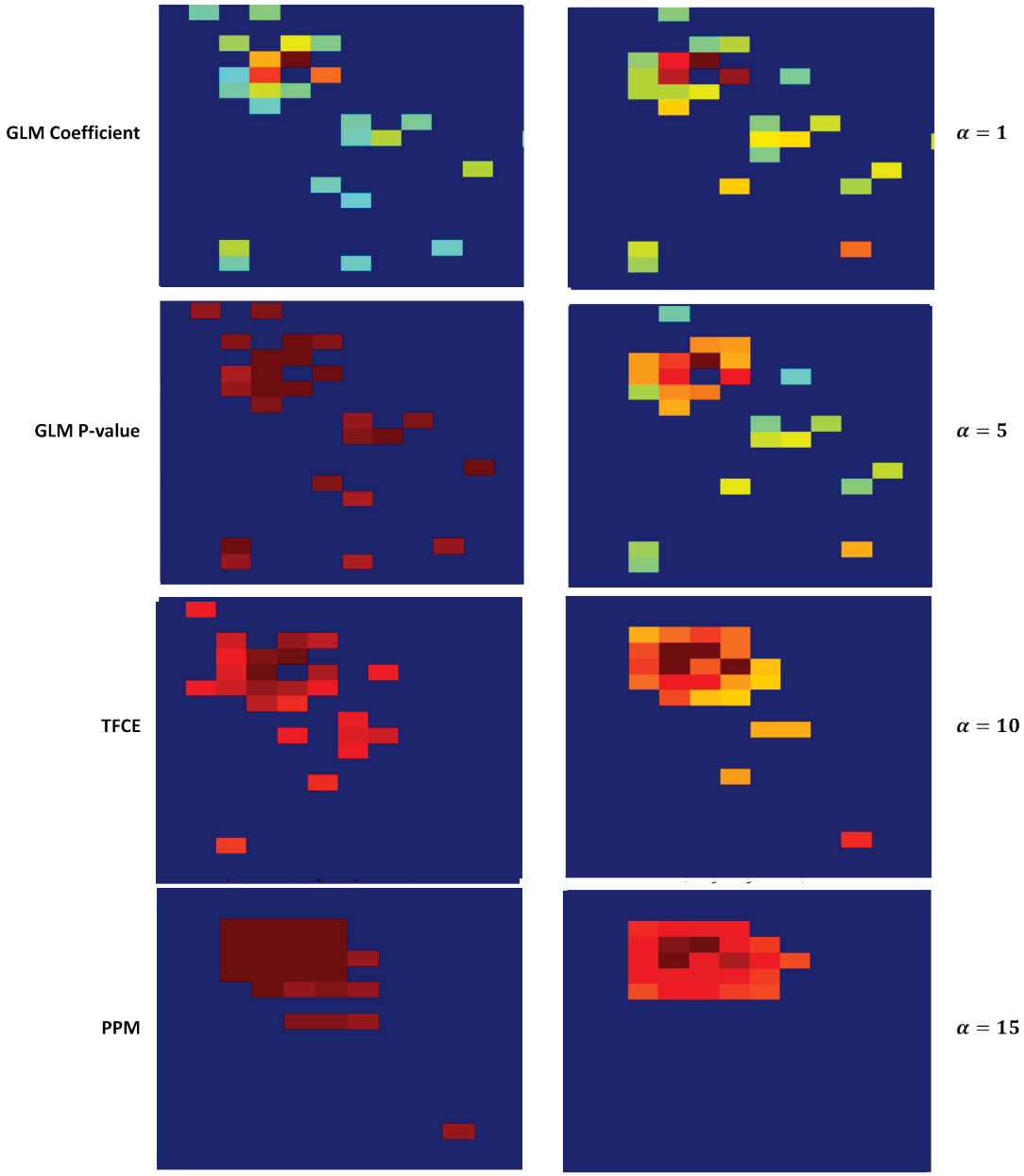


Figure 4-10 Single-subject analysis for Case1 using the task correlated regressor and its derivative

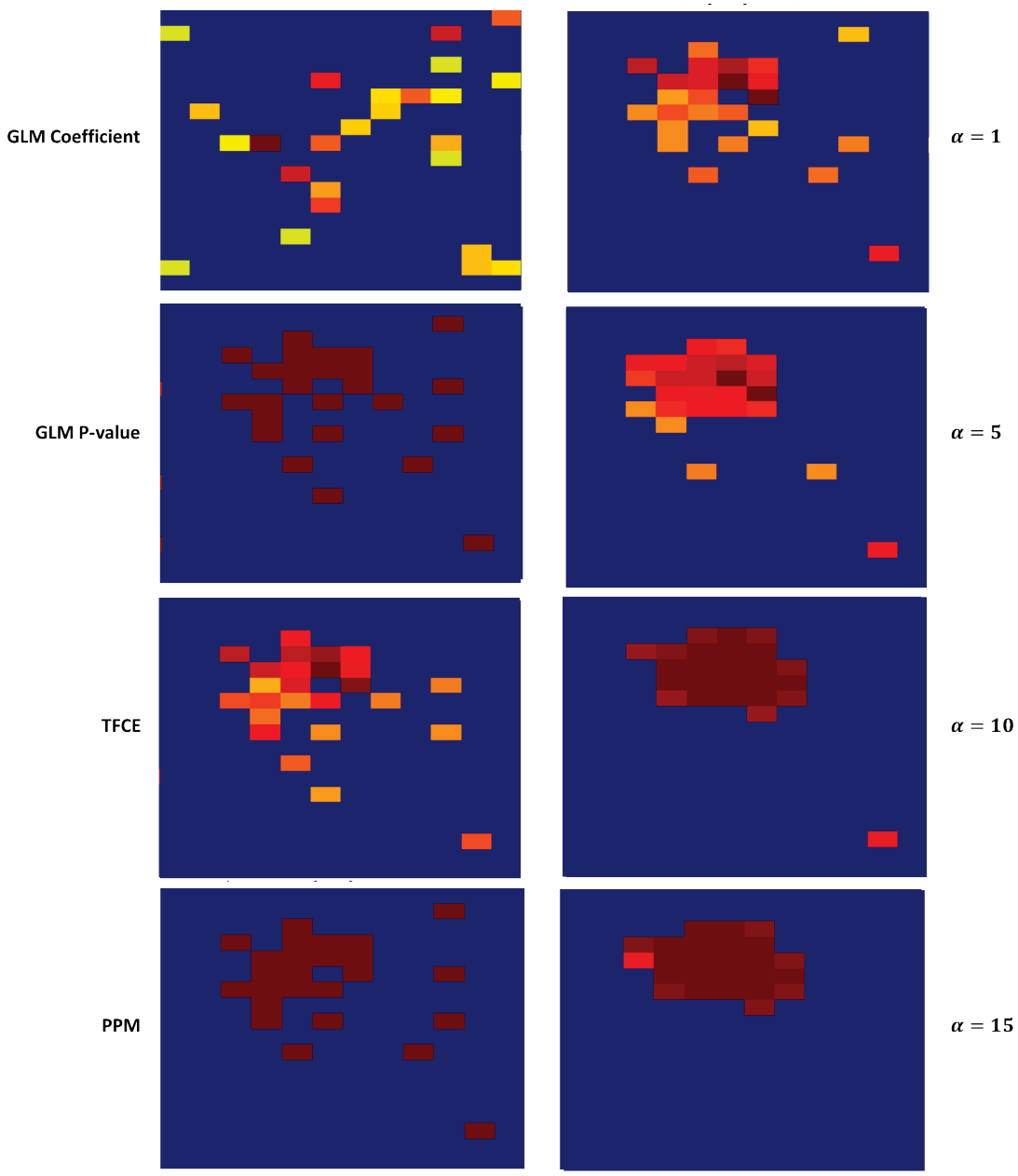


Figure 4-11 Single-subject analysis for Case2 using the task correlated regressor and its derivative

Under all conditions, subplots in the left column of these 4 figures show the results of selecting 25 pixels with the most significant GLM Coefficients, GLM P-values, TFCE-values and PPM-values respectively; and subplots in the right column show the results of selecting top 25 pixels using CR-values with the smoothing parameter chosen from $\{1,5,10,15\}$. Intuitively, we can see that pixels selected in the left column are quite scattered, while some results of our method in the left column with certain values of smoothing parameter contain more locally grouped pixels that are from the black rectangular region in Figure 4-7.

Table 4-1 confirms our observation by calculating the accuracy, specificity sensitivity, AUROC and surface area. From this table we can see that the highest accuracy can be achieved by our method with appropriate values of smoothing parameter. For example, under condition II) when α equals to 10 the corresponding accuracy, specificity, sensitivity, AUROC can be achieved as high as 0.99, 0.96, 0.99 and 0.99 respectively. It is also encouraging to see that the surface area of the results with high accuracy always closes to 20 which is the surface area of the black rectangular area. Therefore, it shows that surface area can act as a metric to determine appropriate range of smoothing parameter. As we can see that the performance of our method maintains with the introduction of the derivative regressor, in the following result part we will only focus on investigating the performance when only the task correlated regressor is used.

Table 4-1 Surface area vs. accuracy, specificity, sensitivity and AUROC in single-subject analysis

	GLM P-value	**GLM Coefficient**	**PPM**	**TFCE**	**α=1**	**α =5**	**α =10**	**α =15**
Surface area (I)	60	60	37	49	58	57	**22**	**22**
Accuracy(I)	0.88	0.88	0.95	0.88	0.88	0.89	0.97	0.97
Specificity(I)	0.48	0.48	0.80	0.48	0.48	0.52	0.88	0.88
Sensitivity(I)	0.93	0.93	0.97	0.93	0.93	0.94	0.98	0.98
AUROC(I)	0.70	0.70	0.98	0.72	0.71	0.84	**0.99**	**0.99**
Surface area (II)	62	78	60	45	62	47	**21**	24
Accuracy(II)	0.89	0.78	0.89	0.93	0.89	0.93	0.99	0.96
Specificity(II)	0.52	0.04	0.52	0.68	0.52	0.68	0.96	0.84
Sensitivity(II)	0.94	0.87	0.94	0.96	0.94	0.96	0.99	0.98
AUROC(II)	0.87	0.53	0.87	0.88	0.88	0.95	**0.99**	0.99
Surface area (III)	56	56	27	43	52	51	31	**19**
Accuracy(III)	0.88	0.88	0.96	0.91	0.88	0.89	0.96	0.99
Specificity(III)	0.48	0.48	0.84	0.6	0.48	0.52	0.84	0.96
Sensitivity(III)	0.93	0.93	0.98	0.95	0.93	0.94	0.98	0.99
AUROC(III)	0.82	0.82	0.97	0.89	0.77	0.86	0.95	**0.99**
Surface area (IV)	55	73	47	45	44	30	23	**19**
Accuracy(IV)	0.90	0.78	0.92	0.92	0.92	0.96	0.95	0.96
Specificity(IV)	0.56	0.04	0.64	0.64	0.64	0.84	0.8	0.84
Sensitivity(IV)	0.94	0.87	0.95	0.95	0.95	0.98	0.97	0.98

AUROC(IV)	0.88	0.61	0.90	0.90	0.87	0.95	0.99	**0.99**

*Values in Bold show the AUROC is high and the surface area is close to 20. Labels (I)-(IV) correspond to results in Figure 4-8, Figure 4-9, Figure 4-10 and Figure 4-11.

The results in Figure 4-8, Figure 4-9, Figure 4-10 and Figure 4-11 are just from one instance of simulated dataset. To test for generalization of these results, we randomly sampled 100 datasets and calculated the distribution of AUROC values. Figure 4-12 shows the boxplots of AUROC values obtained from different methods. We found that the median AUROC of our method with certain smoothing parameters is larger than the median values of other methods (GLM, PPM and TFCE) in this context. For example, in Figure 4-12 (a), when the smoothing parameter is chosen to 10, the median AUROC achieves the highest value of 0.99. We can also observe that the quartile range of our method with smooth parameter of 10 is much smaller than all other methods. It indicates that our method is more stable and robust.

For a multi-subject analysis, we generated 5 samples of time series changes $\boldsymbol{Y}$ using the task correlated regressor in the same way as in the single-subject analysis. We also selected the top 25 pixels to evaluate individual results. Figure 4-13 and Figure 4-14 show the analysis results under different simulation conditions: I) individual sample is generated using the task correlated regressor as described in case 1; II) individual sample is generated using the task correlated regressor as described in case 2.

Subplots in the left column show the GLM P-value results for individual subjects. We can find that results of each subject are significantly different from others and they are quite scattered. Therefore, it is challenge to correctly select top 25 significant features using group level analysis. The first subplot in the right column is from the group GLM P-values. From Figure 4-14, we can see that the group GLM P-value results are not good. In contrast to the group GLM P-values, the results of our method with some certain values of smoothing parameter can more accurately reproduce the black rectangular region (Figure 4-7).

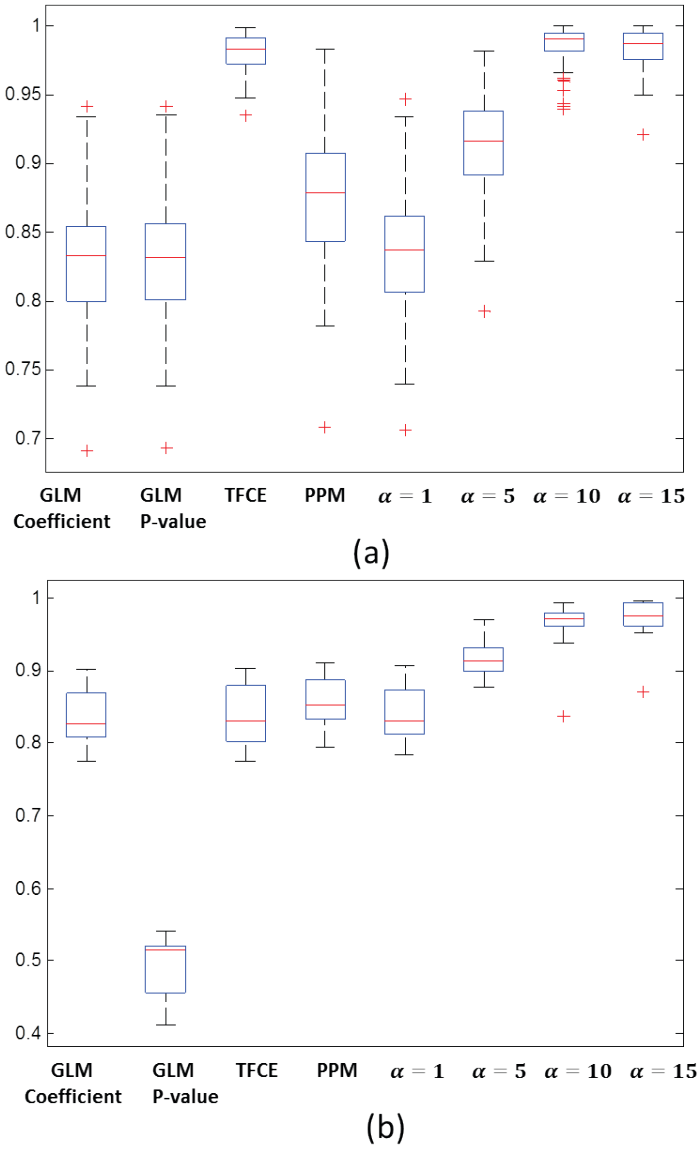


Figure 4-12 The boxplots of AUROC values obtained from 100 samples of simulation data for single-subject analysis. It presents the results of GLM coefficients, GLM P-values, TFCE-values, PPM-values and CR-values with the smoothing parameter α chosen from {1,5,10,15}. (a) The results are for Case1 using the task correlated regressor; (b) The results are for Case2 using the task correlated regressor.

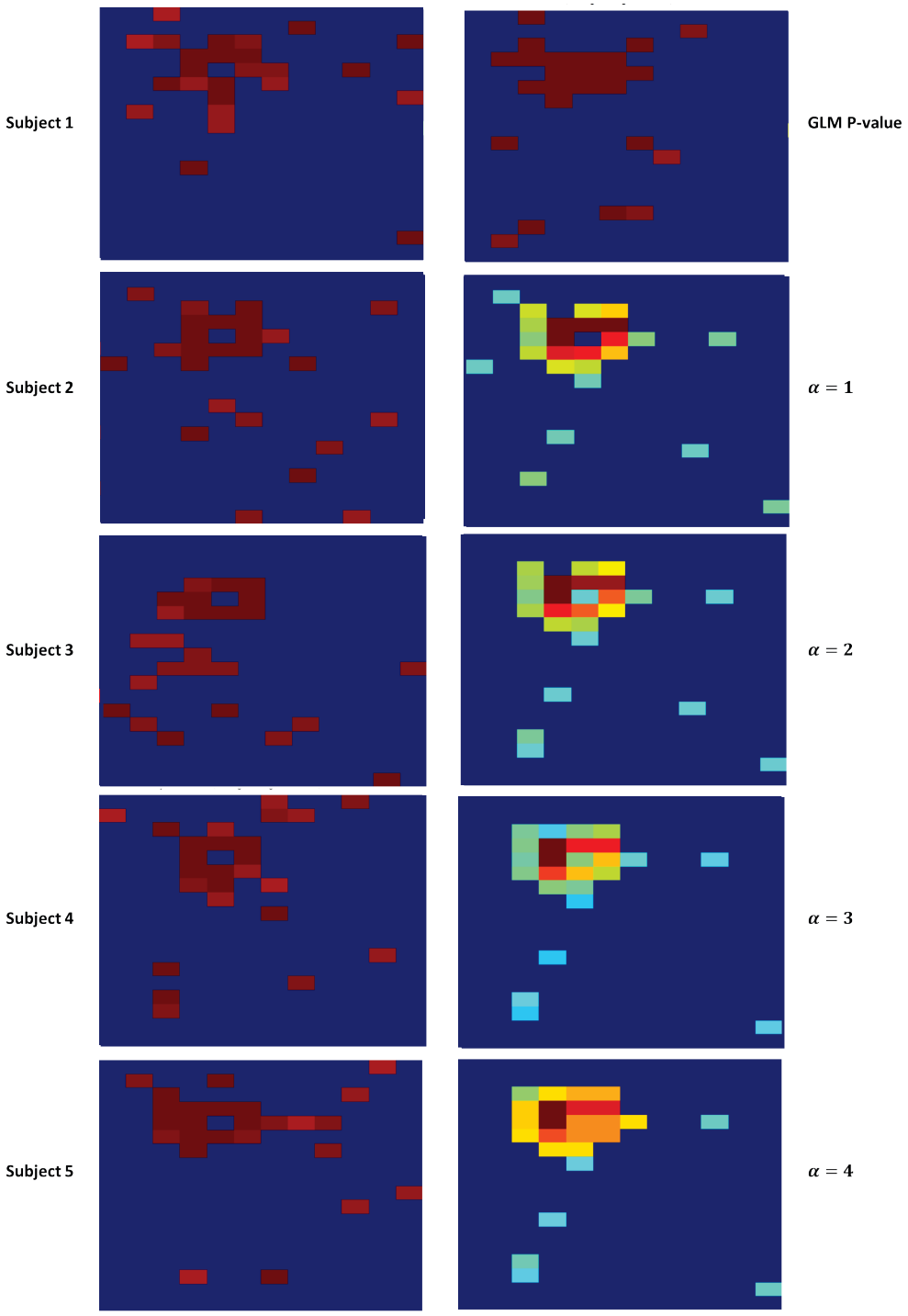


Figure 4-13 Multi-subject analysis for Case1 using the task correlated regressor.

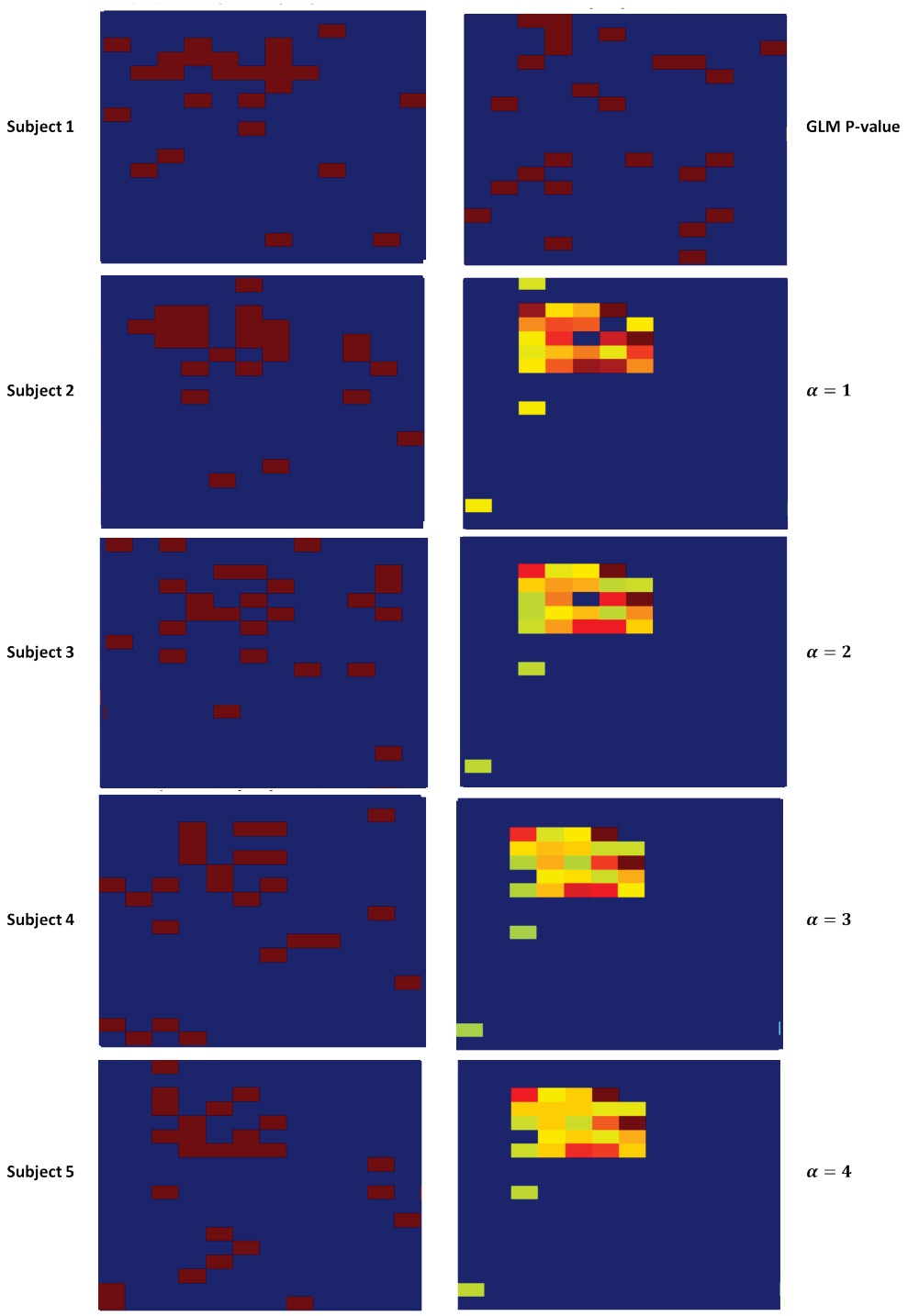


Figure 4-14 Multi-subject analysis for Case2 using the task correlated regressor.

Table 4-2 further demonstrates this observation quantitatively. For example under condition II, we can see that the group GLM P-values can only give the AUROC value of 0.59, while the results from CR-value with smoothing parameter equal to 4 have AUROC as high as 0.99. It shows that our method works much better than traditional method. It is also encouraging to observe from Table 4-2 that results with surface area closing to the ground truth are more

accurate. Therefore, it further confirms that surface area can be a way to choose the appropriate value of smooth parameter.

Table 4-2 Surface area vs. accuracy, specificity, sensitivity and AUROC in multi-subject analysis.

	GLM P-value	α =1	α =2	α =3	α =4
Surface area (I)	47	44	39	**35**	**35**
Accuracy (I)	0.91	0.93	0.94	0.94	0.94
Specificity (I)	0.60	0.68	0.72	0.76	0.76
Sensitivity (I)	0.95	0.96	0.96	0.97	0.97
AUROC(I)	0.93	0.89	0.91	0.93	**0.97**
Surface area (II)	73	32	28	**27**	**27**
Accuracy (II)	0.81	0.97	0.98	0.98	0.98
Specificity (II)	0.16	0.88	0.92	0.92	0.92
Sensitivity (II)	0.89	0.98	0.99	0.99	0.99
AUROC(II)	0.59	0.99	0.99	**0.99**	**0.99**

*****Values in Bold show the AUROC value is high and the surface area is close to 20. Label (I) and (II) correspond to results in Figure 4-13 and Figure 4-14.

4.4.2 Empirical study

4.4.2.1 Data acquisition

The real fMRI dataset we use in this paper is taken from the study by (Tomassini et al. 2012). 23 MS patients and 12 healthy volunteers were studied with fMRI during a visuo-motor tracking task performed with the right hand. After 2 weeks of daily practise, the study was repeated. Here we wish only to demonstrate our method rather than study the differences between healthy volunteers and patients, so we analyse the dataset that only contains healthy controls. fMRI data acquisition was performed on 3-T Varian Inova MRI system (Oxford, UK) using multi-slice gradient-echo EPI sequence [repetition time (TR) = 3000 ms; echo time (TE) = 30 ms; 43 axial slices (3.5-mm thick) providing whole-brain coverage; field of view (FOV), 256 × 192 mm2; matrix 64 × 64]. Data preprocessing is carried out by tools from FSL (Smith et al. 2004), which includes motion correction, brain extraction, spatial smoothing, high-pass temporal filtering, correction for field inhomogeneities, and nonlinear registration from high-resolution T1 structural to MNI standard space. As this real fMRI dataset is only used to demonstrate our method, we reduced its dimensionality to save

computational time. The trilinear interpolation method was used here to down sample the dataset with the voxel size of 8mm in all three directions.

Afterwards, we apply our method to find changes in functional brain regions by comparing the fMRI measurements between two different sessions. We first get images obtained from two separate sessions for each subject. Then we work on the normalized fMRI measurements. The design matrix is same as the one used in the simulation study, reflecting the real dynamics of external motor practice. To save effort, in this study we show the results when only the task correlated regressor is used. In single-subject analysis, the step size of searching λ is set to 1; while in multi-subject analysis, the step size of searching λ is set to 5.

4.4.2.2 Results

We first do single-subject analysis that one subject is chosen to demonstrate our method. Figure 4-15 compares the results of applying GLM and our method with a range of values of the smoothing parameter. The map of the regression coefficient in the second row of Figure 4-15 defines the strength of the relation but not the significance between voxel and design matrix. Therefore the results of GLM coefficients are different from other subplots.

The GLM P-values are displayed in the third row of Figure 4-15 as a logarithmic transformation. Subplots from the fourth row to the last row show the scaled CR-values (10 to the power of standardized CR-value) of our method with the smoothing parameter chosen from $\{1,5,10,15\}$. We can find that with smaller the smoothing parameter, results from our method tend to approximate results generated by GLM P-values. When we add larger smoothness constraints, the values of neighbourhood voxels tend to be closer. As expected, we can find some voxels are consistently more significant than others across different smoothness constraints.

Figure 4-16 shows the histograms of CR-values with different smoothing parameter values. Although Figure 4-15 shows that relative significance of voxels vary with different values of smoothing parameter, the distributions of standardized CR-values, displayed in Figure 4-16, are stable. This means that different smoothness constraints do not have obvious influence on most voxels. That is, only a small fraction of voxels are affected by the smoothing parameter. These voxels originally ignored by GLM P-value results may be potentially meaningful as they are close the some nearby significant ones.

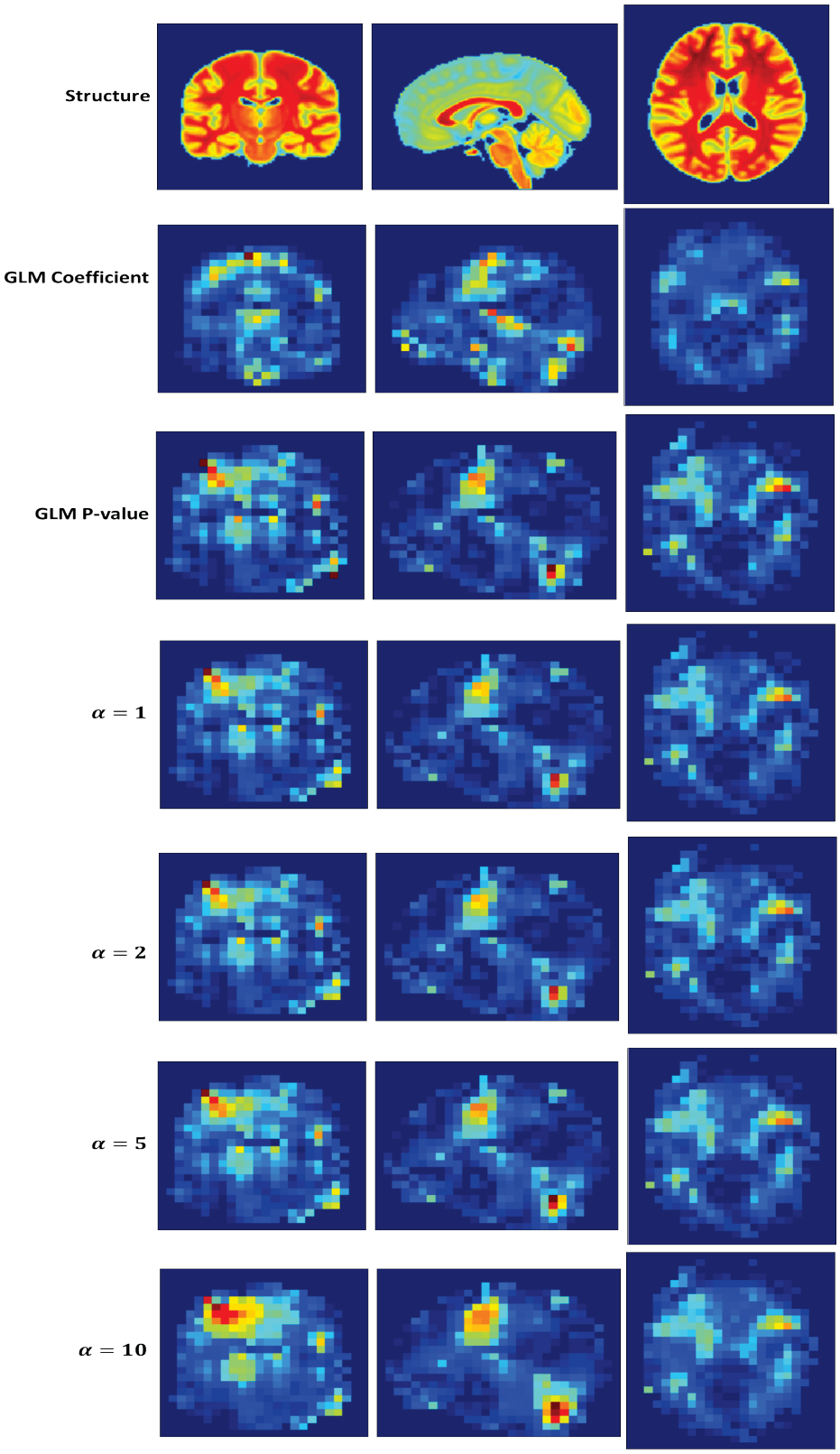


Figure 4-15 Results of single-subject analysis using coefficients, GLM P-values and CR-values. The smoothing parameter α varies from 1 to 10 (we only show some selected figures). For better visualization, a logarithm transformation is performed on GLM P-value results, while CR-value results are scaled by 10 to the power of standardized value.

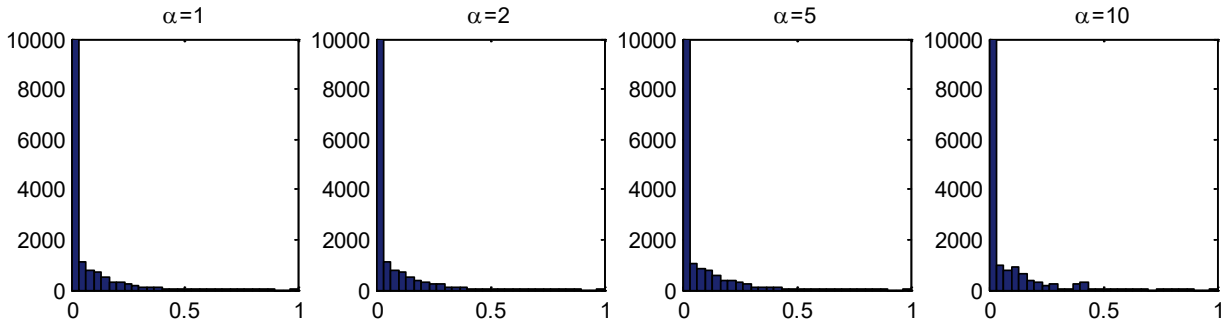


Figure 4-16 Histograms of CR-values in single-subject analysis. The smoothing parameter α is chosen from 1, 2 , 5 and 10. CR-values are normalized into the interval between 0 and 1.

As we have shown in the simulation results, surface areas can be used to choose the most appropriate values of the smoothing parameter. Table 4-3 shows the surface areas of the results generated with smoothing parameter ranging from 1 to 10. Permutation tests are first used to generate P-values for CR-value results. We then select top significant voxels whose P-values exceed the target significance level of 0.05. In Table 4-3, we can observe the general trend of decreasing surface area with increasing value of the smoothing parameter; a stronger smoothing constraint forces significant voxels to be clustered more closely. The optimal value of smoothing parameter is determined by comparing the degree of smoothness between the resulting image and prior knowledge. More specifically, if we have the prior knowledge that the surface area of the resulting image is close to 3700, then we can select the smoothing parameter to be 6 by looking up Table 4-3.

Table 4-3 Surface areas of single-subject analysis with the smoothing parameter varies from 1 to 10. The top significant voxels are selected according to the P-value of CR-value results generated by permutation tests, where the threshold of significance level is 0.05.

α	1	2	3	4	5	6	7	8	9	10
Surface area	3975	3961	3887	3871	3793	3722	3673	3608	3556	3323

We extend this demonstration to show the ability of using grouped CR-values to find significant voxels across all samples. As we have already investigated the influence introduced by different values of smoothing parameter and also the way to choose appropriate ones in the single-subject analysis part, here we focus on comparing the group

GLM P-value results with our results when the smoothing parameter is fixed to 2. Subplots in the first 6 rows of Figure 4-17 show the results of GLM P-values for 12 individual subjects. We can see that significant voxels for different subjects vary significantly, bringing in challenges of finding common significant ones. The results of both group GLM P-values and group CR-values are shown in the last row of Figure 4-17. We can observe that these two subplots are different. As we do not know the ground truth, we cannot simply show our method works better with higher accuracy as we did in the simulation section. However, we use Figure 4-18 to demonstrate the potential advantage of our method relative to the group GLM P-values.

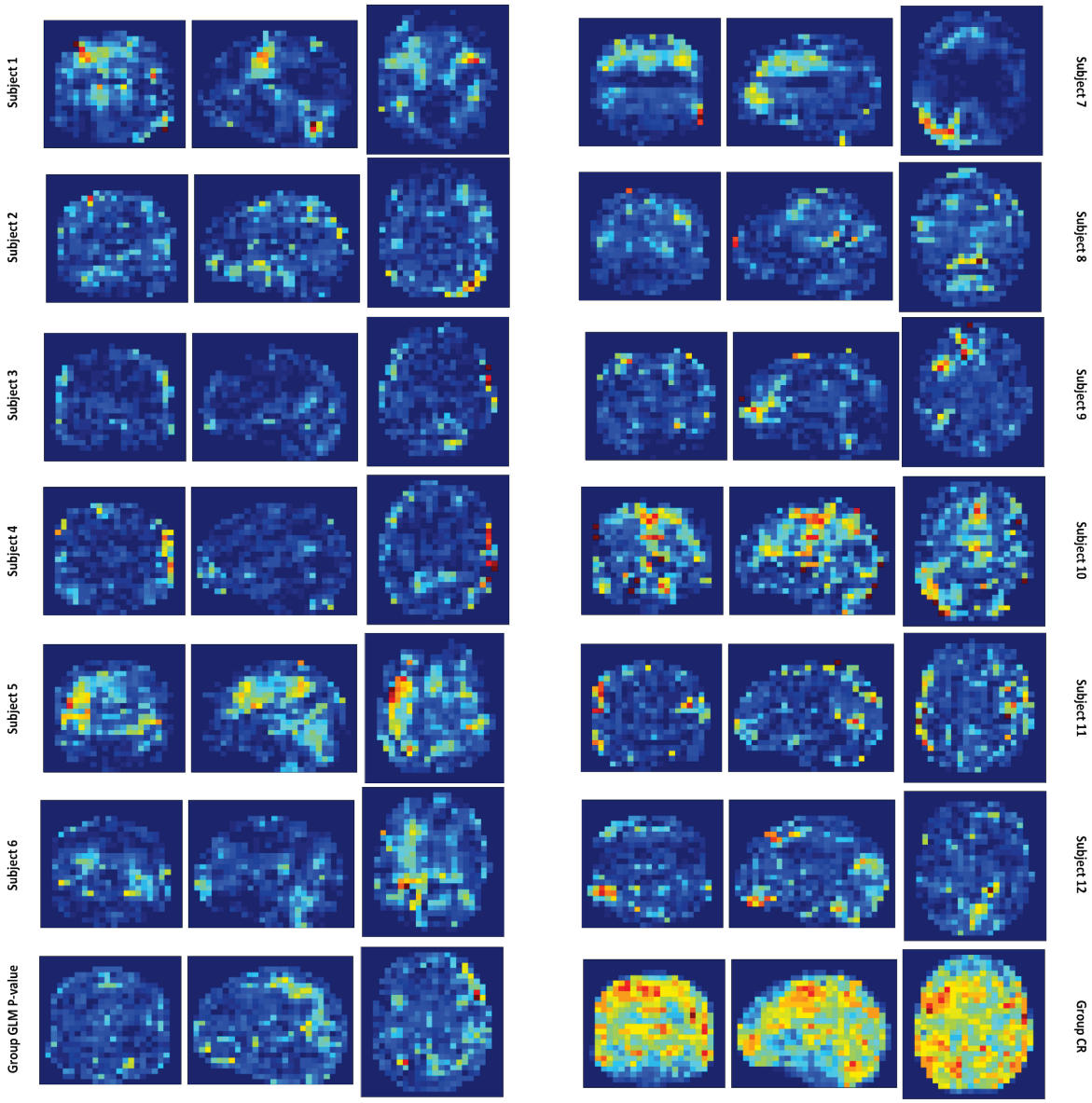


Figure 4-17 Multi-subject analysis using group GLM P-values and CR-values. The results of GLM p-value for individual subjects are shown in the first 6 rows. The last row contains

results of group GLM P-values and CR-values with smoothing parameter α of 2. For better visualization, a logarithm transformation is performed on the group GLM P-value results, while the group CR-value results are scaled by 10 to the power of standardized value.

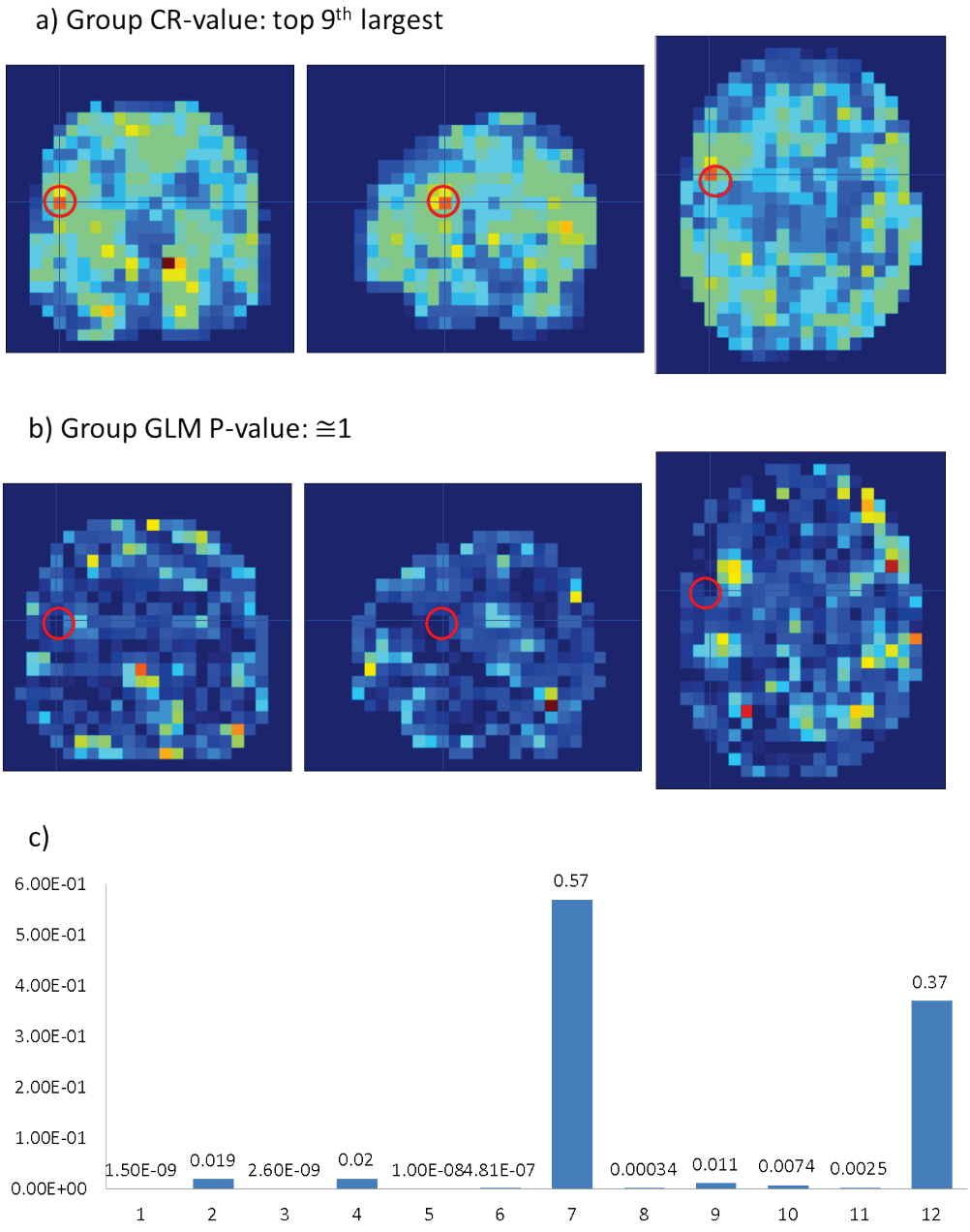


Figure 4-18 An example of differences in group CR-values and group GLM P-values. (a) One highly ranked voxel according to group CR-value. (b) Its corresponding group GLM P-value. (c) GLM P-values of each sample for this voxel.

In Figure 4-18(a), we can find a significant voxel whose group CR-value is ranked as the 9th largest. However, we can see its corresponding P-value from group GLM in Figure 4-18(b),

which approaches to 1, even though 10 out of 12 samples at this voxel are individually highly significant (shown in Figure 4-18(c)). In this instance, it is the other two extremely non-significant ones averaging out the overall significance. We have checked whether this observation is a consequence of smoothness constraint dominating the results. The significance level of this voxel is investigated when no smooth constraint is added. It is interesting to see that the grouped CR-value of this voxel still ranks high. Thus it is the nature of the grouped CR-value method not simply the introduction of smoothness constraint successfully overcoming the problem that some insignificant voxels dominate the group GLM P-values.

4.5 Discussion

CR-values are obtained by changing the value of the parameter λ. By gradually increasing the parameter λ in predefined steps, the CR-value of a certain voxel is recorded as value of the parameter λ, with which the coefficient of this voxel becomes zero. Performed in this way, CR-values are influenced by the choice of size of these step changes; smaller steps will enable estimates of CR-values closer to the real changing point where a voxel's coefficient becomes zero. There is an alternative approach to find CR-values without defining fixed λ step size. It is Least angle Regression (LARS) (Efron et al. 2004) that we described in subsection 3.2.2.3. LARS can sequentially select variables to enter into the linear model, where the selection order reflects the significance of variables. As it has been discussed in (Efron et al. 2004), a minor modification of LARS can bring Lasso estimates. In the future, we can consider detecting CR-values via LARS.

Significant test for Lasso models is generally based on resampling or data splitting (Meinshausen et al. 2008; Wasserman & Roeder 2009; Minnier et al. 2011). Although a recent paper directly introduces a test statistic without resampling, it has some limitations (Lockhart et al. 2014). For example, it is based on many assumptions and does not provide rigorous theory to support the case when number of features is larger than samples. Therefore, in this chapter, we choose the resampling based method. P-values for given CR-values are calculated via permutation test.

The target function that the CR-value method minimizes contains two parameters. In practice, we have allowed the parameter λ to vary with the parameter α being fixed. Although cross-validation is a common way to determine parameters, we do not use it to determine α. As it has been pointed out in (Murphy 2012), cross-validation tends to maximize the predictive

accuracy of regression models rather than recover a true model able to generate the observational data. Cross-validation can overfit the noise. More importantly, in our method, the parameter λ is changing and hence cross-validation cannot be used. Therefore, we have used the metric of surface areas and choose values of α that give "reasonable" values of surface areas based on prior knowledge of the regional activations expected.

In our analysis, we assume that the noise is white Gaussian. Although fMRI noises are usually temporally correlated, some preprocessing steps can be used to 'white' them. For example, the prewhitening method removes the temporal correlation from the GLM prior to estimation. It first runs a regression on the data to obtain the model residuals, from which we can obtain the autocorrelation structure. Then model estimation is carried out after prewhitening both the data and the design matrix. Popular fMRI analysis tools, such as SPM (Penny et al. 2006) and FSL (Jenkinson et al. 2012), have their own schemes to prewhiten the noise. Therefore, our methods directly work on the white noise, assuming autocorrelation has been removed beforehand.

One limitation of the proposed method is its requirement of prior knowledge on smoothness. If we have no prior knowledge of the expected regional activations, then we cannot directly determine the optimal smoothing parameter. In this case, we need to generate different results with different values of smoothing parameter. Then by investigating different results, such as their selected functional brain regions, we can decide plausible values of the smoothing parameter.

In the future, we can further extend our model by adding more terms. As we know, one limitation of Lasso is that it tends to select one variable from a group of highly correlated ones and ignore the others. We can add a ℓ_2 term to avoid this problem and correlated voxels will be simultaneously selected. Then our model will contain ℓ_1 and ℓ_2 regularization terms as well as the smoothing term. It can be regarded as an extension of elastic net (Zou & Hastie 2005).

5 Grouping measurements: a clustering based approach

5.1 Motivation

In section 3.3, we have briefly introduced methods to identify groups of quantitative dataset. Samples are grouped in a way that data points in a single group are more similar than others in other groups. This chapter focuses on the clustering based method of grouping items whose quantitative measurements are similar.

In datafied life research, a massive amount of large-scale datasets are generated by high throughput instruments. For example, mass spectrometry (MS) machines can be used to quantitatively show the Proteomics, Lipidomics and Metabolic profiles of biological samples. Proteins, lipids and metabolites can be segmented into small compounds; and the chemical and physical characteristics of their fragments are quantified in the machine. Mass spectrometry machines record measurements of samples in individual files. Each file contains a list of fragments, where each fragment is specified by quantitative values from measurement. For example, a fragment can be quantified by three variables: m/z, retention time and drift time. Its abundance is saved in the variable named as intensity. Figure 5-1 shows an example plot of fragments in a sample file.

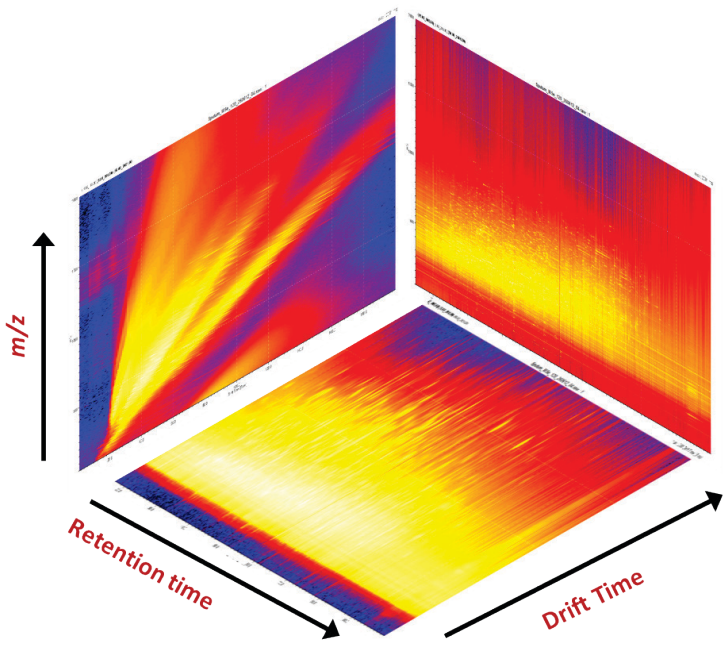


Figure 5-1 An example plot of fragments in a sample file in three measurement directions. The colour indicates value of intensity: a fragment with a larger intensity value has a brighter colour.

As we have discussed in section 2.7, a typical research question is finding significant difference between two sample groups. Figure 5-2 shows an example that different samples have peaks (fragments with high intensity) appearing at similar m/z values. To compare the intensity of fragments across samples, we first need to group similar fragments from different samples using their quantitative measurements. This is for the case that the identities of fragments are not known. Due to the variations of instruments and experimental environments, the quantitative measurements of fragments referring to the identity may be slightly different for different samples.

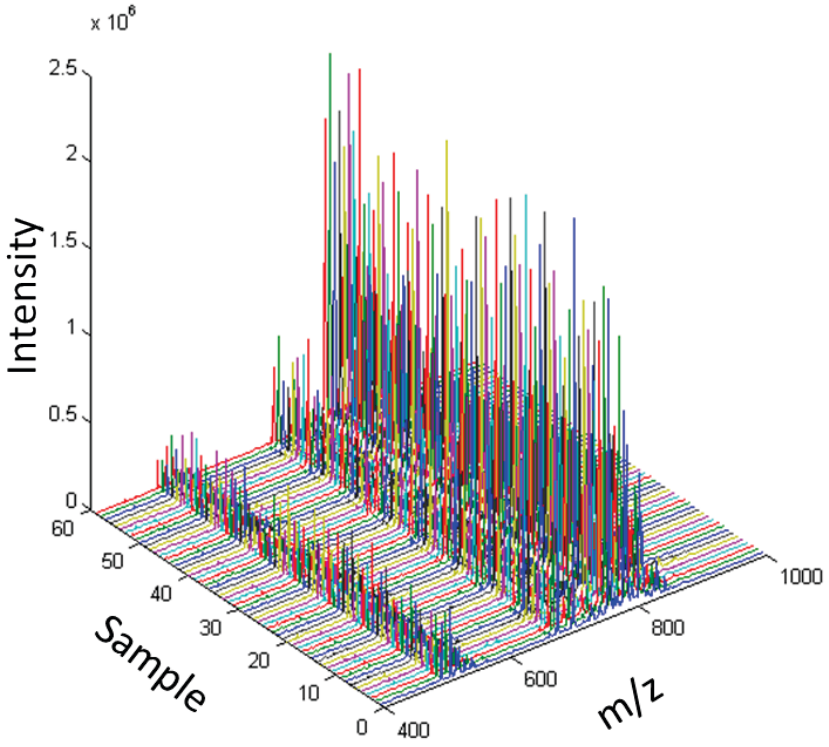


Figure 5-2 Intensities of fragments in different samples along the m/z direction.

One popular way to group similar fragments is discretizing the measurements with a predefined resolution. For example, UBIOPRED Lipidomics study initially attempts to choose the discretization size of 0.1m/z to discretize the measurements in m/z direction. After discretization, we can simply find similar fragments by choosing ones whose discretized measurements are same. This approach is quite straightforward. However, it distorts the original information of the dataset. The results are highly dependent on the predefined resolution.

Another way is doing alignment to correct drifts in measurements across samples. For example, in (Smith et al. 2006), a local regression fitting method, loess (Cleveland & Loader 2001) is used to eliminate deviations in the retention time direction. However, the alignment is not yet used for other dimensions, such as m/z.

The aim of grouping similar fragments can be depicted as Figure 5-3. Fragments are plotted in a 3D space. The aim of grouping similar fragments is finding boxes that contain fragments from different samples. It comes into an idea of applying clustering methods to solve this problem.

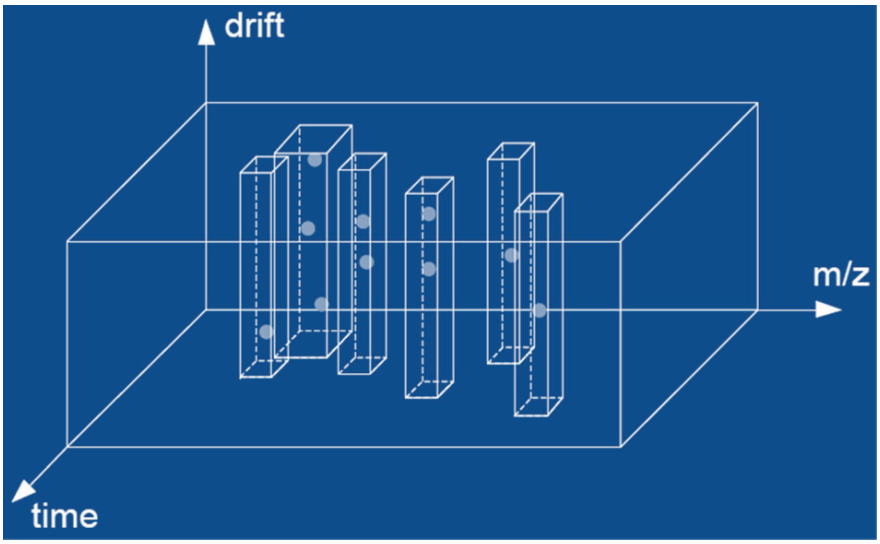


Figure 5-3 An illustration of grouping similar fragments.

In this chapter we propose a clustering based method to match similar ions across samples for the MS Proteomics datasets. It can deal with large dataset and also support parallel implementation. The details will be introduced in the following sections. This work is motivated by the UBIOPRED project. As we have discussed in Chapter 2, this project aims to using various Omics profiles to subtype severe asthmatics. Large amount of datasets are generated by MS machines to quantify Proteomics and Lipidomics profiles. In order to maximally extract information from these datasets, we develop our own method. In this chapter, we only focus on introducing the method for Proteomics datasets. The method developed for Lipidomics datasets is not shown as it is quite similar to the one for Proteomics dataset. In the following parts, we will first introduce the background of large scale Proteomics analysis. Then we will introduce our method and show its performance using experimentally designed datasets.

5.2 Background

Proteomics, based around modern mass spectrometry, is a powerful approach with which to study changes in protein abundance profiles in tissues or cells, resulting from disease or environmental perturbation (Aebersold & Mann 2003). As such, it is a useful resource in biomarker discovery, disease characterisation and drug discovery programs. The main analytical approach in Proteomics is liquid chromatography tandem mass spectrometry (LC-MS/MS), a process which facilitates the separation, measurement and fragmentation of ionized peptides generated through proteolytic cleavage. The traditional and most widely adopted approach in MS for Proteomics involves data-dependent acquisition (DDA) of tandem mass spectra, where a survey scan gathers information on intact precursor ions followed by the selection of a subset of ions with specific characteristics for fragmentation via collision induced dissociation (CID).

While this mode of operation has been extremely successfully in its application to many cell types and tissues, it is inherently biased towards high abundance peptides during precursor selection (Geromanos et al. 2009) and the semi-stochastic nature of precursor selection means that these types of approaches also suffer from issues of missing data between experimental replicates which impacts data analysis downstream. Additionally, as the method of ion selection in DDA does not facilitate sampling of ions across their entire chromatographic peak, quantitation can only be performed on a comparative basis, relying on mass tagging, labelled amino acid incorporation, or pseudo-quantitation based on spectral counting. There are various technical issues with each of these methods and this area has been extensively reviewed (see (Neilson et al. 2011) and (Ow et al. 2009) and (Schulze & Usadel 2010) for examples). Back-end processing of this data with finite genome-centric databases of in silico predicted peptide information adds another level of bias, as only entities contained within that database and passing a strict scoring scheme can be identified. As such, unusually fragmented peptides, peptide modifications and isoform variations are not identified (Sadygov et al. 2004) unless these variable modifications are included in the search parameters, which rapidly expands the false discovery rate (FDR) for every modification considered .

Many of the above approaches are also not generally applicable to large scale studies (Levin et al. 2007; Lucas et al. 2012) which are required in modern biomarker discovery and validation programs. A paradigm that is illustrated by the fact that almost all published biomarkers reported from small scale discovery programs have not been adopted in clinical settings (Drucker & Krapfenbauer 2013). Additionally, reporting guidelines now require

sample size power calculations and rationale for treatment of missing data to be presented and justified (McShane et al. 2006; von Elm et al. 2008; Skates et al. 2013), alongside internal and external validation of prediction models (Collins et al. 2015).

Recent developments in both Liquid Chromatography (LC) and MS instrumentation have led to the development of data independent methods including SWATH and the high definition ($HDMS^E$) approach (Silva et al. 2005), where all ions are fragmented. Thus they potentially deliver information on every amenable molecule introduced into the mass spectrometer and sample these molecules across their entire chromatographic peak width. The incorporation of travelling wave ion mobility (TWIM) as an orthogonal separation in this method, has been shown to increase proteome coverage and identification confidence, particularly for low abundance proteins, without any increase in instrument time. However, such improvements have come at a small cost of sensitivity at due to ion loss, and can encounter detection saturation effects for very high abundance peptides as a result of ion packet concentration in TWIM (Shliaha et al. 2013). To address the complex nature of the data produced in $HDMS^E$ which contains information pertaining to molecular mass, behaviour in liquid chromatography, cross sectional area for precursors, and corresponding information rich 'all-ion' fragmentation data, complex data processing procedures have had to be developed (Silva et al. 2006; Li et al. 2009; Silva et al. 2005) in order to interpret the highly complex spectral information gathered and to address the effects of the technique on ion behaviour as described above (Bond et al. 2013) .

ProteinLynx Global Server (PLGS) (Silva et al. 2005) produces curated ion information from Ion mobility MS^E experiments. This ion information contains precursor characteristic information which we refer to as DRAMI tags (Drift time, Retention time, Accurate Mass, Intensity). Through peak apex time alignment, fragment ions are associated with each precursor peak, and these can be utilized in a complex database searching and scoring procedure within PLGS producing high quality peptide information files (Ion accounting) (Li et al. 2009). Estimation of absolute quantitation of identified peptides can also be achieved through comparison with internal standards that have been introduced at known concentrations via the Hi3 method. Quantitation by this means utilizes the relationship between MS signal response and protein concentration, specifically: the average signal response for the three most intense tryptic peptides is constant per mole of protein. By utilizing an internal standard at known concentration, a universal response factor can be calculated and subsequently used to estimate absolute quantities of proteins in a sample

(Silva et al. 2006). While this method has proven very useful in identifying and quantifying peptides across dynamic abundance ranges and in multiple sources (recent examples include: (Burniston et al. 2014) and (Wegdam et al. 2014)), the data is still biased by database searching issue as described above.

To address this challenge, we have developed a method that makes use of all ion information gathered in the mass spectrometer operating in data independent mode, while using high quality identified data as a reference and for evaluation. We assume that through the use of accurate mass (at >20000 resolution with accuracy aided by the use of a lock-mass channel), Travelling wave Ion mobility drift time (which is inherently stable), and high resolution Nano UPLC, a single DRAMI uniquely identifies any given chemical entity. We also assume that when analysing samples with high resolution DIA, these chemical entities will be sampled in every experiment when present at levels above the limit of detection. Central to this method is accurately matching DRAMI ions across large numbers of experiments of highly complex samples. The following part of this chapter will describe our matching method in details. The performance of our method will be evaluated by high resolution MS^E datasets.

5.3 Method

Raw data files are processed to DRAMI using PLGS. For each sample, PLGS APEX 3D file contains information of all ions, including accurate mass (MZ), drift time (DT) and retention time (RT). PLGS ion accounting file contains peptides that discovered in each sample. Our matching method works on APEX 3D files to group similar DRAMI ions across samples together using MZ, DT and RT information. A retention time alignment step, based on the locally weighted scatterplot smoothing (Lowess) method (Cleveland 1979), is carried out first to correct RT deviations. This step is developed by our collaborator Southampton Proteomics group (leaded by Paul Skipp) and the details are not shown in this chapter. The following parts focus on introducing the details of the matching algorithm.

5.3.1 Data binning

As the data size is quite large, in order to facilitating the ion matching step using hierarchical clustering, it is necessary to split the whole datasets into pieces. Therefore, datasets are first binned by MZ (as this is measured with the highest accuracy) by splitting each file into slices of a fraction of a mass unit (Smith et al. 2006). The MZ bin size is estimated by calculating: 2 times the 95^{th} percentile of value of the measured full width at half maximum (FWHM), which can be obtained from the column value ‘ion_msFWHM’ in PLGS APEX 3D files. In

order to avoid any boundary effect due to arbitrary splitting along the MZ axis, overlapping bins are used in which adjacent bins overlap by half. For example, if we choose the MZ bin size, bin_{MZ}, to be 0.05 *m/z*, then examples of adjacent bins are 100~100.05m/z, 100.025~100.075m/z, 100.05~100.1m/z, etc. Thus, each ion appears twice in two neighbouring bins. A duplicate peak removal scheme is described in subsection 5.3.3.

5.3.2 Ion matching within each bin

Following the binning of ions from all samples, ions are matched within each bin. The pairwise distance d_{ij} between any two ions I_i and I_j is estimated using Euclidean distance. A scaling step is first performed for MZ, RT and DT using the RT difference (Lange et al. 2007) in order to make these comparable. If we let Dif_{mz} represent the maximum difference in MZ between any two ions from the whole dataset then Dif_{rt} and Dif_{dt} are defined in the same way. The weighted difference between ion I_i and I_j is calculated by:

$$d_{ij} = \sqrt{(rt_i - rt_j)^2 + \left(\frac{Dif_{rt}}{Dif_{dt}}(dt_i - dt_j)\right)^2 + \left(\frac{Dif_{rt}}{Dif_{mz}}(mz_i - mz_j)\right)^2} \quad 5.1$$

Their corresponding distances in MZ, RT and DT are calculated by

$$\begin{aligned} d_{ij}^{mz} &= \sqrt{(mz_i - mz_j)^2} \\ d_{ij}^{rt} &= \sqrt{(rt_i - rt_j)^2} \\ d_{ij}^{dt} &= \sqrt{(dt_i - dt_j)^2} \end{aligned} \quad 5.2$$

We then define the thresholds in RT, DT and MZ dimensions, assuming that ions with difference in any of these three dimensions (3D) larger than the corresponding threshold do not represent the same chemical entity. The RT threshold thr_{RT}, is estimated from the chromatographic properties by calculating the 95th percentile of the difference between the peak beginning (given by 'atInflectUpRT' in PLGS APEX3D files) and end (given by 'atInflectDownRT' in PLGS APEX3D files); while the DT threshold thr_{DT} and the MZ threshold thr_{MZ}, are defined as the 95th percentile of the drift and MZ FWHM respectively (given by 'ion_driftFWHM' and 'ion_msFWHM' in PLGS APEX3D files). The distance d_{ij} between two ions, whose $d_{ij}^{mz} > thr_{MZ}$ or $d_{ij}^{rt} > thr_{RT}$ or $d_{ij}^{dt} > thr_{DT}$, is set to be infinite such that ions with d_{ij}^{mz} or d_{ij}^{rt} or d_{ij}^{dt} larger than the threshold will not be grouped together.

Moreover, to avoid grouping ions originating from the same sample together, the distance between any two ions from the same experimental sample is also assigned to infinity. After

the distance values are calculated, an average hierarchical clustering analysis is performed (Olson 1995). The average linkage function specifies the distance between two clusters by

$$D(A,B) = \frac{1}{N_A \times N_B} \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} d_{a_i,b_j} \quad 5.3$$

where A and B are two clusters containing a number of N_A ions a_i and N_B ions b_j respectively. To obtain the matching results, the hierarchical clustering is cut with a designed threshold, thr. As shown in Equation 5.1, the distance in all three dimensions is scaled towards RT, thus the maximum distance between two similar ions potentially having same identity should not exceed the threshold of

$$thr = \sqrt{(thr_{RT})^2 + (thr_{RT})^2 + (thr_{RT})^2} = 3^{\frac{1}{2}} * thr_{RT} \quad 5.4$$

We prune branches off the bottom of the hierarchical tree and assign all the objects below each cut to a single cluster. Each resulting cluster (i.e. ion group) contains a set of ions that have similar characteristics and potentially refer to the same chemical entity.

The process of ion matching within each bin is summarized in Algorithm 5.1. Additionally, when the number of ions in a MZ bin is very large, the binning process will not only be carried out in the MZ direction but also the DT direction. In this case, the number of ions within each bin decreases, and hence the hierarchical clustering based matching algorithm runs faster. The binning process in DT direction is same as it is in MZ direction.

Algorithm 5.1 The ion matching within each bin

Input Ions from all samples, threshold thr, thr_{MZ}, thr_{RT} and thr_{DT}

Output Ion groups, denoted as IonGroup, containing similar ions across samples

// calculate the distance matrix of any pairs of ion

1 **for** any two pairs of ion I_i and I_j

2 $d_{ij}^{mz} \leftarrow \sqrt{(mz_i - mz_j)^2}$

3 $d_{ij}^{rt} \leftarrow \sqrt{(rt_i - rt_j)^2}$

4 $d_{ij}^{dt} \leftarrow \sqrt{(dt_i - dt_j)^2}$

5 **if** $d_{ij}^{mz} > thr_{MZ}$ or $d_{ij}^{rt} > thr_{RT}$ or $d_{ij}^{dt} > thr_{DT}$

6 $d_{ij} \leftarrow \infty$

7 **else if** ion I_i and I_j are from the same sample

8 $d_{ij} \leftarrow \infty$

9 **else**

10 $d_{ij} \leftarrow$

$$\sqrt{(rt_i - rt_j)^2 + \left(\frac{Dif_{rt}}{Dif_{dt}}(dt_i - dt_j)\right)^2 + \left(\frac{Dif_{rt}}{Dif_{mz}}(mz_i - mz_j)\right)^2}$$

11 **end**

12 **end**

13 Form the distance matrix D from d_{ij}

// Hierarchical clustering using average linkage

14 $\text{ClusterTree} \leftarrow linkage(\text{D}, 'average')$

// generate ion groups

15 $\text{IonGroup} \leftarrow cluster(\text{ClusterTree}, 'cutoff', thr)$

5.3.3 Adjustment between overlapping bins

Following the matching of ions in each bin, an adjustment method is proposed to remove duplicate ions appearing in neighbouring bins. An example of matching results of three consecutive bins, bin_0, bin_1 and bin_2 is shown in Figure 5-4(a), where T_{mz} is the MZ cutoff point separating bin_0 and bin_2. From Figure 5-4(a), we can see that a black ion group in bin_1 is separated into to two individual ion groups in bin_0 and bin_2. The separation is resulted from the binning process. The adjustment algorithm is used to add the black ion group of bin_1 to bin_0 and remove the original black ion groups in bin_0 and bin_2 (see expected results in Figure 5-4(b)). The adjusted matching results in bin_0 are saved to the final matching results. This example only shows a simple case of adjustment. In reality, this tends to be more complicated when more than one ion group lying across the splitting boundaries. The process of integrating matching results of overlapping bins is shown in Algorithm 5.2.

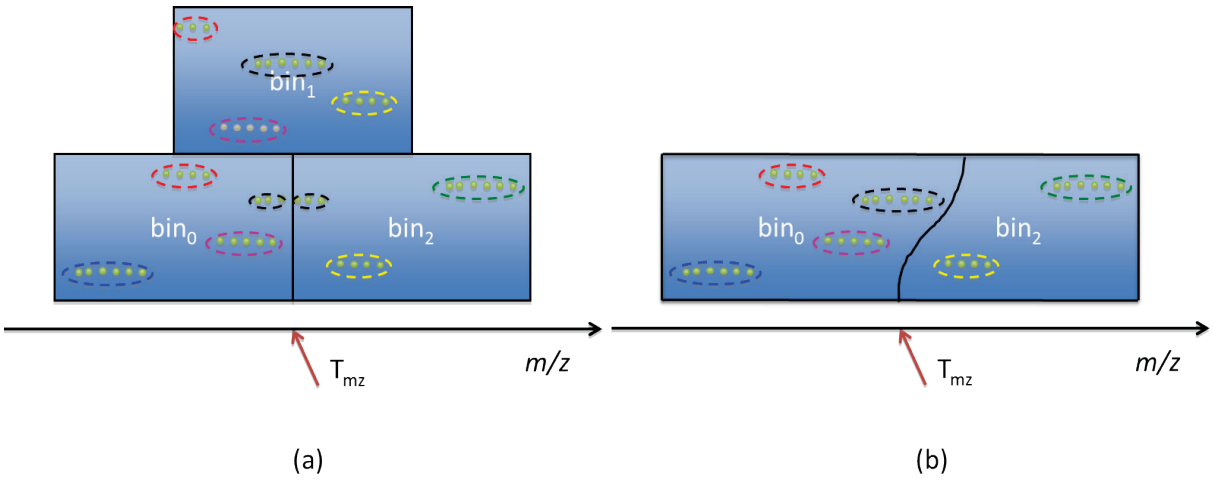


Figure 5-4 An example to show the adjustment method. (a) Matching results of 3 consecutive bins. In bin_1 there is a ion group in black having ions with MZ values across the boundary T_{mz}. This ion group is split into two groups in bin_0 and bin_2. (b) Adjusted results. The black ion group in bin_1 is added to bin_0 and the original black ion groups in bin_0 and bin_2 are removed.

Algorithm 5.2 The adjustment algorithm for overlapping bins

Input Bins containing matching ion groups

Output Matching results with duplicate ions removed

for all consecutive bins bin_0, bin_1 and bin_2

  $T_{MZ} \leftarrow$ MZ cutoff point of bins

  Find ion groups c_n in bin_1 that contain ions whose MZ values range across the boundary T_{MZ}, where $n \in \{1, \dots, N\}$ and N represents the number of selected groups.

  for $n \leftarrow 1$ to N // for each group c_n

    $N_{c_n} \leftarrow$ the total number of ions in group c_n

    Get all matched ion $I_{c_n}^i$ in group c_n, where $i \in \{1, \dots, N_{c_n}\}$

    for $i \leftarrow 1$ to N_{c_n} // for all ions in group c_n

      if the MZ value of $I_{c_n}^i$ is smaller than T_{mz}

        Find the group c' in bin_0 that contains ion $I_{c_n}^i$.

        Record the number of ions in this group as $N_{c'}^i$.

      else

        Find the group c' in bin_2 that contains ion $I_{c_n}^i$.

Record the number ions in this group as $N_{c\prime}^{i}$.

end

end

$N_S \leftarrow N_{c_n}$

while N_S is changing

$N_R \leftarrow 0$

for each ion $I_{c_n}^{i}$ remained in c_n

// check whether ion $I_{c_n}^{i}$ is matched with more ions in neighbouring bins

if $N_{c\prime}^{i} > N_S$

remove $I_{c_n}^{i}$ from c_n

$N_R \leftarrow N_R + 1$

end

end

$N_S \leftarrow N_S - N_R$

end

for each ion $I_{c_n}^{i}$ remained in c_n

Remove the ion in bin_0 or bin_2 that is identical to $I_{c_n}^{i}$

end

Record c_n

Update the results in bin_0 and bin_2 for the next iteration

end

Save the updated results in bin_0 and all updated ion groups c_n to the final matching results

$T_{mz} \leftarrow T_{mz} + bin_{MZ}$

Set the current bin_2 to bin_0 for the next iteration

bin_1 ← Ion matching results in the bin, whose MZ is from $(T_{mz} - bin_{MZ}/2)$ to $(T_{mz} + bin_{MZ}/2)$

bin_2 ← Ion matching results in the bin, whose MZ is from T_{mz} to $(T_{mz} + bin_{MZ})$

end

5.3.4 Parallel implementation

The matching algorithm was implemented in MATLAB 2014a with the parallel computing toolbox increasing the speed of calculation. The whole dataset is first separated into several pieces (e.g., 10). The matching algorithm then processes multiple pieces in parallel and then integrates the results together (as it is shown in Figure 5-4). The computation time can be therefore greatly reduced when a multi-core machine is used.

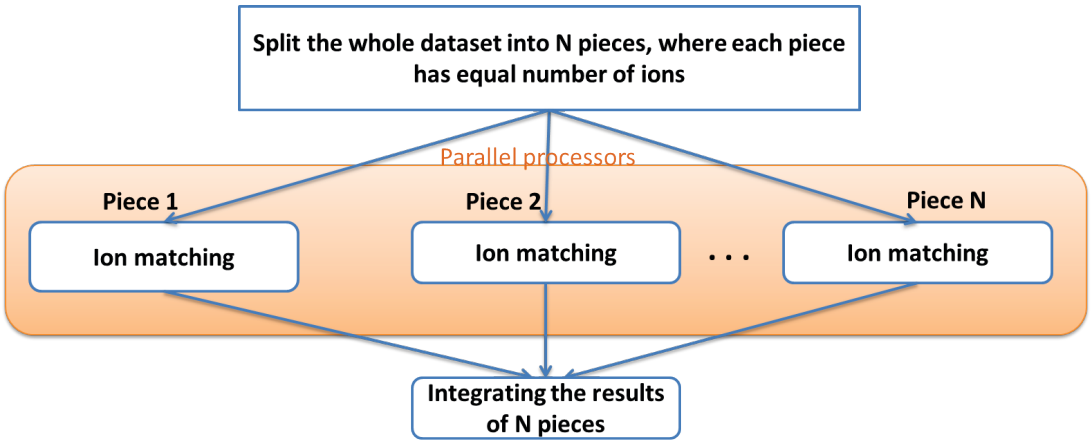


Figure 5-5 Flowchart of doing ion matching in parallel

5.3.5 The whole process

The whole process of ion matching is summarised in Algorithm 5.3. When the number of ions in a MZ bin is quite large, the DT binning is used to further split the data in DT direction. Therefore, ions in each bin can be reduced and hence the processing time on generating large hierarchical tree can be saved.

Algorithm 5.3 The whole process of ion matching

Input Ions from all samples

Output Ion matching results

1 Split the whole dataset into N pieces

2 Initialize N_{max} // DT binning is evoked if the number of ions in MZ is larger than N_{max}

3 **for** $n \leftarrow 1$ to N

4 Define the MZ bins

5 **for** each MZ bin

6 **if** the total number of ion in this bin is larger than N_{max}

7 Define the DT bins

```
8                       for each DT bin
9                           Do ion matching within each DT bin
10                      end
11                      Do the adjustment for overlapping DT bins
12                  else
13                      Do the ion matching within each MZ bin
14                  end
15              end
16              Do the adjustment for overlapping MZ bins
17          end
18          Integrate the results of N pieces
```

5.3.6 Mechanisms of handling batch effect

So far we have discussed about the method of matching ions across samples in a single batch. When the samples are collected in different batches, it is inevitable to have batch effect. The batch effect will introduce variations of 3D information and hence gives challenges to ion matching. In order to facilitate processing samples across batches, we first process samples within each batch separately. Then we use the mean 3D information of each batch to merge results of different batches together.

5.3.7 Performance evaluation

The output matrix (denoted as $Matrix_A$) of the matching method is in the format as in Table 5-1. Each row of the result matrix contains ions of the same ion group. MZ_n^m, DT_n^m and RT_n^m represent the MZ, DT and RT information of an ion from sample m in the nth ion group. When the sample m does not have any ion in ion group n, then the corresponding value of MZ_n^m, DT_n^m and RT_n^m are 0. The output matrix can also contain other information of a DRAMI, such as intensity and some signal parameters. Each DRAMI in the matching results is compared and mapped (where applicable) to peptide data using parameters provided by PLGS, i.e. precursor intensity and signal parameters (position and fwhm) for the MZ, DT and RT dimensions. The format of matrix containing peptide identities of DRAMIs in Table 5-1 (denoted as $Matrix_B$) is shown in Table 5-2. If MZ_n^m, DT_n^m and RT_n^m are 0, then the corresponding $Peptide_ID_n^m$ is empty. If MZ_n^m, DT_n^m and RT_n^m are not zero and PLGS cannot assign the peptide identity to this ion, then we set $Peptide_ID_n^m$ to '###'.

Table 5-1 The format of output matrix generated by the ion matching method

Ion Group	**Sample 1**			**Sample 2**			**Sample 3**			…	**Sample M**		
1	MZ_1^1	DT_1^1	RT_1^1	MZ_1^2	DT_1^2	RT_1^2	MZ_1^3	DT_1^3	RT_1^3	…	MZ_1^M	DT_1^M	RT_1^M
2	MZ_2^1	DT_2^1	RT_2^1	MZ_2^2	DT_2^2	RT_2^2	MZ_2^3	DT_2^3	RT_2^3		MZ_2^M	DT_2^M	RT_2^M
…											…		
N	MZ_N^1	DT_N^1	RT_N^1	MZ_N^2	DT_N^2	RT_N^2	MZ_N^3	DT_N^3	RT_N^3		MZ_N^M	DT_N^M	RT_N^M

Table 5-2 The format of matrix containing peptide identities of DRAMIs

Ion Group	**Sample 1**	**Sample 2**	**Sample 3**	…	**Sample M**
1	$Peptide_ID_1^1$	$Peptide_ID_1^2$	$Peptide_ID_1^3$	…	$Peptide_ID_1^M$
2	$Peptide_ID_2^1$	$Peptide_ID_2^2$	$Peptide_ID_2^3$		$Peptide_ID_2^M$
…					…
N	$Peptide_ID_N^1$	$Peptide_ID_N^2$	$Peptide_ID_N^3$		$Peptide_ID_N^M$

To evaluate the performance of our matching algorithm, only ion groups containing at least one ion with an associated peptide identity are used. That is, we select rows of $Matrix_B$ containing at least one element that is neither empty nor '###'. Then by checking peptide identities of selected rows, we can evaluate the performance of matching algorithm. Ideally, all ions with the same peptide identity across samples should be put in the same ion group. Considering PLGS may make mistakes in identity assignment, we represent the performance of our method through different calculations as follows:

1. Calculate the number of rows in $Matrix_B$ that all ions share the same peptide identity. That is, select the nth row if $Peptide_ID_n^m$ for all $m \in \{1, \dots, M\}$ are identical. Record the selected rows of $Matrix_B$ by a matrix denoted as $Matrix_1$. Record the number of rows as N_1 and calculate P_1 by $\frac{N_1}{N}$. The value of P_1 shows the ability of grouping ions with same identity together.
2. Calculate the number of rows in $Matrix_1$ whose shared peptide identities do not appear in other rows of $Matrix_B$. Record the number as N_2 and calculate P_2 by $\frac{N_2}{N}$. The value of P_2 shows the ability of grouping all ions with same identity together. This value cannot directly reflect the performance of our method, as PLGS may wrongly assign the same identity to ions whose 3D information is quite different.

Thus, we carry out the following calculations to fairly represent our performance by considering possible PLGS errors.

3. Select rows from $Matrix_1$ whose shared peptide identities also appear in other rows of $Matrix_B$. An example is shown in Table 5-3 that all ions in row n_1 have the same identity of $Peptide_ID_1$ but an ion in row n_2 also with this identity. Record the selected rows by a matrix denoted as $Matrix_3$. Calculate the number of rows in $Matrix_3$ that the differences in 3D between ions in the selected row and ions from other rows with same identity are larger than the threshold. Record the number as N_3 and calculate P_3 by $\frac{N_2+N_3}{N}$.

   For the example shown in Table 5-3, calculate the mean MZ value of ions in row n_1 (denoted as $\overline{MZ}_{n_1}$) and also the mean MZ value of ions in row n_2 (denoted as $\overline{MZ}_{n_2}$). Then check whether the difference between $\overline{MZ}_{n_1}$ and $\overline{MZ}_{n_2}$ is larger than thr_{MZ}. Do the similar calculations in RT and DT as well. If the difference in any dimension is larger than the threshold, select the row n_1.

Table 5-3 An example to show ions with the same peptide identity appear in different rows

Ion Group	**Sample 1**			**Sample 2**			**Sample 3**			…	**Sample M**		
n_1	$Peptide_ID_1$						$Peptide_ID_1$				$Peptide_ID_1$		
	$MZ_{n_1}^1$	$DT_{n_1}^1$	$RT_{n_1}^1$	0	0	0	$MZ_{n_1}^3$	$DT_{n_1}^3$	$RT_{n_1}^3$		$MZ_{n_1}^M$	$DT_{n_1}^M$	$RT_{n_1}^M$
n_2				$Peptide_ID_1$									
	0	0	0	$MZ_{n_2}^2$	$DT_{n_2}^2$	$RT_{n_2}^2$	0	0	0		0	0	0

   The N_3 rows selected in this step stand for the case that PLGS assigns the same peptide identity to ions whose 3D difference is larger than the threshold (denoted as Case 1). In this case, we cannot say ions of the same identity are wrongly grouped into different rows by our method. It is quite likely that PLGS makes mistakes in identity assignment.

4. Select rows from $Matrix_B$ whose ions do not have identical peptide identity but their identities do not appear in other rows of $Matrix_B$. Record the number of selected rows as N_4 and calculate P_4 by $\frac{N_2+N_3+N_4}{N}$. The N_4 rows selected in this step are for the case that ions with similar 3D values are assigned with different identities by PLGS (denoted as Case 2). Thus, we cannot speculate that our method makes mistakes of grouping different peptides together.

5. Select rows from $Matrix_B$ whose ions do not have identical peptide identity and their identities also appear in other rows of $Matrix_B$. An example is shown in Table 5-4 that row n_1 and n_2 contain ions with two different peptide identities. Calculate the number of rows that the difference in 3D of ions with same identity among rows is larger than the threshold. Record the number as N_5 and calculate P_5 by $\frac{N_2+N_3+N_4+N_5}{N}$. For the example shown in Table 5-4, we first calculate the mean MZ value of ions with the identity of $Peptide_ID_1$ in row n_1, denoted as $\overline{MZ}_{n_1}^1$. Then we calculate the mean MZ value of ions with the identity of $Peptide_ID_1$ in row n_2, denoted as $\overline{MZ}_{n_2}^1$. Afterwards check whether the difference between $\overline{MZ}_{n_1}^1$ and $\overline{MZ}_{n_2}^1$ is larger than thr_{MZ}. Do the similar calculations in RT and DT as well. If the difference in any dimension is larger than the threshold, do the same check for ions with the identity of $Peptide_ID_2$. If the difference in any dimension is larger than the threshold for ions with $Peptide_ID_2$, then select row n_1 and row n_2. The N_5 rows selected in this step are for the case that our method groups ions according to their 3D information, while PLGS assigns identities to ions without considering 3D difference (denoted as Case 3).

Table 5-4 An example to show some rows whose ions do not have identical peptide identity and their identities also appear in other rows

Ion Group	**Sample 1**			**Sample 2**			**Sample 3**			…	**Sample M**		
n_1	$Peptide_ID_1$			$Peptide_ID_2$			$Peptide_ID_1$				$Peptide_ID_2$		
	$MZ_{n_1}^1$	$DT_{n_1}^1$	$RT_{n_1}^1$	$MZ_{n_1}^2$	$DT_{n_1}^2$	$RT_{n_1}^2$	$MZ_{n_1}^3$	$DT_{n_1}^3$	$RT_{n_1}^3$		$MZ_{n_1}^M$	$DT_{n_1}^M$	$RT_{n_1}^M$
n_2	$Peptide_ID_2$			$Peptide_ID_1$			$Peptide_ID_2$				$Peptide_ID_1$		
	$MZ_{n_2}^1$	$MZ_{n_2}^1$	$MZ_{n_2}^1$	$MZ_{n_2}^2$	$MZ_{n_2}^M$	$MZ_{n_2}^M$	$MZ_{n_2}^M$	$DT_{n_2}^3$	$RT_{n_2}^3$		$MZ_{n_2}^M$	$DT_{n_2}^M$	$RT_{n_2}^M$

6. Calculate the percentage of un-identified ions. Record the number of ions whose peptide identity is '###' in $Matrix_B$ as $\widetilde{N}$. Count the number of ions in $Matrix_B$ whose identity is not empty as $\widetilde{M}$. Calculate P_6 by $\frac{\widetilde{N}}{\widetilde{M}}$. This ratio shows the ability of our method in detecting features that are not recognized by PLGS.

5.3.8 GUI

We have developed a graphical user interface (GUI) for users to do ion matching. The screenshot of GUI is shown in Figure 5-6. There are several steps to complete the whole matching process. The button in the next step will only be enabled if the task in the previous

step has been finished. The program status is displayed in real time, which can be saved into a log file.

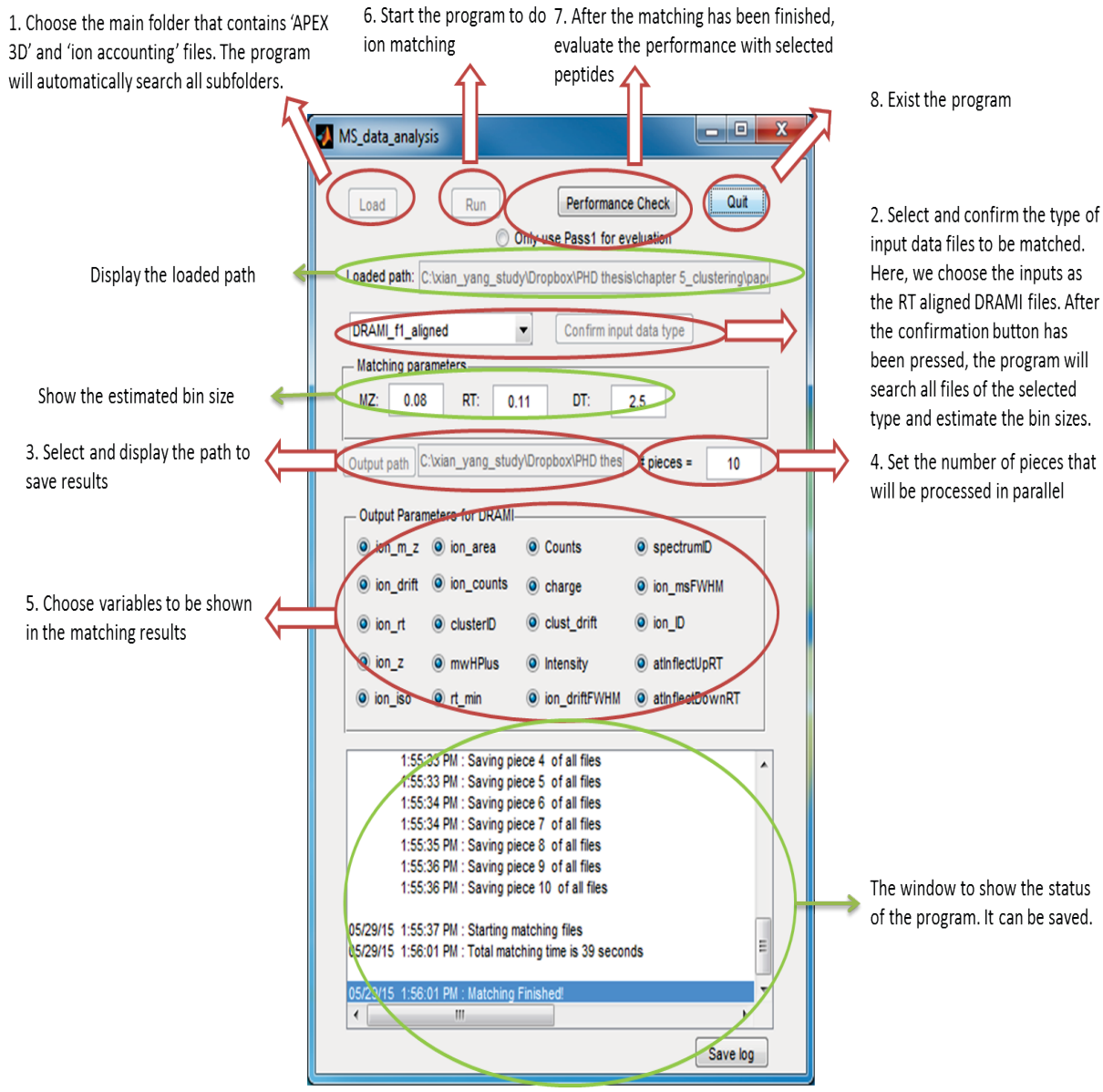


Figure 5-6 The graphic user interface of the developed tool

Users first specify the location of 'APEX 3D' and 'ion accounting' files and in the meanwhile the selected folder path will be displayed in GUI. Then users can choose the type of input file. In this chapter, we focus on analysing RT-aligned low function DRAMI files, so we choose the input file type to be 'DRAMI_f1_aligned'. Users can also analyse other types of input files, including no RT-aligned DRAMI files and ion accounting files. By clicking the button to confirm the input data type, our program will automatically search all subfolders of the specified path to find target input data files. Our program will also return its estimation of

thresholds in 3D. Users can either directly use these values or set thresholds based on their own knowledge. Then users choose a folder to save the matching results. In order to increase the processing speed, we split the whole dataset into pieces, where users can decide the number of pieces or use the default value of 10. There is also a window for users to choose which variables they want to save in the matching results. After all these settings have been done, users can click the 'run' button to start matching ions. The status of processing will be interactively displayed. Users can acquire the progress in real time. After the matching process has been finished, users can use the matching results for further analysing or use our program to evaluate the performance of matching.

5.4 Experiment and results

5.4.1 Experimental design

As it is shown in Table 5-5, there are several datasets used to illustrate our method, including an experimental set derived from various combinations of peptide standard mixtures and small molecules against a complex background, and a non-depleted serum dataset with a spiked red blood cell (RBC) lysate dilution series. The details of preparing samples will be shown in the following part. Experiments are designed in such a way that we would allow us to test: 1) the accuracy and efficiency of our method, and compare this with PLGS ion assignment errors; 2) the ability of identifying peptides with known concentrations and ratios against a complex background of E. coli lysate; 3) the potential of identifying specific non-database and/or non-peptide ions through the addition of lysophosphatidylcholine lipid species; 4) the ability of analysing large and highly complex samples through a serum red blood cell (RBC) spike in experiment; 5) and the capability of handling batch effect.

Table 5-5 Experimental design and rationale

	Samples	**Purpose**
Dataset 1	6 samples of E.coli background	Identify systematic variability in measurement
Dataset 2	2 replicates of *E. coli* + Pepmix 1 2 replicates of *E. coli* + Pepmix 1+PC 2 replicates of *E. coli* + Pepmix 2 2 replicates of *E. coli* + Pepmix 2+½PC	Estimate the accuracy and efficiency of identifying discriminant features against complex background

Dataset 3	RBC Serum Spike-in Series of 51 samples	Show the ability of analysing large scale dataset
Dataset 4	2 replicates of Sputum + Pepmix 1 from batch 1 2 replicates of Sputum + Pepmix 2 from batch 1 2 replicates of Sputum + Pepmix 1 from batch 2 2 replicates of Sputum + Pepmix 2 from batch 2	Show the capability of handling batch effect

5.4.2 Data Acquisition

5.4.2.1 Sample mixtures and Sample preparation

The compositions of each of the specific mixtures tested are outlined in Table 5-6. *E. coli* standard lysate (Waters), Sputum, and Pepmix 1 and 2 (Waters) were re-suspended in MS buffer (3% Acetonitrile 0.1% formic acid), according to manufacturer's instructions. Lysophosphatidylcholine standards mixtures were prepared in the Laboratory of Prof. Tony Postle (Southampton General Hospital, SGH) and diluted 1000 fold in MS buffer. All samples were prepared as described in Table 5-6. A RBC spike-in dataset was prepared from whole serum samples from three individuals collected according to standard ethics approval at SGH. Each sample had packed red blood cell lysate introduced as described in (Kirschner *et al.,* 2011)) and in dilutions described in Table 5-6. Samples from the RBC series (5µl) were diluted 10 fold in 0.1M ammonium bicarbonate (Ambic) heated at 90°C with 10µl 1% RapiGest (waters) surfactant and reduced and alkylated in the standard manner using DTT and IDA and Trypsin digested at 37°C overnight at 1:50 expected protein concentration. Following overnight digestion another 1:50 portion of Trypsin was added and the samples were incubated at 37°C for a further 3 hours. Digests were vacuum dried and cleaned with C18 tips (Protea) according to manufacturer's instructions, vacuum dried and stored at -80°C. 500 ng of each sample was mixed with an internal standard (Waters yeast Enolase digest) such that the concentration of standard in the sample was 50fmol/µl. The mixture was introduced into the mass spectrometer as described below. Each sample was injected twice resulting in 6 replicates for each RBC concentration (except for RBC 100 which was only injected 3 times), equating to 51 injections in total.

Table 5-6 Standard mix compositions

Sample	Composition	Standard mix quantities (per µl)
E. coli + Pepmix 1	200 ng *E. coli* digest + Pepmix 1	50 fmol ENO1, 78.1 fmol ADH, 66.3 fmol GBP, 50fmol BSA
E. coli + Pepmix 2	200 ng *E. coli* digest + Pepmix 2	50 fmol ENO1, 41.5 fmol ADH, 17.02 fmol GBP, 240.3 fmol BSA
Sputum + Pepmix 1	Sputum + Pepmix 1	50 fmol sputum, 78.1 fmol ADH, 66.3 fmol GBP, 50fmol BSA
Sputum + Pepmix 2	Sputum + Pepmix 2	50 fmol sputum, 41.5 fmol ADH, 17.02 fmol GBP, 240.3 fmol BSA
E. coli + Pepmix 1+ PC	200 ng *E. coli* digest + Pepmix 1+ LPC standard mix	*E. coli* + Pepmix 1 and PC Mix of: LPC10:0 (caproyl), LPC11:1 (undecenoyl), LPC14:0 (myristoyl), LPC18:2 (linoleoyl), LPC 20:4 (arachidonoyl), LPC 20:0 (arachidoyl) species in various quantities
E. coli + Pepmix 2 + ½ PC	200 ng *E. coli* digest + Pepmix 2 + 50% dilution of LPC standard mix	*E. coli* + Pepmix 2 and 50% dilution of PC species as above
RBC spike in series	Serum with spiked packed RBC lysate	RBC spike in at (%): 100, 5, 0.5, 0.25, 0.125, 0.063, 0.032, 0.016, 0 (neat)

*ENO1: Yeast Enolase, ADH: Yeast Alcohol dehydrogenase, GBP: Rabbit Glycogen Phosphorylase B, BSA: Bovine serum albumin, LPC: Lysophosphatidylcholine. Pepmix 1 and Pepmix 2 standards were set up such that the ratios between Pepmix 1 and 2 were - ENO1 1:1, ADH 1.9:1, GPB 3.9:1, BSA 0.2:1. RBC: Red blood cells

5.4.2.2 Mass spectrometry

Sample mixtures were injected in duplicate and separated on a Waters Acquity Nano UPLC. Samples were introduced to a C18 trapping column at 5µl/min and washed for 5 minutes. Peptides were eluted via a 15cm C18 analytical column with a gradient of 3-50% ACN over 45 minutes at a flow rate of 0.3µl/minute. The RBC spike-in set was run with similar conditions but at a gradient of 3-40% ACN over 40 minutes. Gas phase ions were produced

by electrospray and were introduced into the orifice of a Waters Synapt G2S HDMS with ion mobility enabled with an IMS velocity of 650 m/s and a wave height of 40V. Ion information was collected via a MS^E (continuum) experimental protocol where the instrument alternated between a low energy state and a high energy state on alternate seconds. In the low energy state intact precursor ions are allowed to reach the detector, and in the high energy state ions were collided with argon, distal to the ion mobility cell, providing fragment information for all fragmentable ions in the sample. Precursor and fragment masses were measured via time of flight, operating in resolution (single reflectron) mode. Intensity of ion packets was determined through measurement with an ADC detector. A lock-mass of glufibrinopeptide was introduced into the instrument for 0.5 seconds every 60 seconds.

5.4.2.3 Data curation and searching

Raw data files were processed to DRAMI using PLGS 2.5.2 with a low energy threshold of 150 counts, a high energy threshold of 35 and a total energy count of 750. This produced the DRAMI information used in our procedure. To produce information for peptide picking and assessment of our matching pipeline, database searches were performed against standard Uniprot/Swissprot human and *E. coli* databases with added sequence information for Pepmix standards. The following setting were used: Fixed modification of Carbamidomethyl cysteine; and variable modifications of Acetyl N terminus, Deamidation of N and Q, and Oxidation of M. Two missed cleavages were allowed with a minimum of 7 fragments per protein, three fragments per peptide and two peptides per protein required for identification. Database searching continued until a false positive rate of 4%, assessed by a decoy database strategy, was reached. PLGS 'APEX 3D' 'Peptide 3D.xml' and 'ion accounting' file folders were saved for each sample.

5.4.3 Results

Based on an analysis of the instrument parameters for the *E. coli* data, thresholds of 0.11 min in retention time, 0.08 ppm in m/z and 2.5 drift bins are used for matching DRAMI ions in Dataset 1 and Dataset 2. For Dataset 3, the thresholds are slightly different, which is 0.13 min in retention time and 0.05 ppm in m/z. For Dataset 4, we choose the thresholds to be 0.5 min in retention time, 0.08 ppm in m/z. We will first investigate the characteristic of MS data using Dataset 1. Then we will use Dataset 2 to investigate the ability of our method in detecting features with both known composition and unknown identify. Dataset3 further illustrates our method by large-scale analysis. The performance of using our method for these

3 datasets is evaluated and some examples of errors from PLGS are shown. Finally, we will use Dataset4 to demonstrate the ability of handling batch effect.

5.4.3.1 Investigating the characteristic of measurements

We first apply our method to Dataset 1, where the inputs are from 6 samples containing 401,057 ions in total. The outputs of our method are 107,938 matching rows, each of which is an ion group containing similar ions across samples. Figure 5-7 provides a visual inspection of matching results using 3D plots of ions. 50 ion groups are randomly selected for illustration and DRAMI ions from the same group are assigned with the same colour. From Figure 5-7, we can get an intuitive impression that ions with similar values of MZ, RT and DT are grouped together.

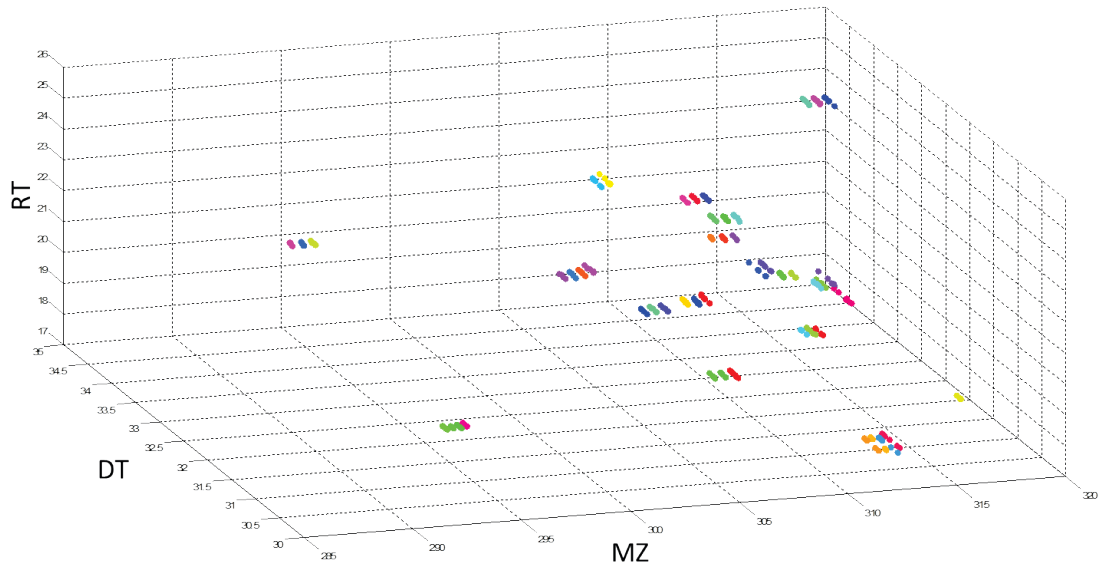


Figure 5-7 Output of the matching algorithm in 3D space defined by m/z, retention time and drift axis. Display 50 of total 107,938 matched ion groups. Each dot represents a DRAMI ion in a single sample. Each colour corresponds to different chemical entities present in the data set.

In each ion group, some ions are assigned with peptide identities by PLGS. Also, not all groups contain ions across all samples. We would like to focus our attention on ion groups that contain at least one ion with identity. There are a number of 19,587 ion groups selected. 6711 of them contain ions that all have identity (labelled as 'all ions identified'), while 3737 groups contain ions across all samples but not all of them having identity (labelled as 'all ions matched'). There are 9139 other ion groups that do not fall into the above categories (labelled as 'other matched groups'). Figure 5-8, Figure 5-9 and Figure 5-10 investigate the variations

in MZ, RT and DT for ion groups in different categories. We can find that ion groups in the 'all ion identified' category has slightly smaller difference in MZ, RT and DT than groups in other categories. This means that ions that are identified by PLGS have relatively smaller 3D variations across samples than others. However, by comparing the median and 3rd quartile values in all boxplots, we can see that the difference among different categories is not significant.

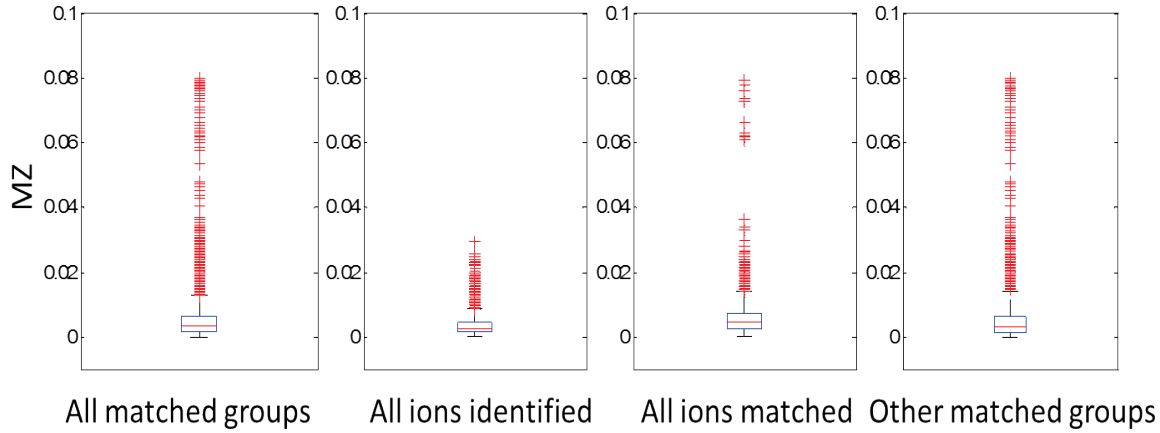


Figure 5-8 Boxplots of MZ difference. The MZ difference of each ion group is calculated as the maximum MZ value minus the minimum MZ value of ions in each group. 4 boxplots are generated for categories of 'all matched groups', 'all ion identified', 'all ions matched' and 'other matched groups'. The median MZ differences are 0.0035, 0.0027, 0.0046 and 0.0034, respectively; and the 3rd quartile values are 0.0063, 0.0045, 0.0073 and 0.0066, respectively;

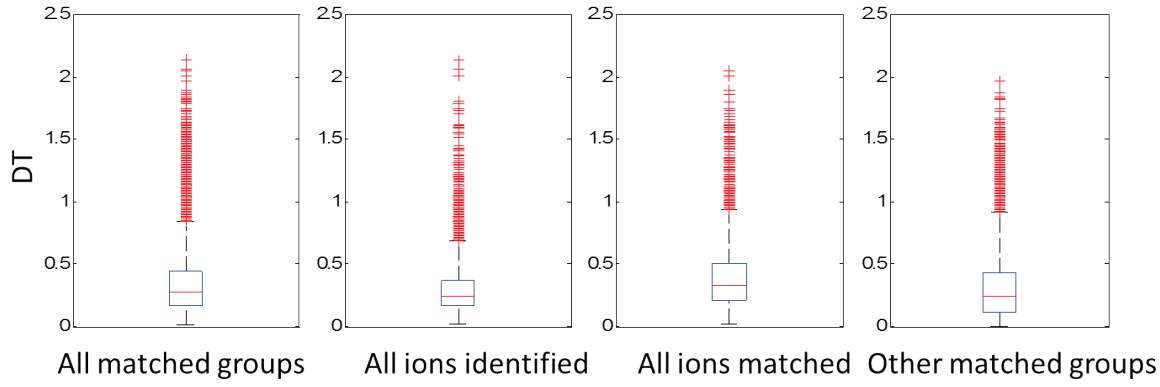


Figure 5-9 Boxplots of DT difference. The DT difference of each ion group is calculated as the maximum DT value minus the minimum DT value of ions in each group. 4 boxplots are generated for categories of 'all matched groups', 'all ion identified', 'all ions matched' and

'other matched groups'. The median DT differences are 0.27, 0.24, 0.33 and 0.24, respectively; and the 3rd quartile values are 0.44, 0.37, 0.50 and 0.43, respectively.

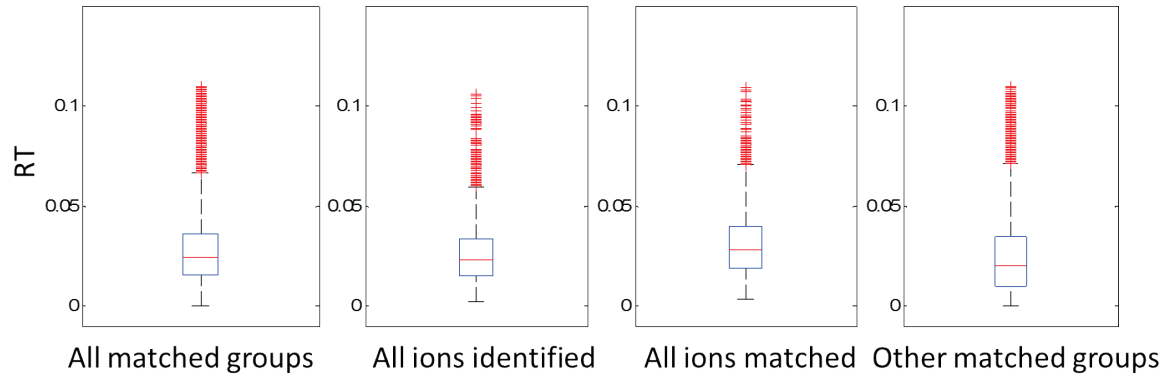


Figure 5-10 Boxplots of RT difference. The RT difference of each ion group is calculated as the maximum RT value minus the minimum RT value of ions in each group. 4 boxplots are generated for categories of 'all matched groups', 'all ion identified', 'all ions matched' and 'other matched groups'. The median RT differences are 0.024, 0.023, 0.028 and 0.021, respectively; and the 3rd quartile values are 0.036, 0.033, 0.040 and 0.034, respectively.

We further compare the counts (i.e. intensities) of ions within different categories. Figure 5-11 shows boxplots of ion counts from 4 different categories. The median counts are 3817, 9123, 1636 and 2376, respectively. It is quite clear that ion groups in the 'all ions identified' category have ions with higher intensity. That is, ion groups having higher number of identity assignments tend to have higher intensity when compared to those ion groups with less assigned identities. This is of our expectation as high intensity peptides are more likely to yield good and consistent fragmentation data. So it is easier for PLGS to assign identities to high intensity ions. Although ion groups with less identity assignment have less abundant ions, their 3D measurements do not vary significantly from those ions in the 'all ions identified' category. This tells us that PLGS can only work well on ions with high intensity, while our method can match all ions regardless of their intensity. Therefore, unlike PLGS, features from low abundant range can be picked out by our method.

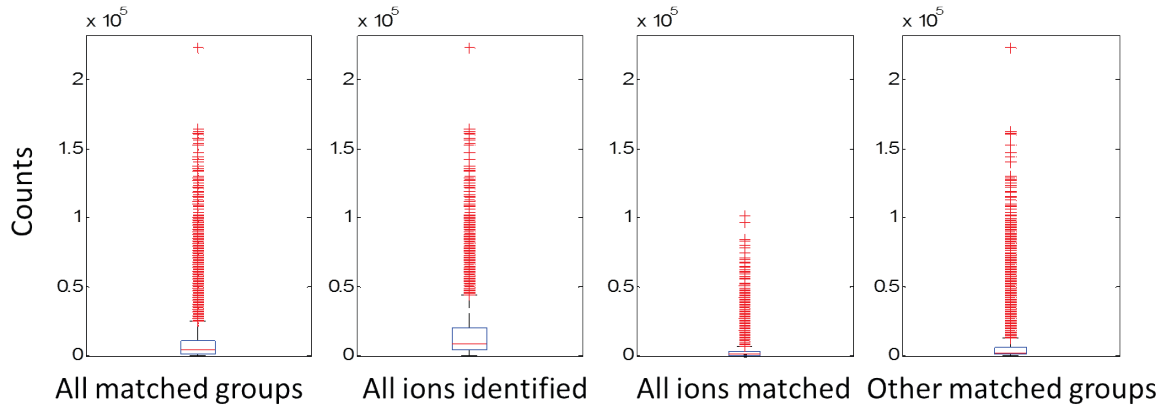


Figure 5-11 Boxplots of count. 4 boxplots show the distributions of ion counts for categories of 'all matched groups', 'all ion identified', 'all ions matched' and 'other matched groups'. The median counts are 3817, 9123, 1636 and 2376, respectively; and the 3rd quartile values are 11184, 20163, 3498 and 5840, respectively.

5.4.3.2 Detecting proteins with known intensity ratios between sample groups

We now use Dataset 2 to assess the matching performance of detecting proteins with known ratios. As it is show in Table 5-6, Pepmix 1 and Pepmix 2 standards were set up in such a way that the ratios of ENO1, ADH, GPB and BSA between Pepmix 1 and 2 are 1:1, 1.9:1, 3.9:1 and 0.2:1. We calculate the top 3 intensity sums for matched ion groups to check whether the estimated ratios of these 4 known proteins are closed to expected values.

We first select ion groups containing ions with the same peptide identity, from which we further select ion groups containing peptides of the target proteins. That is, a number of 340 ion groups are chosen. The total number of ions contained in these 340 ion groups is 1875; and 1053 ions out of them have peptide identities. Therefore, 818 ions without the assignment of identity by PLGS are matched to 1053 identity known ions. Without using our method, these 818 ions will be viewed as non-detection. Figure 5-12 compares the differences between the mean values of identified ions and unidentified ions in MZ, DT and RT for 340 ion groups. The 3rd quartile values in MZ, DT and RT differences are 0.025, 0.18 and 0.023 respectively, which are much lower than their corresponding thresholds. Thus, the 3D characteristics of PLGS identified ions and unidentified ions are quite similar. However, due to the limitation of PLGS, those unidentified ions which could potentially have identities are not discovered.

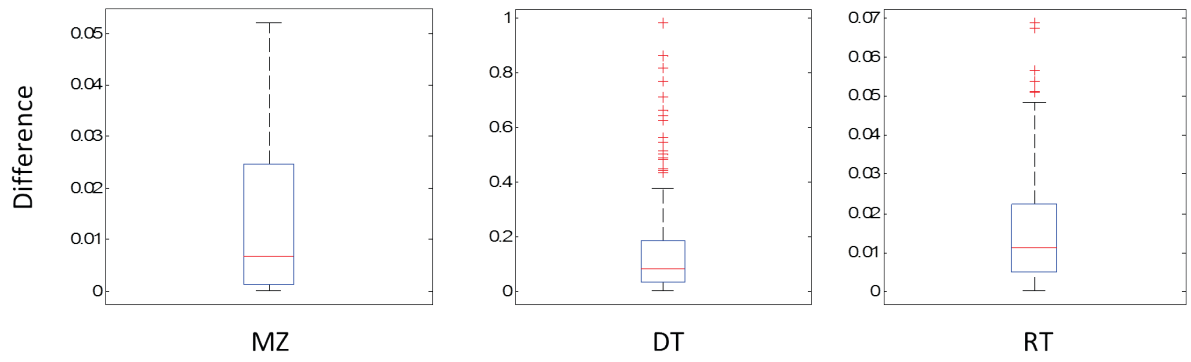


Figure 5-12 Boxplots of differences in MZ, DT and RT between the mean values of identified ions and unidentified ions within each ion group. The median values are 0.0068, 0.083 and 0.011 respectively; and the 3rd quartile values are 0.025, 0.18 and 0.023 respectively.

We then work on the selected 340 ion groups to calculate the top 3 intensity sums of each protein as follows: 1) detect ion groups containing peptides of the target protein; 2) select ion groups containing monoisotopes; 3) choose ion groups with the top 3 highest intensities; 4) calculate the sum of the top 3 intensity; if the number of ion groups selected in 3) is 2, then calculate their sum and times 3/2. Table 5-7 shows the results of estimated ratios of top 3 intensity sums for the 4 known proteins. It is noteworthy to mention that all results use matching groups containing at least 1 ion that is not assigned with an identity. Figure 5-13 shows the boxplot of ratios of other proteins in Dataset 2. The expected ratios of those proteins are 1 and the median value of estimated ratio is 0.93. Considering the systematic variability, the deviation from the expected ratios is not significant. Moreover, the whole Dataset 2 contains a number of 705,032 ions and only 1875 ions are selected in top 3 calculation. It is quite likely that the rest ions have somewhat effects on the accuracy of the results. Therefore, the results in Table 5-7 are quite encouraging. If normalisation is used to remove systematic bias, we could expect the estimated ratios will be closer to the expected values.

Table 5-7 Ratios of top three intensity sums for four known proteins.

Protein	**Ratio Observed vs. Expected**		**Mean Top 3 intensity sum**	
			Mix 1	**Mix 2**
ADH	Estimated	0.61	28509	17547
	Expected	0.53		

ENO1	Estimated	1.06	18609	19573
	Expected	1.0		
BSA	Estimated	4.0	24186	96349
	Expected	5.0		
GPB	Estimated	0.22	18714	4173
	Expected	0.26		

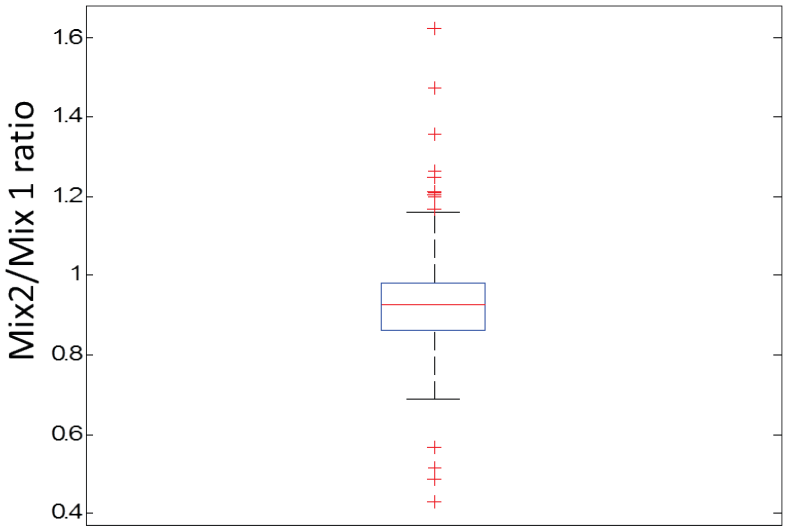


Figure 5-13 Boxplot for ratios between Pepmix 1 and Pepmix 2 of other proteins. The median, 1st quartile and 3rd quartile values are 0.93, 0.86 and 0.98.

5.4.3.3 Detecting unidentified features

The results of applying our method to Dataset 2 are a total number of 160,566 matching ion groups. Only a small number of groups contain ions with identity, which is 18,830. A large portion of ion groups is not identified by PLGS, while it may contain important information. Here, we would like to investigate the ability of our method to detect features that are not identified by PLGS. In our experimental design, LPC standard mix is added to 'E. coli + Pepmix 1+ PC' samples and 50% dilution of LPC standard mix is added to 'E. coli + Pepmix 2 + ½ PC' samples. As PC mix does not contain proteins and its ions cannot be assigned with identity by PLGS, traditional method cannot detect compounds in PC mix. Benefit from our unbiased approach, we can navigate all unidentified ions to find potential compounds in PC mix.

A straightforward way to detect compounds in PC mix is as follows: 1) select ion groups that all ions are unidentified; 2) select ion groups that contain ions appearing in all 'E. coli + Pepmix 1+ PC' and 'E. coli + Pepmix 2 + ½ PC' samples but missing in all 'E. coli + Pepmix 1' and 'E. coli + Pepmix 2' samples; 3) select ion groups where ratios between mean count values of ions from 'E. coli + Pepmix 2 + ½ PC' samples and from 'E. coli + Pepmix 1+ PC' samples are smaller than 2/3; 4) select groups are not from contaminations based on MZ value. The selection of ion groups is in a rough manner. For example, it is very likely that some possible candidates of PC mix compounds can be not detected in all 'E. coli + Pepmix 1+ PC' and 'E. coli + Pepmix 2 + ½ PC' samples. However, we follow this simple way to just illustrate the potential ability of our method in detecting unidentified features.

Following the above steps, there are 295 ion groups selected. Biologists need to manually navigate those ion groups to check whether they contain possible compounds from LPC mix. Here, we can do an initial analysis by inputting the MZ values of ions in selected ion groups into LIPD MAPS (Cotter et al. 2006). For instance, there is an ion group, where the mean MZ value of ions is 412.2 ppm. LIPD MAPS searches the LIPID MAPS Structure Database (LMSD) with the MZ value of 412.2 and turns the identity of this ion group as LPC10:0 (http://www.lipidmaps.org/data/LMSDRecord.php?&LMID=LMGP01050005). It is quite encouraging that our method can detect features that are not identified by PLGS.

5.4.3.4 Analysing large scale dataset

To further test our approach, we analyse a complex dataset of human serum with RBC lysate spiked in at known concentrations (i.e. Dataset 3). There are 51 samples in total and we select ion groups that contain ions appearing in more than 41 samples. The rationale behind this is to focus our analysis on ion groups that have good coverage in Dataset 3. To investigate whether unidentified ions contain important information of the dataset, we create 4 sub-datasets. The details of these sub-datasets are shown in Table 5-8. Sub-dataset 1 contains complete ion matching results of Dataset 3, while Sub-dataset 2 only contains ion groups with at least one ion having peptide identity. Sub-dataset 3 depletes unidentified ions in Sub-dataset 2 and we can see the total number of ions is reduced from 506,026 to 141,280. Sub-dataset 4 select ion groups that all ions are unidentified. We can see that 2,322,657 out of 2,828,683 ions are unidentified. Without using our method, a lot of ions that may contain important information of the dataset will be discarded.

Table 5-8 Details of sub-datasets of Dataset 3

	No. of ion groups	No. of ions	Details
Sub-dataset 1	60,349	2,828,683	all matching ion groups
Sub-dataset 2	10,650	506,026	ion groups with any identification present
Sub-dataset 3	10,650	141,280	only ions that have identities assigned in PLGS
Sub-dataset 4	49,699	2,322,657	ion groups without any associated identifications

Ward's linkage hierarchical clustering method is used to generate dendrograms of these sub-datasets. Figure 5-14, Figure 5-15, Figure 5-16 and Figure 5-17 show the results of clustering samples of Sub-dataset 1, Sub-dataset 2, Sub-dataset 3 and Sub-dataset 4 respectively. The dendrograms in these figures tell us which samples are closely grouped together than others. Most sample names are written in the same the format as 'RBC_0.25_2_001', where '0.25' is the concentration of RBC, '2' means the second sample and '001' stands for the first replicate. For samples whose RBC concentrations are 100, their file names are specified as 'RBC_001', where '001' represents the first replicate. For samples with no RBC, their file names are written as 'RBC_Neat_2_001'. From these figures, we can observe that samples with similar concentrations of RBC tend to be clustered together. Particularly, the replicates are always grouped closely. This tells that the results of the matching method maintain the characteristic of original dataset, that experimental designed similar samples can be detected by a straightforward clustering method.

Figure 5-14 depicts the clustering results of Sub-dataset 1, where all ions are used, while Figure 5-17 is generated from Sub-dataset 2 that only unidentified ions are used. By comparing Figure 5-14 with Figure 5-17, we can see that these two sub-datasets return similar clustering results. It tells that unidentified ions contain a massive amount of information of the original dataset and the clustering result of the whole dataset is largely dominated by unidentified ions. Although those unidentified ions are not recognized by PLGS, they are still very important features from which similarity of samples can be detected.

The clustering results of Figure 5-15 and Figure 5-16 are quite similar. Sub-dataset 2 used for Figure 5-15 contains ion groups with at least one ion having identity, while Sub-dataset 3 used for Figure 5-16 depletes all unidentified ions in Sub-dataset 2. Therefore, ions in Sub-

dataset 3 are from a subset of Sub-dataset 2. Comparing Figure 5-15 with Figure 5-16, we can see that the clustering results are quite similar. It means that unidentified ions added to Sub-dataset 3 do not override the original information conveyed by identified ions.

In this result part, the clustering analysis is directly done on the raw datasets. In the future, before we do clustering, we can first do some manipulations of the datasets, such as normalisation and log transformation. Moreover, we can impute missing values with a pseudo limit of detection, which is calculated as half of the lower limit of detection of any ion.

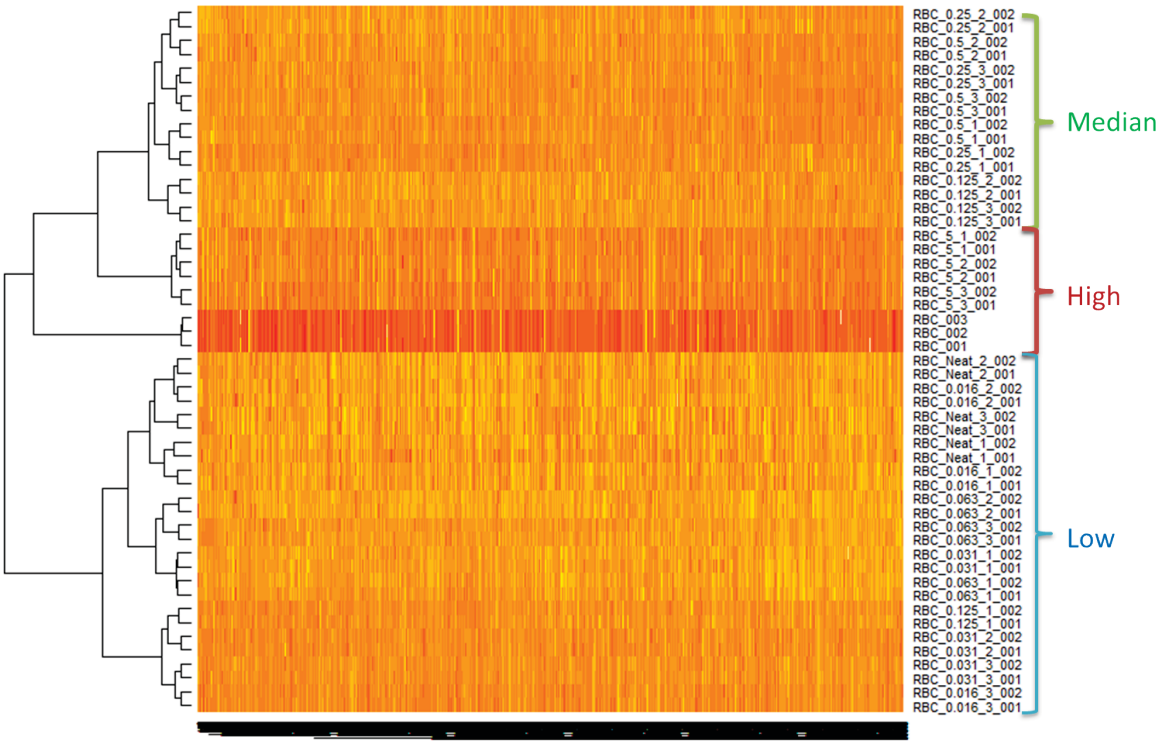


Figure 5-14 Heatmap and dendrogram of Sub-dataset 1. Samples with RBC concentrations ranging from 0.125 to 0.5 are grouped together. Samples with RBC concentrations of 5 and 100 form a cluster labelled as High. Other samples whose RBC concentrations are relatviely low, ranging from 0 to 0.125, are also clustered together.

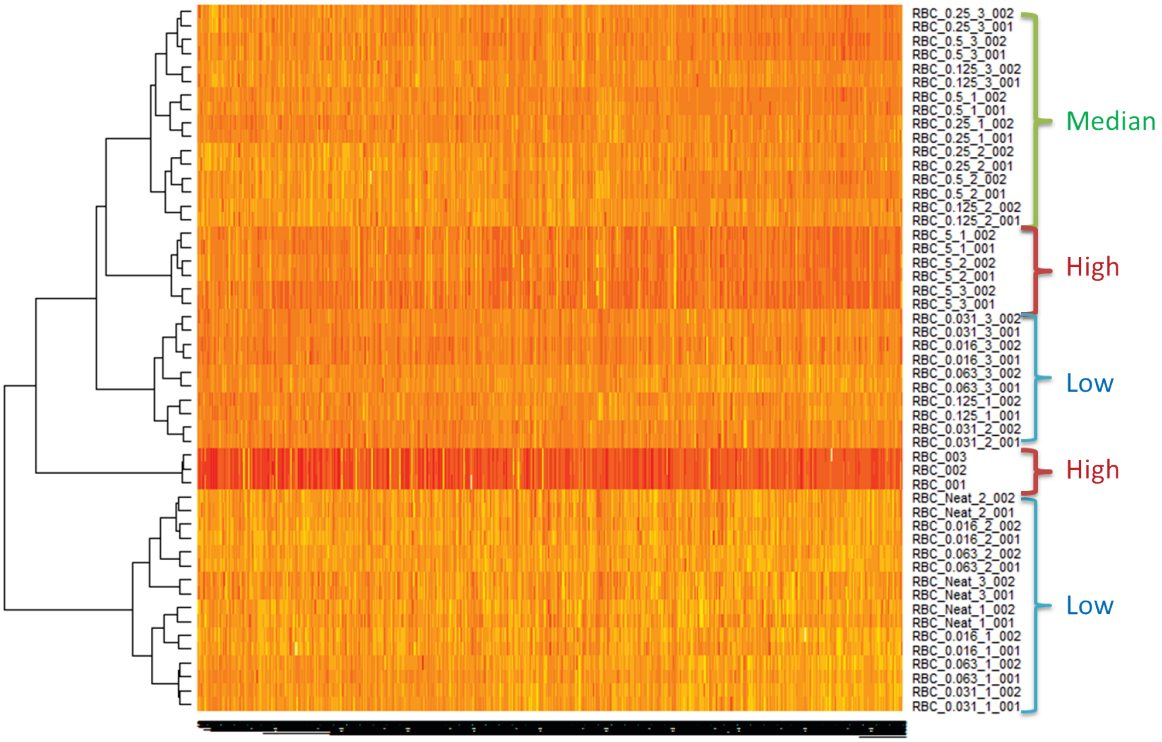


Figure 5-15 Heatmap and dendrogram of Sub-dataset 2. Samples with RBC concentrations in median range of 0.125 to 0.5 are grouped together. Samples with RBC concentrations of 100, which are much larger than all other samples form a single group. Also samples with RBC concentration as high as 5 are clustered together. There are two groups containing samples with RBC concentrations in low ranges, which are from 0 to 0.063 and 0.016 to 0.125.

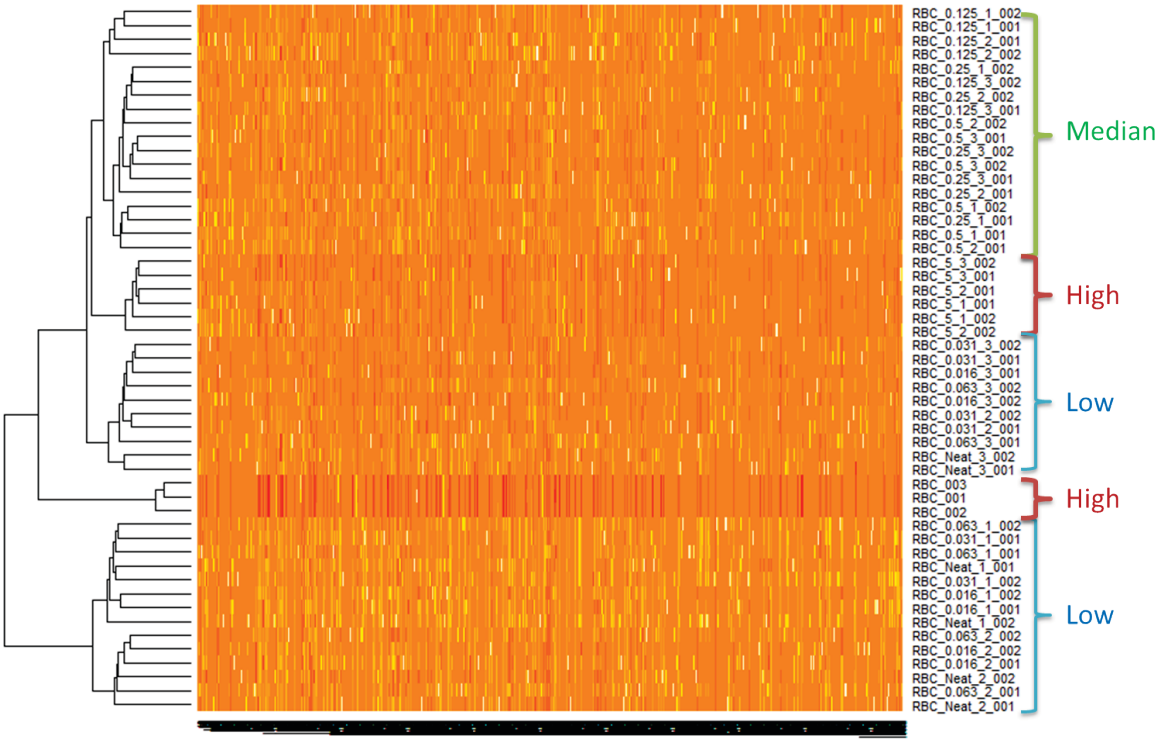


Figure 5-16 Heatmap and dendrogram of Sub-dataset 3. Samples with RBC concentrations in median range of 0.125 to 0.5 are grouped together. Samples with RBC concentrations of 100, which are much larger than all other samples form a single group. Also samples with RBC concentration as high as 5 are clustered together. There are two groups containing samples with RBC concentrations ranging from 0 to 0.063.

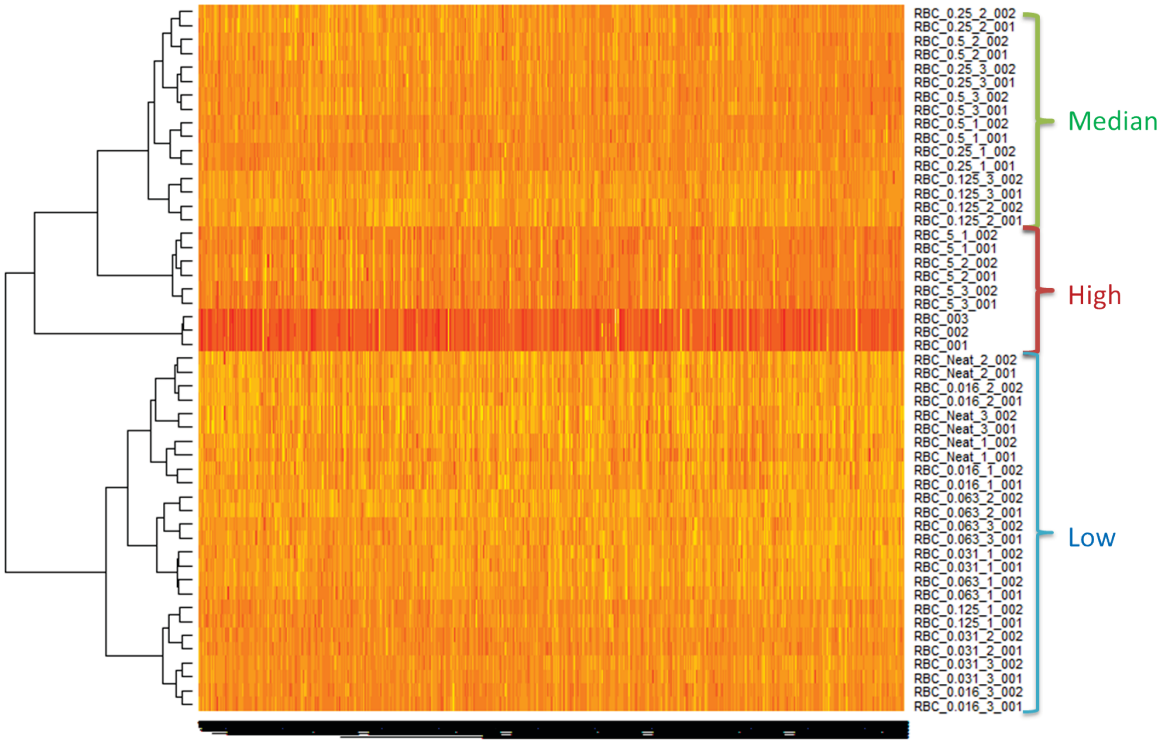


Figure 5-17 Heatmap and dendrogram of Sub-dataset 4. We can see that samples with RBC concentrations ranging from 0.125 to 0.5 are grouped together. Samples with RBC concentrations of 5 and 100 form a cluster labelled as High. Other samples whose RBC concentrations are relatviely low, ranging from 0 to 0.125, are also clustered together. The results are similar with the results in Figure 5-14.

5.4.3.5 Performance evaluation

In order to assess the accuracy of ion matching in an automated fashion we built an assessment step into the analysis pipeline to assess the number of ion groups with high quality identification. The details of performance evaluation are described in subsection 5.3.7. Table 5-9 shows the summary of matching results.

Table 5-9 A summary of matching results

	Dataset 1	**Dataset 2**	**Dataset 3**
P1	0.96	0.92	0.92
P2	0.91	0.81	0.65
P3	0.95	0.91	0.86
P4	0.97	0.95	0.92

P5	0.99	0.99	0.94
P6	0.26	0.42	0.62
# matching rows used for performance evaluation	19,587	23,189	11,957
# matching rows in the output	107,938	160,566	777,393
Execution time	42s	68s	1,056s
# ions	401,057	705,032	7,938,412

First, we can observe that the smallest dataset, Dataset 1, contains 401,057 ions and the largest dataset, Dataset 3, have 7,938,412 ions which is the 20 times of Dataset 1. The execution time does not increase exponentially with the increasing number of ions. More specifically, the matching algorithm only takes 42 seconds to analyse Dataset 1, while the execution time for Dataset 3 is around 25 times of 42 seconds (i.e., 1,056s). This observation tells us that our method can work on large-scale dataset within a reasonable time.

Secondly, we can see that only a small fraction of matching rows is selected to evaluate the performance. The performance evaluation highly depends on the assignment of peptide identity by PLGS. P1 to P5 in Table 5-9 show the accuracy of our method as described in subsection 5.3.7. We can see that those values for different datasets are maintained at a high level. Although P2 for Dataset 3 is only 0.65, P3, P4 and P5 are at a satisfactory level where errors that may be made by PLGS are taken into consideration. Here, we would like to introduce some examples of PLGS errors, corresponding to the cases described in subsection 5.3.7. These examples are from the matching results of Dataset 1.

1. Case 1 (shown in Table 5-10): From Table 5-10, we can see that PLGS assigns '0_2712_FPMGK_None_PepFrag1_579_2' to three ions appearing in different ion groups. The RT values of these ions are 20.1799, 20.8287 and 20.4866 respectively. Hence the differences in RT are larger than the threshold of 0.11 mins. PLGS may have made mistakes in assigning the same identity to these ions whose RT difference is significant.

Table 5-10 Examples of Case 1

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6

Ion group 1: Peptide Identity					0_2712_FPMGK_None_PepFrag1_579_2	###
MZ	0	0	0	0	290.1511	290.1514
DT	0	0	0	0	31.702	31.821
RT	0	0	0	0	20.1799	20.165
Counts	0	0	0	0	2563	2200
Ion group 2: Peptide Identity			###	0_2712_FPMGK_None_PepFrag1_579_2	###	###
MZ	0	0	290.1509	290.1521	290.1526	290.1509
DT	0	0	32.078	32.111	32.043	32.081
RT	0	0	20.7872	20.8287	20.7276	20.7467
Counts	0	0	1130	1004	2899	2580
Ion group 3: Peptide Identity			###	###	###	0_2712_FPMGK_None_PepFrag1_579_2
MZ	0	0	290.1519	290.151	290.1523	290.1516
DT	0	0	32.123	32.167	32.055	32.042
RT	0	0	20.5519	20.5268	20.4952	20.4866
Counts	0	0	1312	1342	3606	3856

2. Case 2 (shown in Table 5-11): From Table 5-11, we can observe that each ion group contains ions of two different peptide identities. For example, ion group 1 contains two ions with the identify of '0_3845_IADAVK_None_PepFrag1_616_2' and one ion with the identity of '0_5097_LNELK_None_PepFrag1_616_2'. Although these ions have different identities, their 3D values are quite close. In this case, we cannot speculate that our matching method has mis-grouped ions of different identities.

Table 5-11 Examples of Case 2

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Ion group 1: Peptide Identity	0_3845_IADAVK_None_PepFrag1_616_2	###	###	###	0_3845_IADAVK_None_PepFrag1_616_2	0_5097_LNELK_None_PepFrag1_616_2
MZ	308.6892	308.6889	308.6884	308.6883	308.6886	308.6892
DT	32.825	32.734	32.66	32.756	32.755	32.693

RT	19.5382	19.5279	19.5384	19.5312	19.4937	19.4917
Counts	6226	5285	5643	5940	10288	9777
Ion group 2: Peptide Identity	1_3845_IADAVK_None_PepFrag1_616_2	###	###	###	1_3845_IADAVK_None_PepFrag1_616_2	1_5097_LNELK_None_PepFrag1_616_2
MZ	309.1892	309.1897	309.1898	309.1897	309.1886	309.1896
DT	32.799	32.789	32.741	32.784	32.715	32.712
RT	19.5396	19.521	19.5331	19.5341	19.4966	19.4941
Counts	6226	5285	5643	5940	10288	9777
Ion group 3: Peptide Identity	2_3845_IADAVK_None_PepFrag1_616_2	###	###	###	2_3845_IADAVK_None_PepFrag1_616_2	2_5097_LNELK_None_PepFrag1_616_2
MZ	309.689	309.6905	309.6907	309.6906	309.688	309.6894
DT	32.839	32.663	32.794	32.828	32.727	32.763
RT	19.5469	19.5342	19.5451	19.5298	19.4963	19.4879
Counts	6226	5285	5643	5940	10288	9777

3. Case 3 (shown in Table 5-12). In Table 5-12, there are two ion groups and each one contains ions with two different peptide identities. We can see that in ion group1 two ions with the peptide identity of '0_2934_NIIGIK_None_PepFrag1_657_2' have RT values around 26.65, which are quite close to the other ions in this group. In ion group2, there are four ions with the peptide identity of '0_2934_NIIGIK_None_PepFrag1_657_2'. Their RT values are around 24.13, which are close to other ions in ion group2. However, the difference between 24.13 and 26.65 exceeds the RT threshold of 0.11 min. Moreover, the counts of these ions with the peptide identity of '0_2934_NIIGIK_None_PepFrag1_657_2' in ion group 1 and ion group 2 are quite different. Therefore, it is quite likely that PLGS assigns wrong identities to these ions.

Table 5-12 Examples of Case 3

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Ion group 1: Peptide Identity	0_2934_NIIGIK_None_PepFrag1_657_2	0_2879_NILGLK_None_PepFrag1_657_2	0_2879_NILGLK_None_PepFrag1_657_2	0_2879_NILGLK_None_PepFrag1_657_2	0_2879_NILGLK_None_PepFrag1_657_2	0_2934_NIIGIK_None_PepFrag1_657_2

MZ	329.2189	329.2184	329.2191	329.2191	329.219	329.2197
DT	35.695	35.653	35.706	35.676	35.557	35.4
RT	26.6828	26.673	26.684	26.681	26.6461	26.6488
Counts	3418	3357	3139	2597	8097	7430
Ion group 2: Peptide Identity	0_2879_NIL GLK_None_ PepFrag1_65 7_2	0_2934_NII GIK_None_P epFrag1_657 _2	0_2934_NII GIK_None_P epFrag1_657 _2	0_2934_NII GIK_None_P epFrag1_657 _2	0_2934_NII GIK_None_P epFrag1_657 _2	0_2879_NIL GLK_None_ PepFrag1_65 7_2
MZ	329.2191	329.2186	329.2203	329.2175	329.2188	329.218
DT	34.653	34.593	34.432	34.467	34.356	34.264
RT	24.1463	24.142	24.1445	24.1342	24.1226	24.114
Counts	818	756	887	925	1641	1639

5.4.3.6 Batch effect handling

In the last part of the result section, we work on Dataset 4 to show the performance of our method when batch effect is at presence. We first directly apply the ion matching method to process all samples in Dataset 4 in one go and record the performance (run 1). Then we apply the approach in section 5.3.6 to handle the batch effect and also record the performance (run 2). Table 5-13 compares the performance of run 1 and run 2. We can find that the results of run 2 have significantly smaller mis-matching rates. It demonstrates that processing samples batch by batch and integrate results using batch means can greatly increase the performance. Therefore, using this approach our method can handle batch effect effectively.

Table 5-13 A summary of the matching results for Dataset 4

	Run 1	**Run 2**
P1	0.98	0.99
P2	0.64	0.87
P3	0.76	0.93
P4	0.77	0.93
P5	0.78	0.94
P6	0.33	0.3
# matching rows used for performance evaluation	4485	3920

5.5 Discussion

The matching algorithm presented in this chapter uses 3D information to group similar ions across samples. As each DRAMI file contains hundreds of thousands of low energy (precursor) ions, matching all ions across all samples concurrently computationally intensive, particularly for large sample sets. Thus the first step in our ion matching algorithm is data binning. As the MZ measurement has the smallest variation, the binning process is carried out in the MZ direction. Considering the case that the number of input files grows large and a large amount of ions dropped in a MZ bin, we introduce an extra binning scheme that ions in the MZ bin will be further binned in the DT direction. The DT direction is chosen instead of RT since DT is less variable than RT. The core of matching algorithm is based on the hierarchical clustering. We choose average linkage to generate cluster tree. This is because the distance between any two ion clusters is calculated as the mean distance between ions of each cluster. Therefore, similar ions are more likely to be grouped into a single cluster. In the algorithm, we use overlapping bins and each ion will appear twice in neighbouring bins, hence we develop an adjustment method to remove duplicates. The influence of arbitrary splitting boundary is reduced using our method. To accelerate the execution of the algorithm, the whole dataset is split into pieces that each parallel worker in MATLAB can process its individual piece without interfering others. Moreover, MATLAB only needs to load a portion of the whole large dataset into memory which saves execution time as well.

We have highlighted the accuracy of our approach, and include in our output a mismatching summary which enables the user to evaluate performance. We have shown that the majority of errors in the output, when error is defined through software assigned peptide identities onto ions, are as a result of false discovery of those identities rather than through an error in our approach.

In our method, we choose the binning size fully dependent on the characteristic of spectra. Users can set different values of binning size. Different settings of binning size can greatly influence the performance of our method. For example, if we double the binning size, then the number of mismatches can be double as well. This is one limitation of our method, which is not fully parameter free. However, as we have shown in the experiment part, by setting the binning size using features from the PLGS APEX 3D files we can get promising results.

Isoquant is also an unbiased data-independent acquisition mass spectrometry method, utilizing MZ, RT and DT information to do ion matching. Similar to our method, Isoquant

also uses clustering method to group similar ions. However, before clustering it samples MZ, RT and DT values using instrument-dependent mass p.p.m resolution value, user defined retention time and drift-time resolution. Then density-based clustering method is applied on the sampled 3D space. The behaviour of clustering is controlled by user-defined number of neighbour points for clustering expansion inside a neighbourhood radius and its size. Different from our method, Isoquant introduces feature space sampling and much more user defined parameters. Isoquant annotates ion groups and discards ion groups containing ions with no unique peptide assignment. Our program, in contrast, records all matched ion groups and informs the users the assignment of peptide identities. Users using our program can know whether mismatching is caused by our program or by PLGS using our built-in evaluation functions. It has been mentioned in the Isoquant website that one limitation of Isoquant is that the processing time increase nonlinearly with the increasing size of the dataset. In our case, this problem is alleviated by introducing MZ binning and further splitting MZ bins in the DT direction when large number of ions falling in a MZ bin.

Our approach has the potential to have a large impact in the area of biomarker discovery, particularly in the analysis of large numbers of experimental samples via data independent acquisition. The approach is able to dramatically increase the available protein space with complete coverage, meaning that statistical analyses of data will be more robust and the application of many sophisticated data mining methodologies is possible, without the negative effect of imputation strategies on native data structure. The nature of the approach, in which discriminating features are discovered through data mining of ion information, results in a greatly reduced feature space, which means those features can be interrogated for identity in an expanded search space without massively inflated false discovery rates. An added benefit of this is that experimentation need not be constrained within the traditional proteomic space but can also access proteome, peptidome and metabolome features in a single analysis.

We are currently applying this approach to MS^E data gathered for the UBIOPRED severe asthma program, including data from immune-depleted serum for a cohort of 350 adults, and induced sputum supernatant for 270 individuals. Moreover, we are carrying out more sophisticated analysis on Dataset 3. We first apply the topological data analysis (Zomorodian 2011) to find groups of samples. Then significantly discriminating features are selected. We initially interrogate the Ion accounting files to determine if these features have peptide identities. As expected many of the ions showing differential abundance mapped to peptides

from haemoglobin ($p < 5.1E-5$). Additionally we see differential abundance in peptides from a number of other high abundance serum proteins including A1AT, A2MG, TRFE and THRB. These proteins are likely showing differential abundance due to effects of dilution when preparing samples. We next investigate the discriminating ion features containing no peptide information. By following an expanded database strategy and we are identifying a number of non-standard peptide features that are consistently identified across all samples. This analysis is still being undertaken, as it requires a lot of manual work from biologists. Still, we can expect promising results based on the assessment we have done on Dataset 1 and Dataset 2. We find differential peptide features between samples of Dataset 1 and Dataset 2 from the results of our method and traditional approach (identity known). If we regard peptides found by traditional approach are real positives, then the false negative rate is 35% and the false positive rate is close to zero. The real false negative rate should be even much smaller than this value, as we find that a lot of positives regarded as real are in fact not real.

In summary, this chapter introduces a clustering based method to match similar identities across samples in quantitative Proteomics measurements. The key idea is using hierarchical clustering to automatically group similar measurements. Considering high dimensionality of the measurement dataset, the matching is sequentially carried out bin by bin. A post-matching processing step removes the effect caused by binning. Our pipeline provides in-built evaluation functions to automatically evaluate the performance of matching. Thus, users can easily acquire the quality of matching and also possible errors caused by machine.

6 Obtaining classifiers: a cross validation based approach

6.1 Motivation

As we have discussed in section 3.4, classification can be done by using machine learning methods to fit the quantitative dataset into a predictive model. The input dataset contains observations whose category membership is known. Then the model can identify which of a set of categories that a new coming observation falls into. A popular approach to select classification model is using cross validation (CV) methods. Cross validation partitions the training dataset into complementary subsets, performs the analysis on one subset (i.e., training dataset), and validate the analysis on the other subset (i.e. validation set). To avoid inefficient evaluation, multiple rounds of cross-validation are performed with different ways of partitions. The validation results are averaged over rounds. The process of CV is shown in Figure 6-1.

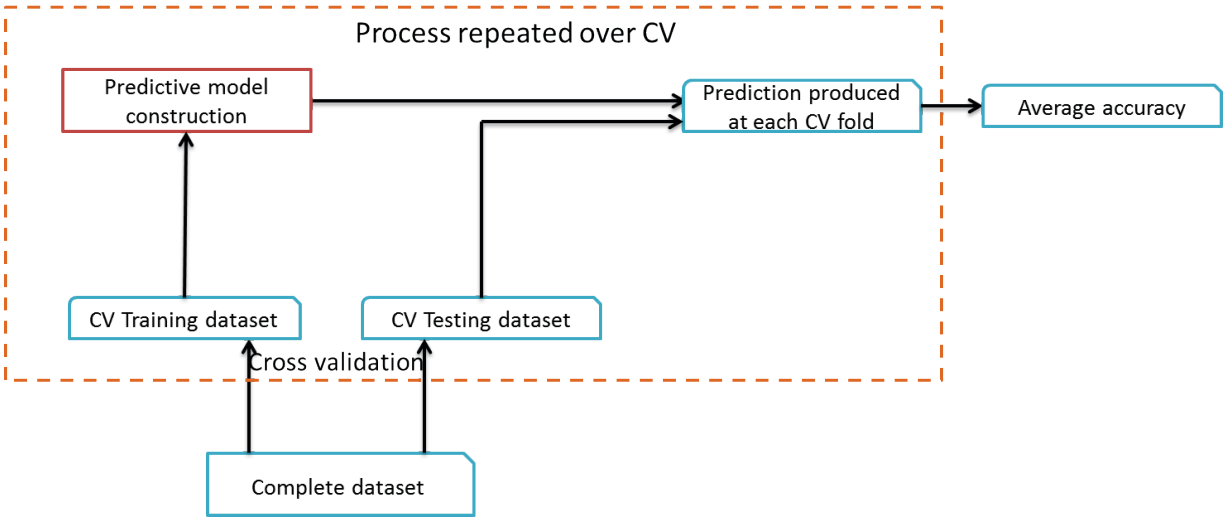


Figure 6-1 The process of cross validation for classification

A common issue in accurate model construction is that the number of features is usually much larger than the number of samples. For example, in UBIOPRED Proteomics analysis, each sample file contains millions of ions (features) while the number of samples is less than a thousand. Similarly, in fMRI analysis, each fMRI image can also contain up to millions of voxels (features). As the number of features is thousand times of the number of samples, the model may easily overfit into the dataset.

To overcome this dimensionality problem, feature selection is commonly used to reduce the feature space. One straightforward way is applying a statistic test, such as t-test, Kolmogorov–Smirnov test (Arnold & Emerson 2011) and permutation test, to select

discriminant features. The whole dataset is split into different groups, which can be, for example, determined by different disease types, stimuli, and treatments. Statistic tests check features one by one to see whether the distributions of each feature in different groups are significantly different. Only the features whose p-values are smaller than the significant level (e.g., 0.05) will be used in the model construction. One limitation of this filter approach is that features are investigated separately. It is often the case that some individual features are not discriminant but have strong predictive power when they join together. Another limitation is that the number of selected features depends on the value of the significant level. Although it is convention to set the significant level to be 0.05 or 0.01, we can hardly say any features whose p-value is slightly larger than this value do not have discriminant power.

Thus it comes to the idea of embedding the feature selection into a prediction model. A typical example is the Lasso (Tibshirani 1994) method for constructing a linear model, which penalises the regression coefficients, shrinking many of them to zero. Another example is applying linear sparse Bayesian learning (Tipping 2001) to construct a predictive model and rank features by their weights. By recursively removing features with relatively low weights, we can determine the optimal number of features with best performance. The process of this embedded feature selection approach is shown in Figure 6-2.

There are various classification algorithms, including linear classifiers (e.g., Fisher's linear discriminant (Fisher 1936), Logistic regression (Cox 1958), Naïve Bayes classifier (Rish 2001)), Support Vector Machine (Cortes & Vapnik 1995), decision trees (Quinlan 1987), random forests (Breiman 2001), Neural networks (Bishop 1995) and Relevance vector machine (Tipping 2001). It is never easy to say which method can always perform better than the others. Thus, it is worthwhile to try different classification algorithms to see which one has the best performance for your dataset. Similarly, we can also try different feature selection methods as well, including Lasso, elastic net (Zou & Hastie 2005) and sparse Bayesian learning. So the process of predictive model construction is repeated with different combination of feature selection and machine learning method. In this chapter, we focus on discussing the case when the machine learning and feature selection methods have already been chosen.

The process in Figure 6-2 cannot generate a single prediction model for future sample prediction. This is because the model and the selected features are different for different training datasets. Generating a single and robust model with an optimal feature set is quite

important as the ultimate goal is building a model that is not only sensitive to the input dataset but can work well for an independent validation dataset. Obtaining a single optimal feature set is therefore of significance. The feature set contains the most relevant features for prediction model. The selected features, such as biomarkers, can indicate some biological states or conditions.

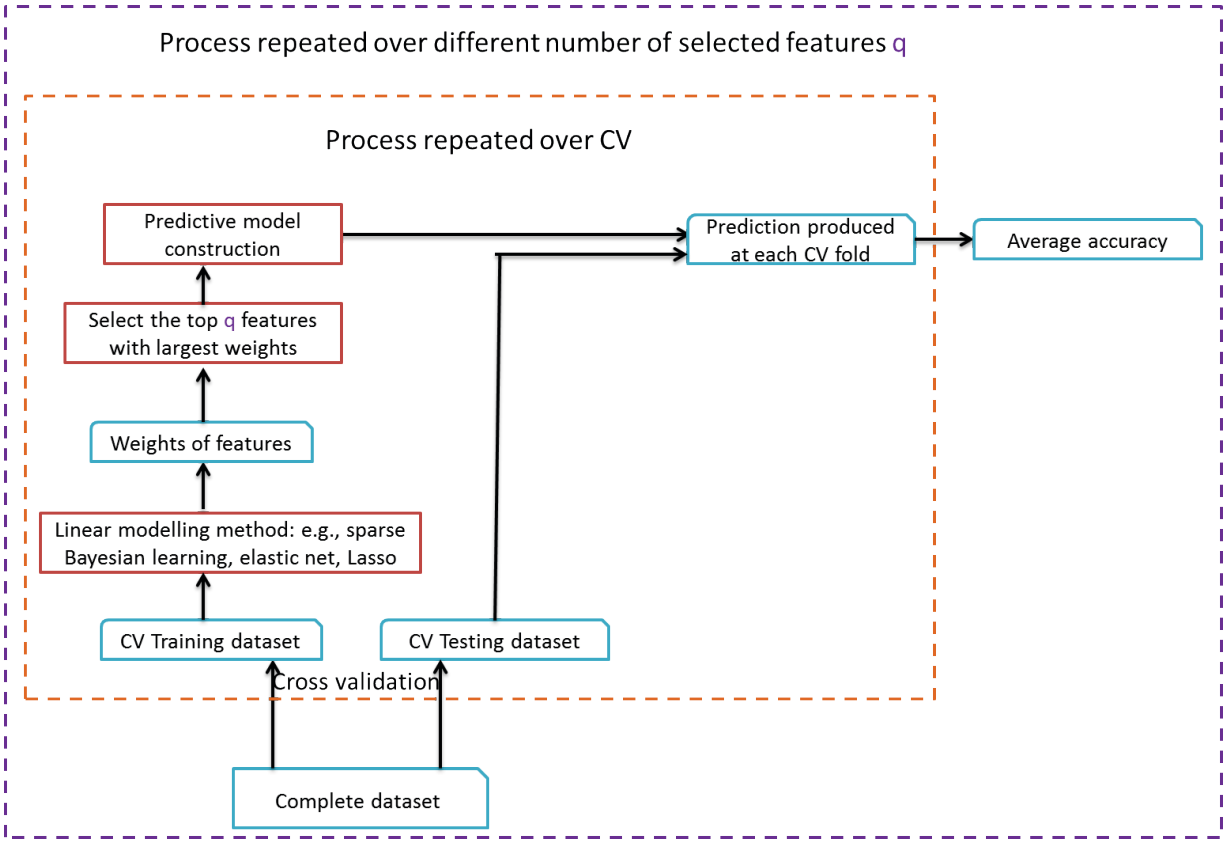


Figure 6-2 The process of cross validation with the embedded feature selection approach

One way to determine an optimal set is checking its stability. An ideally stable feature should be consistently chosen under different training datasets. There are increasing literature discussing the stability of feature selection (Abeel et al. 2009; Zucknick et al. 2008; Ahmed et al. 2011; Alexander & Lange 2011). The general way to assess the stability is: 1) subsample the whole datasets; 2) construct classifier for each subset; 3) select features for each classifier; 4) assess the agreement among the resulting sets of selections.

It has been pointed out that the stability of the feature selection itself alone is not meaningful, which should be always assessed together with predictive performance. A pipeline to balance the predictive performance and stability of feature selection has been proposed in (Kirk et al. 2013). In this chapter, we adopt this pipeline for fMRI analysis. Details of this approach will

be introduced in the later section. We also incorporates a feature selection algorithm which we proposed in (Yan, Yang, Wu, Guo, et al. 2014). We will show that our method works well in terms of accuracy and stability.

6.2 Background

A popular brain imaging technique, functional MRI (fMRI), measures the changes of brain blood flow signal (BOLD) occurred during the completion of cognitive tasks. Multivariate Pattern Analysis (MVPA) in fMRI decodes patterns of brain activity by constructing predictive models using signals from multiple voxels of fMRI images (Norman et al. 2006). An enormous range of classification methods can be used in MVPA studies and the most popular category is linear classifiers, such as linear discriminant analysis (Carlson et al. 2003; O'Toole et al. 2005; J.-D. Haynes & Rees 2005; J. D. Haynes & Rees 2005), neural networks with no hidden layer (Polyn et al. 2005), support vector machine with linear kernel (SVM) (Motoda & Fawcett 2004; Kamitani & Tong 2005; O'Toole et al. 2005) and Gaussian Naïve Bayes classifiers (Motoda & Fawcett 2004). All linear classifiers are constructed by a weighted linear combination of features. There are also some non-linear classifiers used in MVPA, including nonlinear SVM (Cox & Savoy 2003; Davatzikos et al. 2005) and neural networks with hidden layer (Hanson et al. 2004). However, as it has been discussed in (Cox & Savoy 2003; Kamitani & Tong 2005) that the benefits of using non-linear ones in MVPA are not clear, our chapter focuses on using linear classifiers. An illustration of the MVPA study is shown in Figure 6-3 (from (Norman et al. 2006)).

In MVPA, the number of voxels is usually several orders of magnitudes larger than samples. Due to the discrepancy between the sample and feature size, classifiers easily become overfitting to the training dataset. Therefore, feature selection methods are used beforehand to reduce the feature space. Here, we adopt the random subspace sparse Bayesian learning (RS-SBL) method that we developed in (Yan, Yang, Wu, Guo, et al. 2014) to solve the dimensionality problem. Sparse Bayesian learning (SBL) can return a sparse solution of linear model (Tipping 2001), from which relevant voxels can be first selected. Random subspace (RS) (Tao et al. 2006) can improve the performance of SBL by randomly generating multiple sets of subspaces and aggregating results of them all. In (Yan, Yang, Wu, Guo, et al. 2014), we have demonstrated the RS-SBL method works well in MVPA in the aspect of predictive accuracy. In this chapter, we use this method as a feature selection approach.

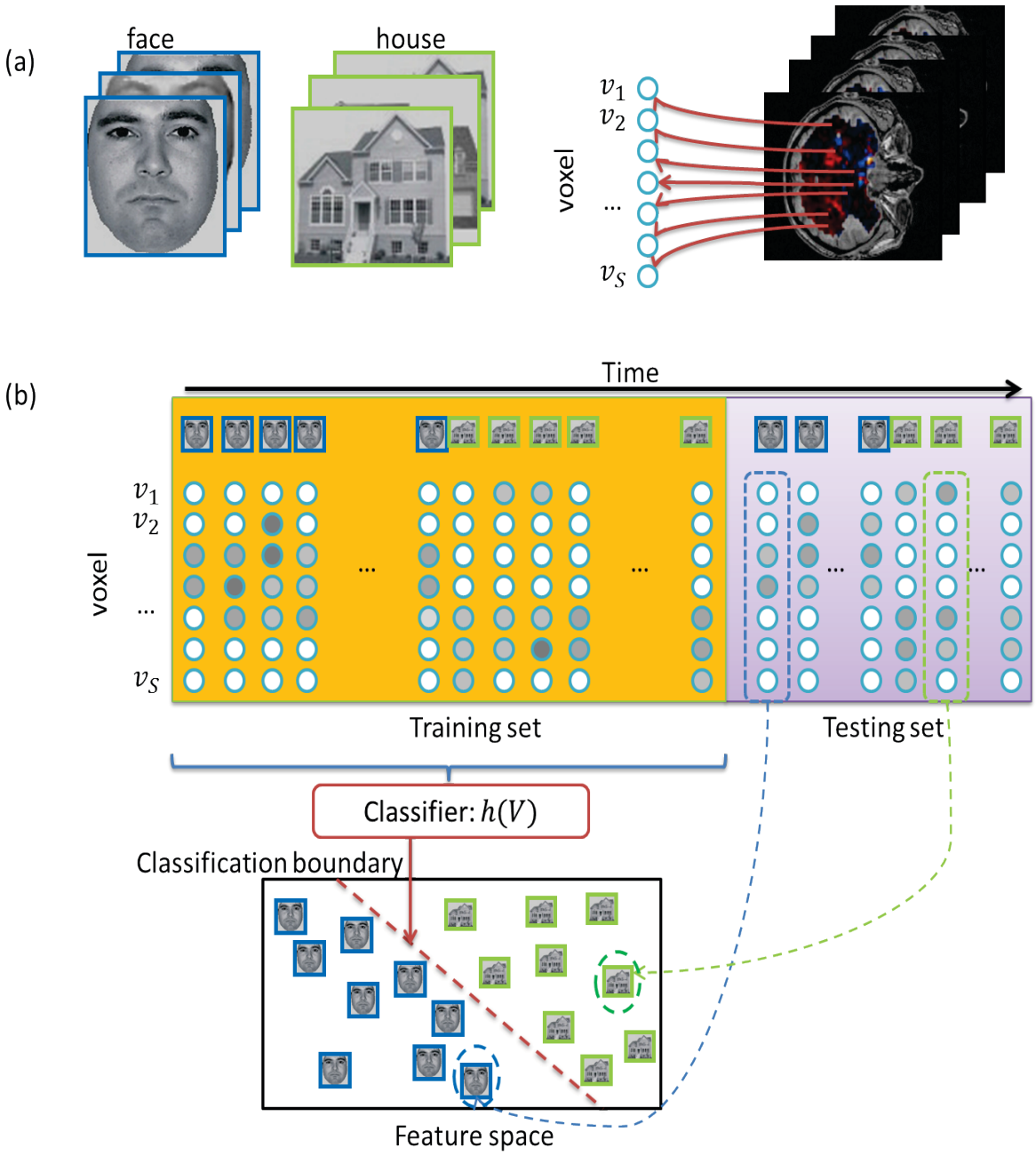


Figure 6-3 An illustration of the MVPA study (from (Norman et al. 2006)). a) Subjects view stimuli from two categories: faces and houses. fMRI images contain multiple voxels, which are features for classification. b) The fMRI time series is decomposed into training and testing parts. The brain pattern at a specific time point is represented by values of voxels, corresponding to experimental condition (face vs. house). Classifiers work on the training dataset to find the relation between brain pattern and experimental condition. The output of classifiers is a classification boundary, with which we can predict the experimental condition for each brain pattern from the testing dataset.

Cross validation is a common approach to evaluate the performance of the predictive model. In the cross validation process, different training dataset is used for feature selection and predictive model construction in each fold. Therefore, selected feature sets from all folds may vary due to the variation of training datasets. A recent trend in MVPA is generating a predictive map as well as a classifier (Norman et al. 2006). Predictive map contains relevant voxels used for prediction, which can indicate which brain regions participate in a functional brain activity. It is therefore meaningful that such a predictive map is robust with respect to different training data. A method to return a feature set that is consistently used in all folds of cross validation is of great demand. This is regarded as the feature stability problem.

In this chapter, to generate a meaningful and stable predictive map as well as having predictive power in MVPA, we develop a pipeline where the method in (Kirk et al. 2013) is adopted and the RS-SBL feature selection method is incorporated. This is a novel MVPA pipeline bringing the stability into consideration. In the following parts of this chapter, we will first explain this pipeline in detail. Then a simulation dataset and a real fMRI dataset are used to demonstrate our method. A summary of our observations is described in the discussion part.

6.3 Method

6.3.1 Overview of our method

Our aim is to generate a feature set, with which the predictive accuracy of the classifier is maintained at a high level. In the meanwhile, selected features are stable with small fluctuations of the input dataset. The key idea is using cross validation method to select features which are both consistent across all folds of cross validation and can jointly contribute to an accurate prediction. The flowchart of the proposed method is shown in Figure 6-4. There are two loops in the model optimisation process. The inner loop is a K-fold cross validation (CV) process to find the optimal feature set with a given value of q, where q is the predefined number of features in the model. The outer loop is to find the optimal value of q by repeating the CV process with different q.

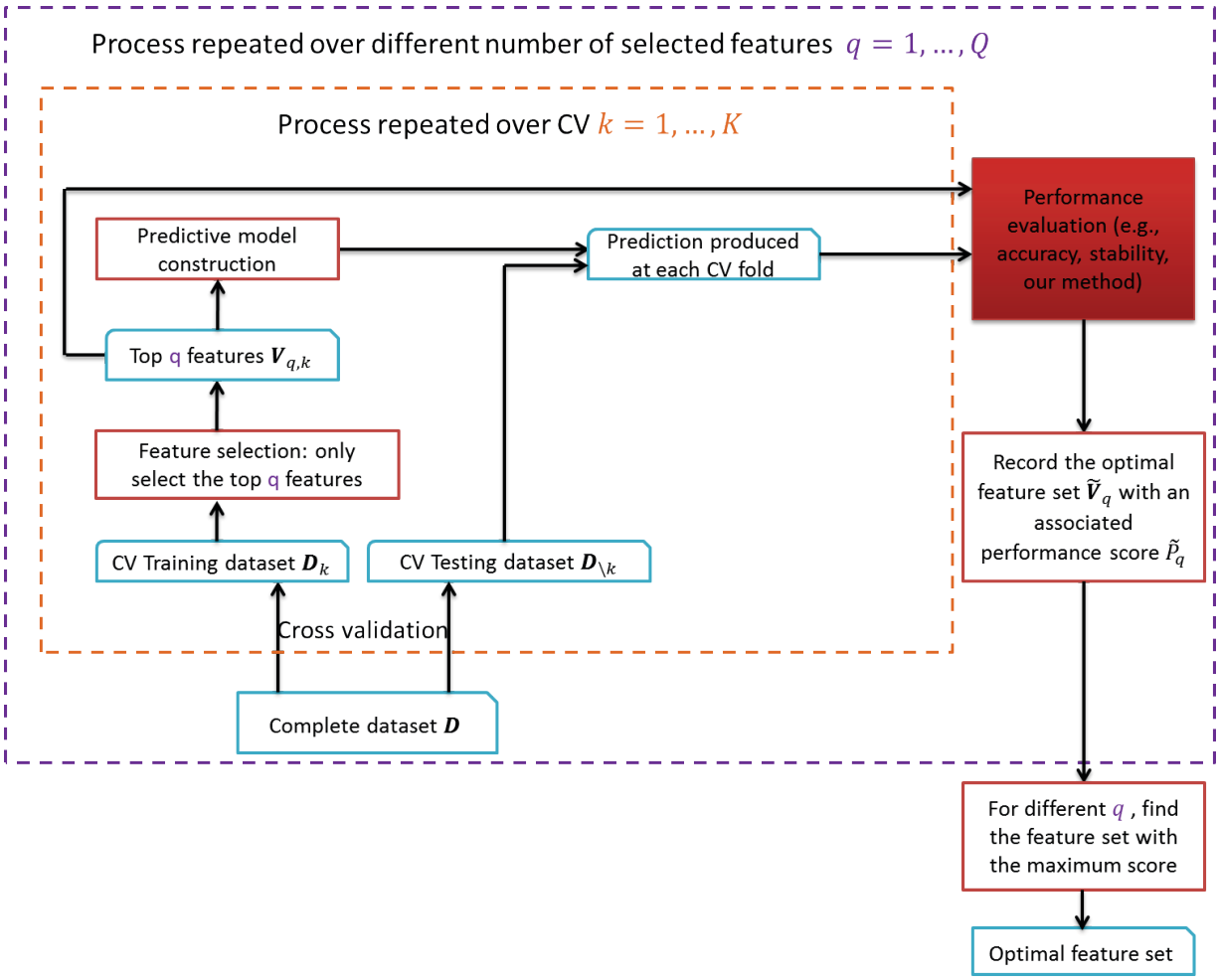


Figure 6-4 The flowchart of optimal feature selection through cross validation

In the kth fold of cross validation, the whole dataset $\boldsymbol{D}$ is split into two subsets: CV training and CV testing datasets, which are denoted as $\boldsymbol{D}_k$ and $\boldsymbol{D}_{\backslash k}$. Feature selection methods can work on the training dataset $\boldsymbol{D}_k$ to rank and select the top q features, labelled as $\boldsymbol{V}_{q,k}$. After features have been selected, a predictive model for classification is constructed using $\boldsymbol{V}_{q,k}$. Various machine learning methods can be used, such as support vector machine, decision tree and neural network. The prediction results at this CV fold are recorded for later evaluation.

To get the complete prediction results, we repeat the above steps for all folds of CV. There are different ways to represent the performance, including the accuracy of the predictive model and the stability of selected features. The method we use to balance these two metrics will be described in subsection 6.3.4. It will return an optimal feature set $\tilde{\boldsymbol{V}}_q$ with an associated performance score $\tilde{P}_q$. With the varying value of q, we can get a set of $\tilde{\boldsymbol{V}}_q$ and $\tilde{P}_q$ for $q \in 1, \dots Q$. By checking the maximum value of $\tilde{P}_q$, we can determine the optimal value of

q and the corresponding optimal feature set $\widetilde{\boldsymbol{V}}_q$. The detected optimal set can be used to construct a predictive model for future prediction.

Please note that the proposed pipeline can be used for different combinations of feature selection and classification algorithms. In this chapter, our focus is on generating a stable and predictive feature set when the feature selection and classification algorithms have been chosen in advance. If we want to get models from different algorithms, we can simply apply our pipeline many times for various algorithms. The feature selection method chosen in this chapter, is our previously developed method, called random subspace sparse Bayesian learning (Yan, Yang, Wu, Guo, et al. 2014). RS-SBL incorporates random subspace method to improve the performance of sparse Bayesian learning (SBL). SBL can be in the same form of the support vector machine but uses Bayesian inference to provide probabilistic classification. Compared to SVM, SBL avoids the set of free parameters of the SVM. The details of this method will be introduced in subsection 6.3.3. The machine learning method used for classification in this chapter is also sparse Bayesian learning. In subsection 6.3.2, we will briefly explain the SBL method.

6.3.2 Preliminaries of SBL

SBL is used twice in our pipeline: the feature selection method incorporates SBL and also classification model is constructed by the SBL method. Let us first briefly review the SBL method for the regression problem. With a given set of input vectors $\{\mathbf{x}_n\}_{n=1}^N$ along with corresponding targets $\{t_n\}_{n=1}^N$, we wish to learn the underlying functional mapping which is defined by a parameterised function $\mathrm{y}(\mathbf{x};\mathbf{w})$ (Tipping 2006):

$$\mathrm{y}(\mathbf{x};\mathbf{w}) = \sum_{j=1}^{J} w_j \phi_j(\mathbf{x}) \tag{6.1}$$

where the output is the linear weighted sum of J basis functions. Usually, the prior distribution of the weights, $p(\mathbf{w}|\boldsymbol{\alpha})$, is assumed to follow a zero-mean isotropic Gaussian:

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{j=1}^{J} N\left(w_j|0,\alpha_j^{-1}\right) \\ &= \prod_{j=1}^{J} \left(\frac{\alpha_j}{2\pi}\right)^{\frac{1}{2}} \exp\{-\frac{\alpha}{2} w_j^2\} \end{aligned} \tag{6.2}$$

where $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^J$ contains the inverse variance of $\mathbf{w}$.

The output data t_n can be regarded as a noisy realisation of the model as:

$$t_n = \mathrm{y}(\mathbf{x}_n; \mathbf{w}) + \epsilon_n \tag{6.3}$$

where the noise component ϵ_n is assumed to follow a Gaussian distribution $\epsilon_n \sim n(0, \sigma^2)$. Therefore, the likelihood of all the data $\mathbf{t} = (t_1, t_2, \ldots t_N)'$ is of the form:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) &= \prod_{n=1}^{N} (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{\{t_n - y(\mathbf{x}_n; \mathbf{w})\}^2}{2\sigma^2}\right] \\ &= \prod_{n=1}^{N} (2\pi\sigma^2)^{-1/2} \exp\left[-\frac{\left||\boldsymbol{t} - \boldsymbol{\Phi w}|\right|^2}{2\sigma^2}\right] \end{aligned} \tag{6.4}$$

where $\boldsymbol{\Phi}$ is the $N \times J$ design matrix with $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]'$, wherein $\boldsymbol{\phi}(\mathbf{x}_n) = (\phi_1(\mathbf{x}_n), \phi_2(\mathbf{x}_n), \ldots, \phi_J(\mathbf{x}_n))'$.

As $p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2)$ and $p(\mathbf{w}|\boldsymbol{\alpha})$ are both Gaussian, the posterior distribution is also Gaussian (Berger 1985). We can get $p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\begin{aligned} \boldsymbol{\mu} &= \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}'\mathbf{t} \\ \boldsymbol{\Sigma} &= (\sigma^{-2}\boldsymbol{\Phi}'\boldsymbol{\Phi} + \boldsymbol{A})^{-\mathbf{1}} \end{aligned} \tag{6.5}$$

where $\boldsymbol{A} = diag(\alpha_1, \ldots, \alpha_J)$.

SBL follows an expectation-maximisation manner to iteratively update parameters $\boldsymbol{\alpha}$ and σ^2. The iterative re-estimation formulae are:

$$\alpha_j^{new} = \frac{\gamma_j}{\mu_j^2} \tag{6.6}$$

and

$$(\sigma^2)^{new} = \frac{\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|^2}{N - \sum_{j=1}^{J} \gamma_j} \tag{6.7}$$

where $\gamma_j = 1 - \alpha_j \boldsymbol{\Sigma}_{j,j}$ and $\boldsymbol{\Sigma}_{j,j}$ is the element in the j^{th} row and j^{th} column of the covariance matrix $\boldsymbol{\Sigma}$ (MacKay 1992). The derivation steps of these equations will be shown in Chapter 8.

In our pipeline, SBL is used for the classification problem. The classification follows an essentially identical framework as for regression, except the target variable is binary, that is $t_n \in \{0,1\}$. To adapt the change in the target variable, the Bernoulli likelihood and logistic sigmoidal link function are applied. The likelihood is updated as

$$p(\mathbf{t}|\mathbf{x},\mathbf{w}) = \prod_{n=1}^{N} S\{y(\mathbf{x}_n;\mathbf{w})\}^{t_n}[1 - S\{y(\mathbf{x}_n;\mathbf{w})\}]^{1-t_n} \quad 6.8$$

where $S\{\cdot\}$ is the sigmoidal function in the form of $S\{y\} = 1/(1+e^{-y})$. Note that there is no noise in the likelihood.

Unlike the regression problem, there are no closed-form expressions for the weight posterior $p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha})$. Therefore, an iterative process to approximate $\mathbf{w}$ is introduced in (Tipping 2001), which is based on Laplace's method. For the current estimated value of $\boldsymbol{\alpha}$, the most probable weights, $\mathbf{w}_{MP}$ is obtained by finding the maximum of $p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha})$, which gives the mode (i.e. most probable) of the posterior distribution. As $p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha}) \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha})$, finding the maximum of $p(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha})$ is equivalent to maximizing the following object function:

$$F(\mathbf{w}) = \ln p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\boldsymbol{\alpha}) = \sum_{n=1}^{N}[t_n \ln \mathrm{y}_n + (1-t_n)\ln(1-\mathrm{y}_n)] - \frac{1}{2}\mathbf{w}'\boldsymbol{A}\boldsymbol{w} \quad 6.9$$

with $\mathrm{y}_n = S\{y(\mathbf{x}_n;\mathbf{w})\}$. This is a standard procedure, since Equation 6.9 is a penalised logistic log-likelihood function, and necessitates iterative maximization. Various optimization methods can be used to obtain $\mathbf{w}_{MP}$, where in (Tipping 2001) 'iterative reweighted least-squares' (Rubin 1983) is adapted to find $\mathbf{w}_{MP}$.

Laplace's method approximates the log-posterior distribution around its mode $\mathbf{w}_{MP}$ using low order Taylor expansion, which is equivalent to a local Gaussian approximation around $\mathbf{w}_{MP}$ (Friston et al. 2007). The covariance matrix is obtained via

$$\boldsymbol{\Sigma} = -\boldsymbol{H}|_{\mathbf{w}_{MP}}{}^{-1} \quad 6.10$$

where $\boldsymbol{H}|_{\mathbf{w}_{MP}} = \nabla^2_{\mathbf{w}}(F(\mathbf{w}_{MP}))$ is the second order derivative with respect to $\mathbf{w}$ at $\mathbf{w}_{MP}$, which is regarded as a Hessian matrix. To get $\boldsymbol{H}$, we first derive the gradient of the objective function $F(\mathbf{w})$ as

$$\begin{aligned}
\nabla_{\mathbf{w}}F(\mathbf{w}) &= -\boldsymbol{A}\boldsymbol{w} + \sum_{n=1}^{N}\nabla_{\mathbf{w}}[t_n \ln \mathrm{y}_n + (1-t_n)\ln(1-\mathrm{y}_n)] \\
&= -\boldsymbol{A}\boldsymbol{w} + \sum_{n=1}^{N}\nabla_{\mathbf{w}}[-t_n \ln\left(1+e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}\right) - (1-t_n)\ln(1+e^{\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)})] \quad 6.11 \\
&= -\boldsymbol{A}\boldsymbol{w} + \sum_{n=1}^{N}[t_n \frac{\boldsymbol{\phi}(\mathbf{x}_n)e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}}{1+e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}} - (1-t_n)\frac{\boldsymbol{\phi}(\mathbf{x}_n)}{1+e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}}]
\end{aligned}$$

$$= -\boldsymbol{A}\boldsymbol{w} + \sum_{n=1}^{N} \frac{t_n \boldsymbol{\phi}(\mathbf{x}_n)(1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}) - \boldsymbol{\phi}(\mathbf{x}_n)}{1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}}$$

$$= -\boldsymbol{A}\boldsymbol{w} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n)(\mathrm{y}_n - t_n)$$

Let $\boldsymbol{Y} = (\mathrm{y}_1, \ldots, \mathrm{y}_n)'$, then Equation 6.11 can be turned into

$$\nabla_{\mathbf{w}} F(\mathbf{w}) = -\boldsymbol{A}\boldsymbol{w} - \boldsymbol{\Phi}'(\boldsymbol{Y} - \boldsymbol{t}) \tag{6.12}$$

Then the hessian of $F(\mathbf{w})$ is

$$\begin{aligned}
\boldsymbol{H} &= -\boldsymbol{A} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \nabla_{\mathbf{w}}(\mathrm{y}_n) \\
&= -\boldsymbol{A} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \nabla_{\mathbf{w}} \left(\frac{1}{1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}} \right) \\
&= -\boldsymbol{A} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \frac{\boldsymbol{\phi}(\mathbf{x}_n) e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}}{\left(1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}\right)^2} \\
&= -\boldsymbol{A} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \boldsymbol{\phi}(\mathbf{x}_n) \frac{1}{1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}} \frac{e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}}{1 + e^{-\mathbf{w}'\boldsymbol{\phi}(\mathbf{x}_n)}} \\
&= -\boldsymbol{A} - \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n) \boldsymbol{\phi}(\mathbf{x}_n) \mathrm{y}_n (1 - \mathrm{y}_n) \\
&= -\boldsymbol{A} - \boldsymbol{\Phi}' \boldsymbol{B} \boldsymbol{\Phi}
\end{aligned} \tag{6.13}$$

where $\boldsymbol{B} = diag(\beta_1, \beta_2, \ldots, \beta_N)$ is a diagonal matrix with $\beta_n = \mathrm{y}_n(1 - \mathrm{y}_n)$. According to Equation 6.10, the covariance matrix is

$$\boldsymbol{\Sigma} = (\boldsymbol{A} + \boldsymbol{\Phi}' \boldsymbol{B} \boldsymbol{\Phi})^{-1}|_{\mathbf{w}_{MP}} \tag{6.14}$$

Using $\boldsymbol{\Sigma}$ and $\mathbf{w}_{MP}$ (in place of $\boldsymbol{\mu}$), parameter $\boldsymbol{\alpha}$ is updated using Equation 6.6, which is in the same manner as for the regression case.

6.3.3 Feature selection algorithm

As we have mentioned, the feature selection algorithm chosen in this chapter is RS-SBL, which incorporates random subspace to sparse Bayesian learning. The SBL used in this step is linear where $\boldsymbol{\Phi} = \mathbf{x}$. Thus, the estimated weights can directly indicate the importance of features. For the fMRI data, signals from spatially neighbouring voxels are highly correlated. By introducing random subspace, correlated voxels within a single subspace for a SBL is largely reduced. The usage of multiple SBL for random subspaces gives the chance of selecting important features including correlated ones.

The flowchart of the RS-SBL feature selection method is shown in Figure 6-5. The input of this pipeline is the CV training dataset $\boldsymbol{D}_k = \{\mathbf{x}_k, \mathbf{t}_k\}$. In MVPA, matrix $\mathbf{x}_k \in R^{N\times S}$ contains the fMRI measurement, where S is the number of voxels and N is the number of samples in the training dataset. Let $\mathbf{t}_k \in R^{N\times 1}$ represent the corresponding brain state. As we only focus on binary classification, the brain state is encoded into binary digits, that is $t_{k,n} \in \{0,1\}$. Each voxel is regarded as a feature that can be used for prediction. The whole feature set is denoted as $\mathbf{V} = \{v_s\}_{s=1}^{S}$.

The random subspace method is first applied to generate L subspaces $\{\mathbf{x}_k^l\}_{l=1}^{L}$, each of which contains M features randomly sampled from the whole feature set $\mathbf{V}$, that is $\mathbf{x}_k^l \in R^{N\times M}$. Denote each subset of features as $\overline{\mathbf{V}}^l$. Then SBL constructs linear models using individual subspaces. Each model automatically returns a sparse feature space, where each feature corresponds to a non-zero element in the estimated linear model. Suppose the selected feature set in the SBL model of the lth subspace is $\overline{\overline{\boldsymbol{V}}}_l$. Then we can estimate the frequency of each feature being selected in all subspaces as:

$$\lambda_i = \frac{\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\overline{\boldsymbol{V}}}_l)}{\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\mathbf{V}}^l)} \text{ for } i \in \{1, \dots, S\} \tag{6.15}$$

$\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\mathbf{V}}^l)$ calculates the times of the feature v_i being sampled by L subspaces and $\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\overline{\boldsymbol{V}}}_l)$ estimates the times of the feature v_i being selected by SBL.

In order to eliminate the effects introduced by the randomness of subspace sampling, we repeat the RS-SBL process R times. The frequency of features being selected is averaged across repetitions. The top q features with the top q largest frequency values will be the output of this feature selection process. Let us denote the output as $\boldsymbol{V}_{q,k} = \{v_i\}_{i=1}^{q}$, where k indicates the kth fold of CV.

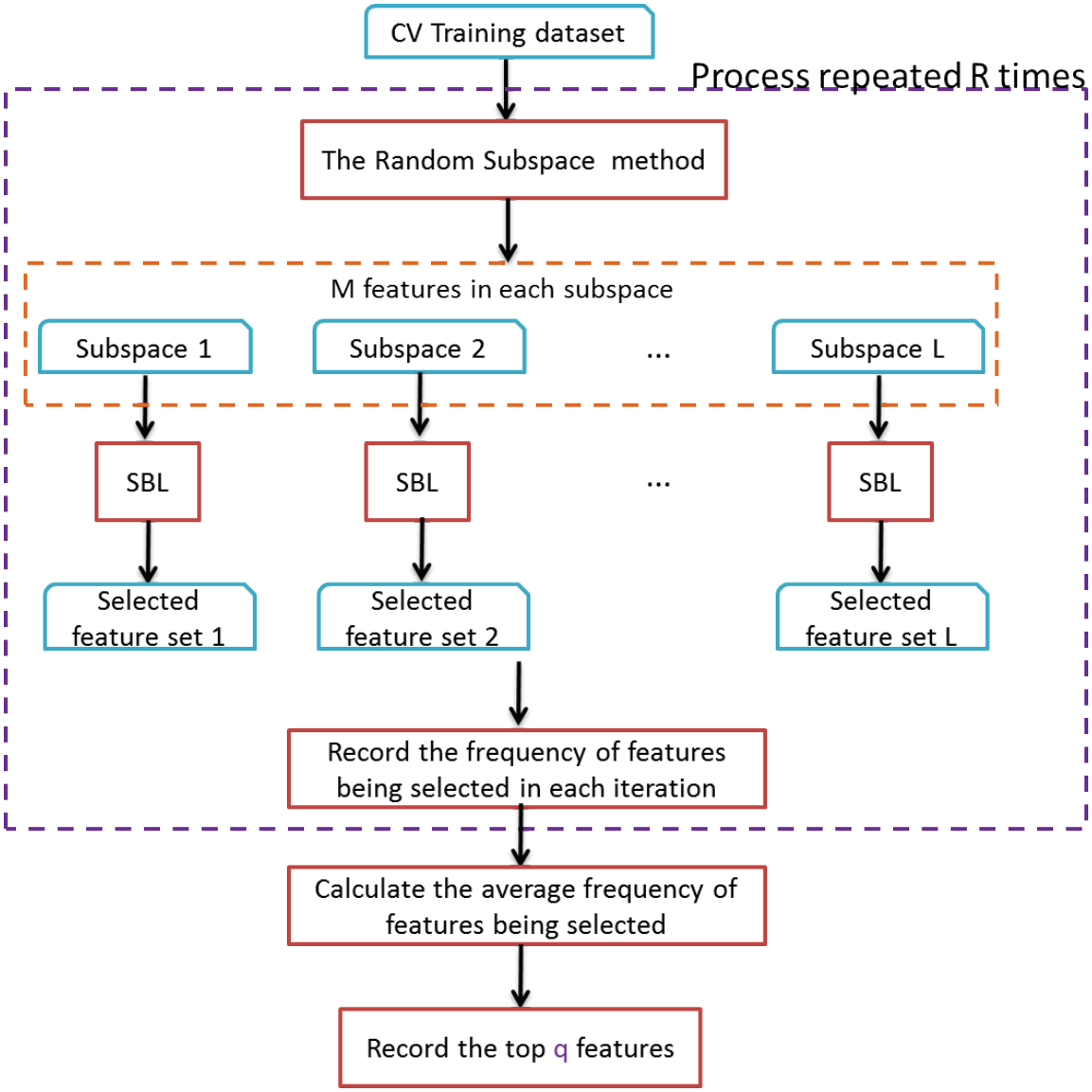


Figure 6-5 The flowchart of the RS-SBL feature selection algorithm

6.3.4 Performance evaluation method to balance stability and predictive performance

After the features have been selected, we use them to construct a predictive model, h(·). The model is then applied to the CV testing dataset, $\boldsymbol{D}_{\backslash k} = \{\mathbf{x}_{\backslash k}, \mathbf{t}_{\backslash k}\}$. Let $\mathrm{h}(\mathbf{x}_{\backslash k}) \in \{0,1\}$ represent the prediction result. Then various performance evaluation methods can be used as described in (Kirk et al. 2013).

The most straightforward way is evaluating the predictive power of the model by calculating the correct classification rate. Given h(·) and $\boldsymbol{D}_{\backslash k}$, the correct classification rate c_q with a given value of q is estimated as the proportion of times the predicted and observed brain state labels are equal:

$$\begin{aligned}\mathbb{P}(\{\text{calssify correctly}|h(\cdot),q\}) &= c_q \\ &= \frac{1}{K}\sum_{k=1}^{K} c_{q,k} \\ &= \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N'}\sum_{n=1}^{N'} \mathbb{I}(h(x_{\backslash k,n}) = \mathrm{t}_{\backslash k,n})\end{aligned} \tag{6.16}$$

where $c_{q,k}$ is the accuracy estimated in the kth fold of CV, and N' is the number of samples in the kth testing CV dataset.

We can not only obtain a classification model h(·) and also a set of selected features $\boldsymbol{V}_{q,k}$ in each fold of CV. For any subset of the whole feature set $\mathbf{V}$, we can estimate the probability of the features in $\dot{\mathbf{V}}$ are among those selected sets $\{\boldsymbol{V}_{q,k}\}_{k=1}^{K}$ as

$$\mathbb{P}\left(\{\text{select } \dot{\mathbf{V}}|h(\cdot),q\}\right) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(\dot{\mathbf{V}} \subseteq \boldsymbol{V}_{q,k}) \tag{6.17}$$

This represents the stability of a feature set $\dot{\mathbf{V}}$ being selected (Meinshausen & Bühlmann 2010).

Equation 6.16 can only provide the assessment of predictive accuracy but give no information in regards to the degree of agreement among selected features $\{\boldsymbol{V}_{q,k}\}_{k=1}^{K}$. Equation 6.17, on the other hand, can estimate the stability of a feature set in cross validation but does not indicate its predictive power. Therefore, we adopt the method in (Kirk et al. 2013) to estimate the accuracy and stability at the same time. Here, we choose the 6th strategy proposed in (Kirk et al. 2013) to approximate the joint probability of stability and predictive accuracy by:

$$\begin{aligned}\mathrm{P}_q(\dot{\mathbf{V}}) &= \frac{1}{q}\sum_{\mathrm{v}_i\in\dot{\mathbf{V}}}\mathbb{P}(\{\text{select } \mathrm{v}_i \text{ and calssify correctly}|h(\cdot),q\}) \\ &= \frac{1}{q}\sum_{\mathrm{v}_i\in\dot{\mathbf{V}}}\frac{1}{K}\sum_{k=1}^{K} c_{q,k}\mathbb{I}(\mathrm{v}_i \in \boldsymbol{V}_{q,k})\end{aligned} \tag{6.18}$$

where, $\dot{\mathbf{V}}$ is a feature set of size q. Therefore, to choose the optimal feature set $\widetilde{\boldsymbol{V}}_q$ with the largest joint probability (denoted as $\tilde{P}_q$), we can simply calculate $\mathbb{P}(\{\text{select } \mathrm{v}_i \text{ and calssify correctly}||h(\cdot),q\})$ for all $\mathrm{v}_i \in \boldsymbol{V}_{q,k}$. We then select the top q optimal features with the highest scores.

6.3.5 The whole process

The whole process of our pipeline is summarized in Algorithm 6.1.

Algorithm 6.1 The whole pipeline

Input Complete dataset $\mathrm{D} = \{\mathbf{x}, \mathbf{t}\}$; parameter R, K and Q

Output Optimal feature set $\widetilde{\boldsymbol{V}}_{\tilde{q}}$

$S \leftarrow$ the total number of features

$M \leftarrow$ the number of features in each subspace

$L \leftarrow$ the number of subspaces

// for different sizes of the selected feature set

for $q \leftarrow 1$ **to** Q

    // cross validation

    for $k \leftarrow 1$ **to** K

        Set the training dataset $\boldsymbol{D}_k = \{\mathbf{x}_k, \mathbf{t}_k\}$

        Set the testing dataset $\boldsymbol{D}_{\setminus k} = \{\mathbf{x}_{\setminus k}, \mathbf{t}_{\setminus k}\}$

        // repetitions of RS

        for $r \leftarrow 1$ **to** R

            //Randomly generate L subsets of M features

            for $l \leftarrow 1$ **to** L

                Randomly generate subspace $\mathbf{x}_k^l$, which contains a set of M features, $\overline{\mathbf{V}}^l$.

                $\overline{\overline{\boldsymbol{V}}}_l \leftarrow \mathrm{SBL}(\mathbf{x}_k^l, \overline{\mathbf{V}}^l)$ // features selected by SBL

            end

            for $i \leftarrow 1$ **to** S

                $\lambda_{i,r} \leftarrow \frac{\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\overline{\boldsymbol{V}}}_l)}{\sum_{l=1}^{L} \mathbb{I}(v_i \in \overline{\mathbf{V}}^l)}$

            end

        end

        $\lambda_i \leftarrow \frac{1}{R} \sum_{r=1}^{R} \lambda_{i,r}$

        Determine $\boldsymbol{V}_{q,k}$ that contains features with the top q largest λ_i

        //construct a classification model

        $\mathrm{h}(\cdot) \leftarrow \mathrm{Classifier}(\mathbf{x}_k, \boldsymbol{V}_{q,k})$ // return a prediction model

20	Obtain the prediction results of the testing dataset, $\mathrm{h}(\mathbf{x}_{\backslash k})$
21	$N' \leftarrow$ the number of samples in $\mathbf{x}_{\backslash k}$
22	$c_{q,k} \leftarrow \frac{1}{N'}\sum_{n=1}^{N'} \mathbb{I}(h(x_{\backslash k,n}) = \mathrm{t}_{\backslash k,n})$// predictive accuracy
23	**end**
24	**for** $i \leftarrow 1$ **to** S
25	$p_i \leftarrow \frac{1}{K}\sum_{k=1}^{K} c_{q,k}\mathbb{I}(\mathrm{v}_i \in \boldsymbol{V}_{q,k})$
26	**end**
27	Determine $\widetilde{\boldsymbol{V}}_q$ that contains features with the top q largest p_i
28	$\tilde{P}_q \leftarrow \frac{1}{q}\sum_{\mathrm{v}_i \in \widetilde{\boldsymbol{V}}_q} p_i$
29	**end**
30	Select the optimal value of $\tilde{q}$ with largest $\tilde{P}_{\tilde{q}}$
31	Record $\widetilde{\boldsymbol{V}}_{\tilde{q}}$

6.4 Experimental Results

We will investigate the performance of our method using both a synthetic dataset and a real fMRI dataset. The results of our method are compared with the results from stability and accuracy strategies. For the stability method, the process of selecting the optimal value of $\tilde{q}$ is similar as Algorithm 6.1, where the 24-28$^{\text{th}}$ lines are modified as:

24	**for** $i \leftarrow 1$ **to** S
25	$p_i \leftarrow \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}(\mathrm{v}_i \in \boldsymbol{V}_{q,k})$
26	**end**
27	Determine $\widetilde{\boldsymbol{V}}_q$ that contains features with the top q largest p_i
28	$\tilde{P}_q \leftarrow \frac{1}{q}\sum_{\mathrm{v}_i \in \widetilde{\boldsymbol{V}}_q} p_i$

For the accuracy method, we also determine $\widetilde{\boldsymbol{V}}_q$ by selecting the top q most frequent voxels across CV as above. The corresponding 28$^{\text{th}}$ line is replaced by:

24	$\tilde{P}_q \leftarrow \frac{1}{K}\sum_{k=1}^{K} c_{q,k}$

As the ground truth of the simulation dataset is known, we can check whether our method has selected real features that are used in simulation dataset generation.

6.4.1 Simulated data analysis

6.4.1.1 Data generation

The simulation dataset is generated in the same manner as (Varoquaux et al. 2012), where the key characteristic of real fMRI datasets is maintained: the measurement matrix $\mathbf{x}$ is locally correlated and the non-zero elements of $\mathbf{w}$ is spatially clustered. Specifically, we work on a 3D feature space with the dimension of $20 \times 20 \times 20$. Therefore, the total number of features is $S = 8000$. We set the number of samples to be 100. The measurement matrix $\mathbf{x} \in R^{100\times8000}$ is constructed by an i.i.d Gaussian random matrix that is spatially smoothed with a Gaussian filter of standard deviation of 0.8. A sample of $\mathbf{x}$ is shown in Figure 6-6, from which we can see that spatially neighbouring voxels have smoothed values.

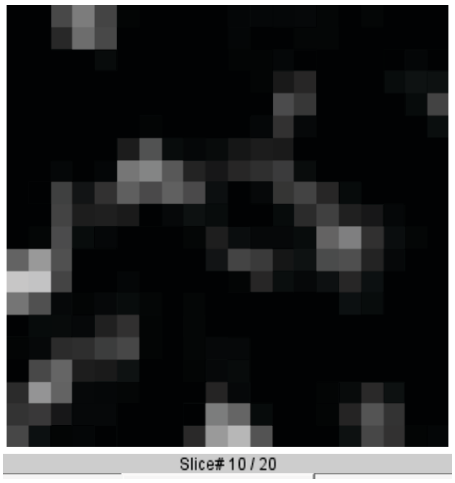


Figure 6-6 An example of $\mathbf{x}$ to show the measurement matrix is locally correlated. The 50th sample of measurement is selected for illustration. Then we convert this sample vector with the dimension of 1×8000 into a 3D matrix with the dimension of $20 \times 20 \times 20$. The 'imshow3D' function (http://uk.mathworks.com/matlabcentral/fileexchange/41334-imshow3d--3d-imshow--new-version-released--see--imshow3dfull-) is used to display this 3D matrix slice by slice in grayscale. In this figure, we show the 20th slice. The image is composed exclusively of shades of grey, varying from black at the lowest value to white at the largest.

To investigate the performance of our method with different sparsity of feature space, three different datasets are generated where the number of non-zero elements of $\mathbf{w}$ is set to be 16, 54 and 128 respectively. The brain state vector $\mathbf{t}$ is generated as follows: a response vector is first generated by the linear model using $\mathbf{w}$ and $\mathbf{x}$ with additive noises (SNR=0.9); then the

sigmoidal function transforms the response vector into sigmoid curve; threshold it with 0.5 to obtain $\mathrm{t}_n \in \{0,1\}$.

There are two parameters involved in the RS-SBL feature selection, which are M and L. In our experiments, we investigate the influence of these two parameters by selecting different values from $L \in \{10,40,70,100\}$ and $M \in \{\frac{1}{5}S, \frac{1}{3}S, \frac{7}{15}S, \frac{3}{5}S\}$. To save computational effort, the number of repetitions for random subspace generation is set as $R = 50$. We choose the value of K to be 10, and thus 10-fold cross validation is used in our experiment.

Features, whose corresponding elements of $\mathbf{w}$ are non-zero, are regarded as real positives. Ideally, the optimal feature set $\widetilde{\boldsymbol{V}}_{\tilde{q}}$ should be equal to the set of all real positives. We estimate the ability of our method in correctly selecting positive features by calculate false positive (FP) and false negative (FN) rates (explained in subsection 3.4.1).

6.4.1.2 Results

We first compare the results of our method with the results from accuracy and stability methods. Dataset 1, which contains a number of 16 non-zero elements in $\mathbf{w}$, is used for demonstration. All these three methods (i.e. stability, accuracy and our method) return optimal sets of features $\widetilde{\boldsymbol{V}}_q$ with associated scores $\tilde{P}_q$ for $q \in \{1, \dots, 30\}$. With a given value of q, the best feature set is achieved by finding the highest score, which is calculated according to 6.18 and equations shown in subsection 6.4 for different methods respectively.

The results are shown in Figure 6-7 and Figure 6-8, where 16 subplots are obtained from different pairs of L and M. From these two figures, we can observe that the scores of stability and our methods follow a quite similar trend. They both increase gradually to reach their maximum scores around $q = 16$ and then go down with $q > 16$. In contrast, the score of the accuracy method quickly reaches a steady state roughly around $q = 5$. As the true number of non-zero elements in $\mathbf{w}$ is 16, the accuracy method selects much less features. This is because SBL aims to return the most sparse feature space that can maintain the best predictive performance. As the fMRI voxels are spatially correlated, SBL will only select a subset of correlated features, which are fair enough to construct an accurate predictive model. However, in 10-fold CV, as the input training dataset is different in each fold and the feature subspaces are randomly generated, the selected feature set in each fold by the accuracy method may vary significantly. Each fold of CV may randomly choose features from the correlated ones, and the agreement in feature sets across will not be high.

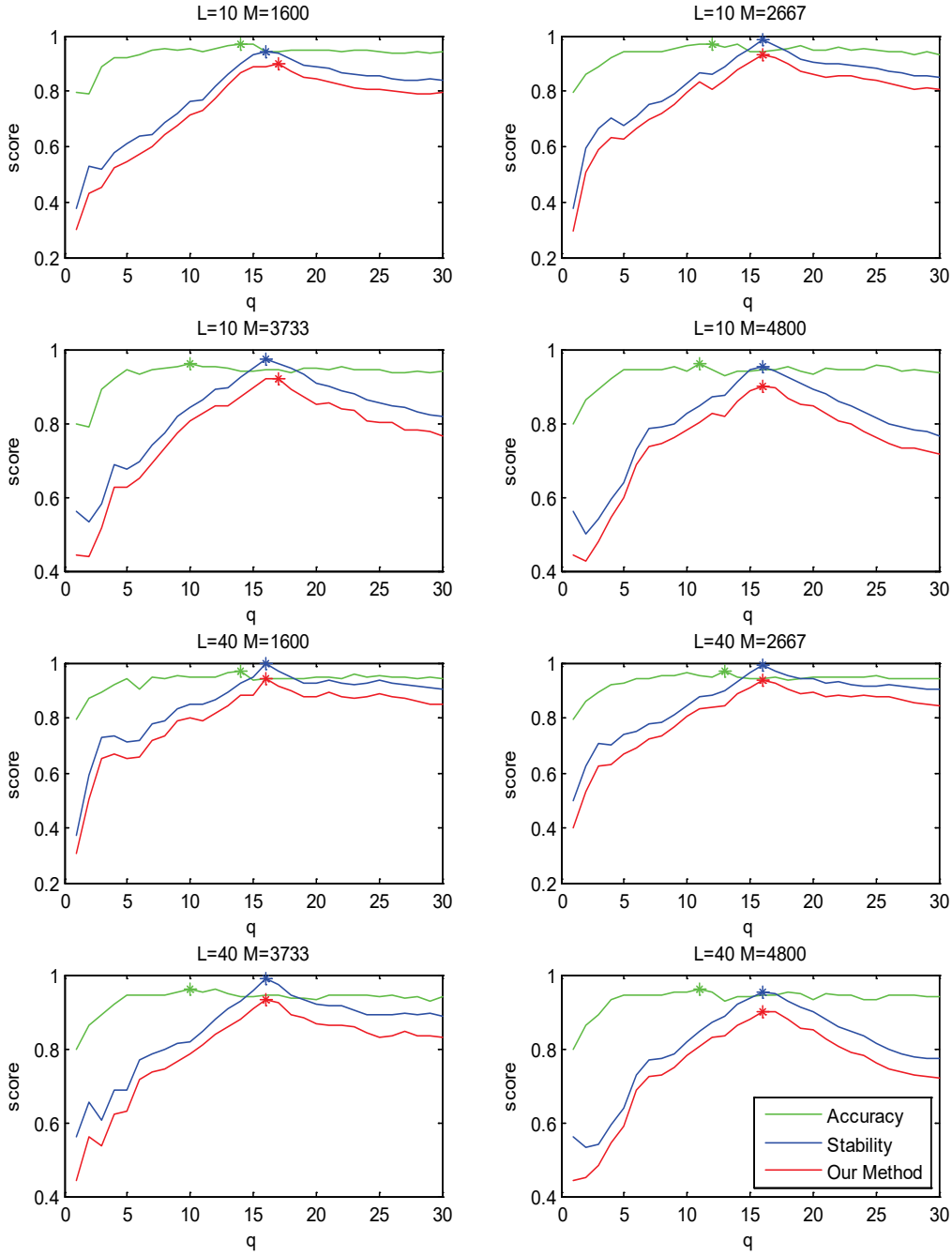


Figure 6-7 Values of $\tilde{P}_q$ with q varying from 1 to 30. 8 subplots are generated with different pairs of L and M. More results with different values of L and M are shown in Figure 6-8.

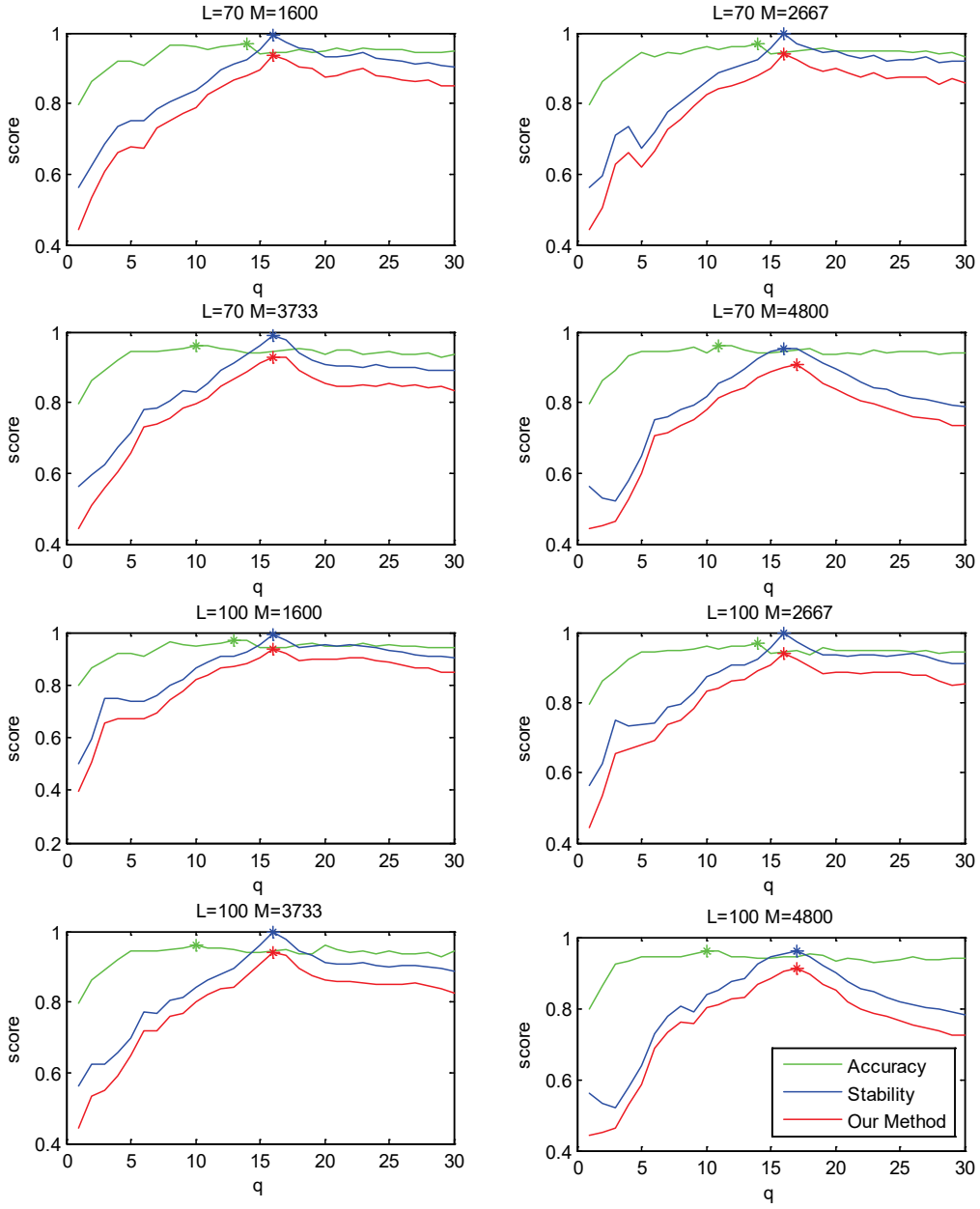


Figure 6-8 Values of $\tilde{P}_q$ with q varying from 1 to 30. 8 subplots are generated with different pairs of L and M. More results with different values of L and M are shown in Figure 6-7.

Table 6-1, Table 6-2 and Table 6-3 record the optimal results under different combinations of L and M obtained by the accuracy, stability and our methods respectively. The 'accuracy' columns in all tables are calculated following Equation 6.16. Thus, in Table 6-1 values in 'score' (the accuracy method calculates the score using Equation 6.16 as well) and 'accuracy' columns are identical.

From Table 6-1, we can observe that the maximal score of the accuracy method is obtained when $(L, M) \in \{(10,1600), (10,2667), (40,2667), (100,1600)\}$. The accuracy is achieved as high as 0.972 and FP rate is as low as zero. However, some real positives are not discovered by the accuracy method and FN is above zero. This is because the optimally determined values of q are lower than its real value (i.e. 16).

Table 6-2 shows the results from the stability method, where the maximal score is 0.996 obtained when $(L, M) \in \{(40,1600), (70,2667), (100,2667), (100,3733)\}$. Different from the accuracy method, the stability method can maintain both the FP and FN rates at zero. However, its accuracy is 0.944 which is lower than 0.972 from the accuracy method. Therefore, the stability method sacrifices the accuracy to increase FP and FN rates. The results of our method are shown in

Table 6-3. The maximal score of our method can be obtained with $(L, M) \in \{(100,3733)\}$, where the accuracy value is 0.944 and FP&FN are both zero. In contrast to the other methods, our method only returns one optimal pair of (L, M). It suggests that our method can give a unique solution of optimal settings.

Table 6-1 Results obtained by the accuracy method with different pairs of L and M. The optimal results with highest score are written in red

L	M	q	Associated score	Accuracy value	FP	FN
10	1600	14	0.972	0.972	0	0.125
10	2667	12	0.972	0.972	0	0.25
10	3733	10	0.963	0.963	0	0.375
10	4800	11	0.959	0.959	0	0.313
40	1600	14	0.969	0.969	0	0.125
40	2667	13	0.972	0.972	0	0.188
40	3733	10	0.963	0.963	0	0.375
40	4800	11	0.959	0.959	0	0.313
70	1600	14	0.969	0.969	0	0.125
70	2667	14	0.969	0.969	0	0.125
70	3733	10	0.963	0.963	0	0.375
70	4800	11	0.959	0.959	0	0.313

100	1600	13	0.972	0.972	0	0.188
100	2667	14	0.969	0.969	0	0.125
100	3733	10	0.963	0.963	0	0.375
100	4800	10	0.963	0.963	0	0.375

Table 6-2 Results obtained by the stability method with different pairs of L and M. The optimal results with highest score are written in red.

L	M	q	Associated score	Accuracy value	FP	FN
10	1600	16	0.941	0.944	0	0
10	2667	16	0.984	0.944	0	0
10	3733	16	0.973	0.944	0	0
10	4800	16	0.953	0.944	0	0
40	1600	16	0.996	0.944	0	0
40	2667	16	0.992	0.944	0	0
40	3733	16	0.988	0.944	0	0
40	4800	16	0.953	0.944	0	0
70	1600	16	0.992	0.944	0	0
70	2667	16	0.996	0.944	0	0
70	3733	16	0.988	0.944	0	0
70	4800	16	0.953	0.944	0	0
100	1600	16	0.992	0.944	0	0
100	2667	16	0.996	0.944	0	0
100	3733	16	0.996	0.944	0	0
100	4800	17	0.960	0.947	0.0625	0

Table 6-3 Results obtained by our method with different pairs of L and M. The optimal results with highest score are written in red.

L	M	q	Associated score	Accuracy value	FP	FN
10	1600	17	0.897	0.945	0.0625	0

10	2667	16	0.929	0.945	0	0
10	3733	17	0.921	0.947	0.0625	0
10	4800	16	0.901	0.945	0	0
40	1600	16	0.940	0.945	0	0
40	2667	16	0.936	0.945	0	0
40	3733	16	0.933	0.945	0	0
40	4800	16	0.901	0.945	0	0
70	1600	16	0.936	0.945	0	0
70	2667	16	0.940	0.945	0	0
70	3733	16	0.929	0.945	0	0
70	4800	17	0.908	0.947	0.0625	0
100	1600	16	0.936	0.945	0	0
100	2667	16	0.940	0.945	0	0
100	3733	16	0.941	0.944	0	0
100	4800	17	0.911	0.947	0.0625	0

Having compared the performance of our method with other methods using Dataset 1, the next step is to investigate our performance when the sparsity of $\mathbf{w}$ increases. Here we use Dataset 2 and Dataset 3, whose corresponding number of non-zero elements in $\mathbf{w}$ is 54 and 128. The value of q varies from 1 to 70 and 1 to 150 for Dataset 2 and Dataset 3 respectively. Table 6-4 summarizes the optimal results achieved by three different methods for Dataset 2 and Dataset 3. Consistent with the previous observation of Dataset 1, our method returns only one combination of (L, M).

For Dataset 2, the score of stability method peaks at multiple pairs of (L, M), indicating that the solution may not be stable. The optimal value of q found by this method is 2, which is far smaller than the real value (i.e. 54). Thus the FN rate is as high as 0.963. The stability method can reduce the FP rate to zero at the expense of discarding a large amount of real positives. The accuracy value is only 0.898. The accuracy method obtains 4 optimal settings of (L, M), with which the accuracy can reach to 0.984. The FP and FN range from 0.056 to 0.259 and 0.148 to 0.333 respectively. Our method, on the other hand, can achieve the optimal results with a single pair of (L, M). The optimal value of q is 47 and the accuracy is 0.97 that is slightly smaller than the accuracy achieved by the accuracy strategy. Encouragingly, the FP rate is zero and the FN is as low as 0.13. In summary, we can observe that: 1) the optimal value of q obtained by our method is between the values obtained by the stability and

accuracy methods; 2) the FP and FN values achieved by our method are best; 3) the accuracy of our method is maintained at a high level, slightly smaller than the value of the accuracy method. Similar observations can be found for Dataset 3.

Table 6-4 Optimal results from different methods for Dataset 2 and Dataset 3

		(L,M)	q	Accuracy value	FP	FN
Dataset 2	**Stability method**	All pairs except (10,1600)	2	0.898	0	0.963
	Accuracy method	(10,4800)	48	0.984	0.222	0.333
		(100,2667)	49	0.984	0.056	0.148
		(70,3733)	7	0.984	0.222	0.167
		(70,3733)	59	0.984	0.259	0.167
	Our method	(100,1600)	47	0.970	0	0.13
Dataset 3	**Stability method**	(100,1600)	115	0.947	0.110	0.211
	Accuracy method	(40,4800)	137	0.964	0.344	0.273
		(40,4800)	148	0.964	0.406	0.250
		(40,4800)	149	0.964	0.414	0.250
	Our method	(100,1600)	129	0.951	0.180	0.172

Figure 6-9 plots the scores of three different methods with $(L, M) = (100,1600)$. The values of L and M are set to be optimal for our method as shown in Table 6-4. Figure 6-9 (a) is obtained from Dataset 2, where q varies from 1 to 70. From this figure, we can observe that the score of the stability method quickly reaches to its peak at $q = 2$. This might be because Dataset 2, whose $\mathbf{w}$ and $\boldsymbol{x}$ are randomly generated while maintaining the spatial correlation of voxels, contains a lot of correlated voxels. SBL may only select one out of a group of correlated ones in each subspace. In consequence, the agreement among feature sets selected by multiple RS-SBL is not large. Thus, the stability score tends to reach a local maximum with small value of q when strong correlation of voxels exists. Our method, taking both stability and accuracy into account, can avoid this situation. The highest score of our method peaks are $q = 47$. Similarly, in Figure 6-9 (b) the stability score for Dataset 3 also has a local

maximum around $q = 20$. By smoothing the stability score with the accuracy score, our method does not have a significant peak as much as stability at small value of q.

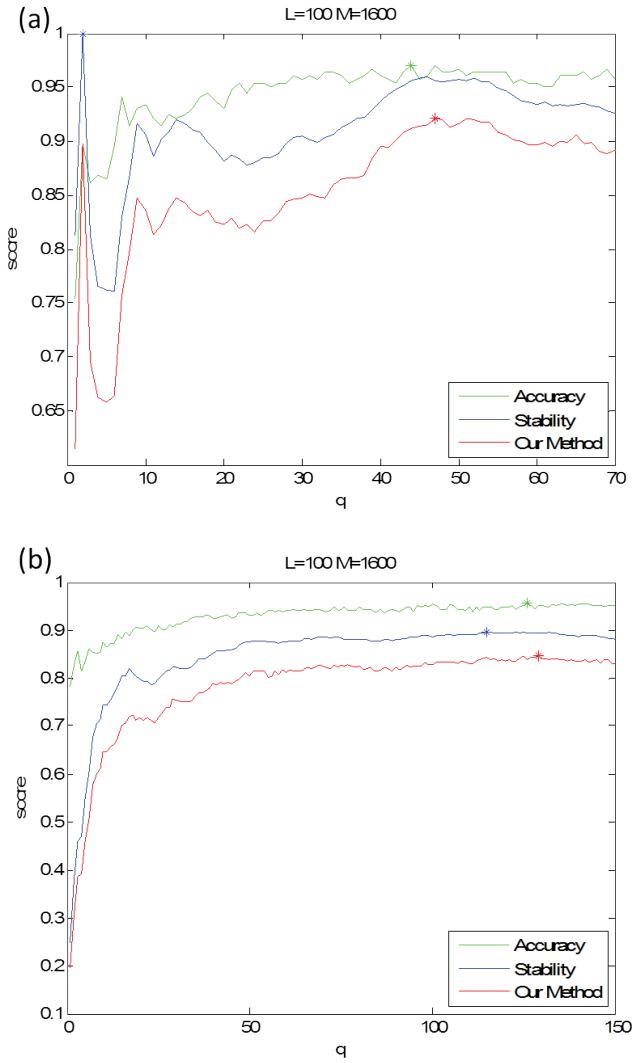


Figure 6-9 Plots of scores obtained from different methods with $(L, M) = (100,1600)$. a) results are for Dataset 2; b) results are for Dataset 3.

6.4.2 Real data analysis

6.4.2.1 Data collection

We use the fMRI datasets provided by Human Connectome Project (HCP) (Van Essen et al. 2012) to demonstrate the applicability of our method on fMRI real dataset. Seven different tasks are performed by HCP to generate multiple task-evoked fMRI datasets. In our analysis, we choose the dataset for relationship processing task. Subjects undertake two different kinds of processing: relation processing and matching processing. In the relation processing, subjects are given with a pair of objects and need to identify in which dimension these two

objects are different; then they need to justify whether another pair of objects also differ in the same dimension. In the matching processing, subjects need to find whether an object matched to either of the other two objects on a suggested dimension. The fMRI datasets are generated by a 3T WU-Minn HCP scanner with a modified 3T Siemens Skyra system. Each subject undertook two individual sessions, where each session contains 3 blocks for each processing task and rest state. The preprocessing steps of raw fMRI datasets have been completed by HCP and the fMRI images of all subjects are registered to the standard MNI125 space.

In our experiment, a predictive model is constructed to classify two processing tasks (relation and matching). Two unrelated subjects are selected for demonstration. For each subject, datasets from two sessions are concatenated together. Thus, the fMRI dataset for each subject contains in total of 310 samples and 228,000 voxels. The number of features is much larger than the number of samples as $228{,}000 \gg 310$. The number of repetitions for RS generation is set to 50, and 10-fold CV is used to validate results. Two parameters involved in RS-SBL are set as $L \in \{10{,}30{,}50\}$ and $M \in \{\frac{1}{10}S, \frac{13}{60}S, \frac{1}{3}S\}$, where $S = 228{,}000$. The value of q ranges from 1 to 100.

6.4.2.2 Results

Figure 6-10 and Figure 6-11 show the plots of scores from different settings of L and M for subject 1 and subject 2 respectively. We can first observe that the stability and our scores reach to their maximum much more quickly than the accuracy score. With larger values of M, the optimal values of q for these two methods are smaller. This is because, when M is large, the number of features selected in random subspaces are large. Then, it is more likely that each subspace samples features that are highly correlated. As SBL only select a small subset of correlated features, quite a few ones are repeated selected across repetitions.

Table 6-5 shows optimal results obtained from different methods for Subject 1 and Subject 2. The optimal values of q found by the stability method for these two subjects are as low as 3 and 1 respectively. Although, values in the 'Stability' column for the Stability method are larger than values for other methods, it is quite likely that some true positives are discarded. Moreover, the accuracy values obtained by the stability method are lower than values from other methods. The accuracy method, on the other hand, selects a larger feature set with the accuracy value as high as 0.998. As this method does not consider the stability of features

selected across cross validation, its values in the 'Stability' column are lower than values from other methods.

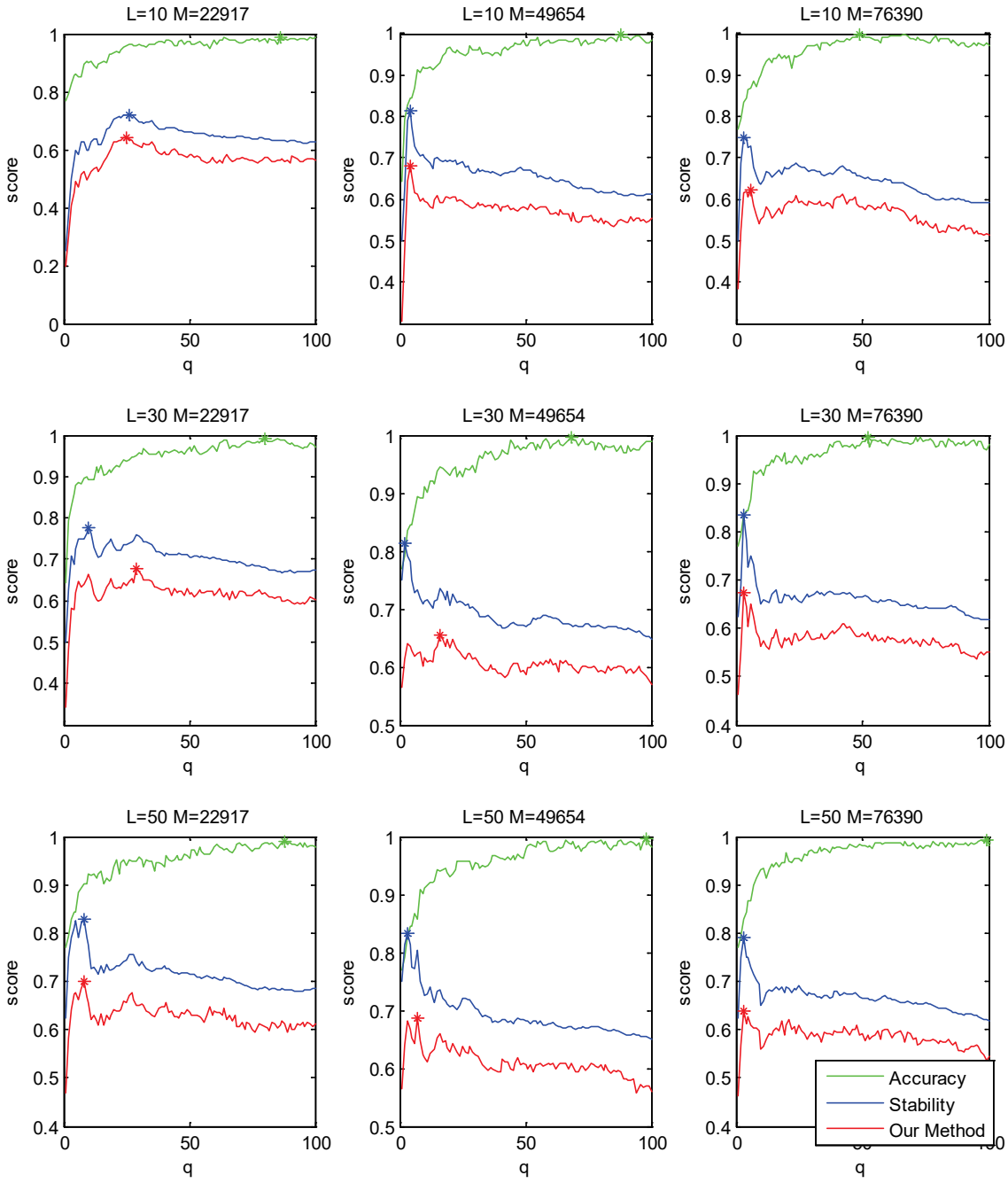


Figure 6-10 Values of $\tilde{P}_q$ with q varying from 1 to 100 for subject 1. 9 subplots are generated with different pairs of L and M.

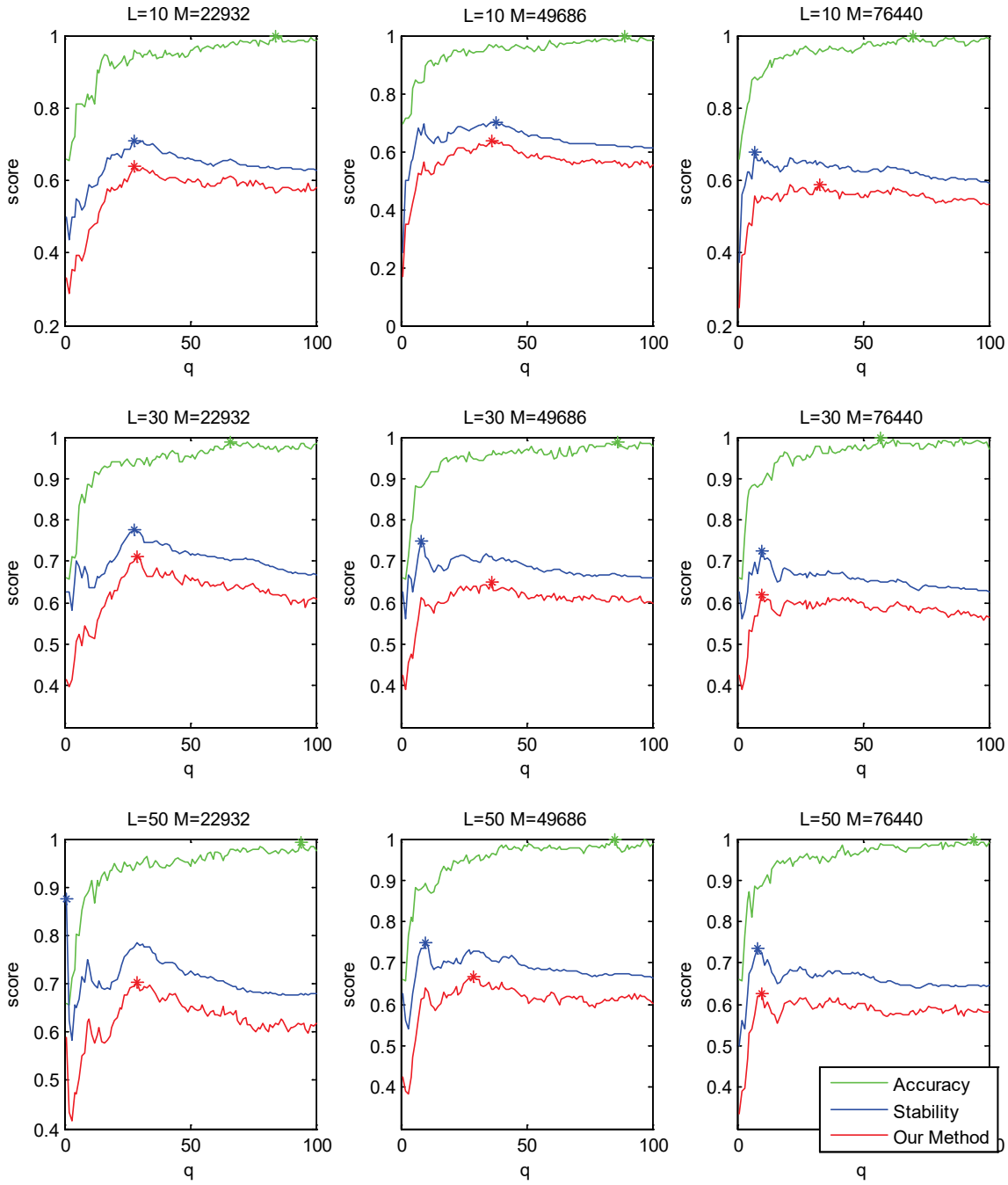


Figure 6-11 Values of $\tilde{P}_q$ with q varying from 1 to 100 for subject 2. 9 subplots are generated with different pairs of L and M.

Table 6-5 Optimal results from different methods for Subject 1 and Subject 2.

		(L,M)	q	Accuracy value	Stability value
Subject 1	**Stability method**	(50,49654)	3	0.828	0.833
		(30,76390)	3	0.838	0.833
	Accuracy method	(10,76390)	49	0.998	0.658
		(10,76390)	50	0.998	0.653
		(10,76390)	66	0.998	0.638
		(10,76390)	67	0.998	0.636
		(10,49654)	88	0.998	0.612
	Our method	(50,22917)	8	0.901	0.828
Subject 2	**Stability method**	(50,22932)	1	0.660	0.875
	Accuracy method	(30,76440)	57	0.998	0.651
		(10,76440)	70	0.998	0.621
		(10,22932)	84	0.998	0.631
		(50,49686)	85	0.998	0.674
		(30,76440)	89	0.998	0.633
		(50,76440)	94	0.998	0.645
		(50,49686)	97	0.998	0.668
	Our method	(50,22932)	29	0.962	0.772

Our method makes a balance between the stability and accuracy methods: 1) the optimal value of q found by our method is between the values detected by the stability and accuracy methods; 2) the accuracy achieved by our method is slightly lower than the value from the accuracy method but much larger than the value from the stability method; 3) the stability value obtained by our method is close to the values of the stability method and larger than the ones from the accuracy method. Consistent with the observations from the simulation analysis, our method returns a unique optimal settings of L and M. Moreover, we can observe that the optimal results can be obtained with the largest L and smallest M. We have explained that larger value of M influences the stability. With a large value of L, more random subspaces can be generated and it is more likely that the whole feature space is covered by RS. Therefore, it is preferable to choose large L and relatively small M.

6.5 Discussion

This chapter introduces a method to construct a principled predictive model while generating an optimal feature set. Our method can return a feature set with the accuracy value higher than the value obtained from the stability method and the stability value higher than the value from the accuracy method. We can see our method balances the accuracy and stability. Moreover, compared with the other two methods, our method tends to select a unique parameter setting. All of these improvements are simply achieved by a small modification on traditional methods, which ensures that computational complexity is not significantly increased.

The performance of our method is demonstrated using a set of simulation datasets and a real fMRI dataset. In the simulation analysis, Dataset 1 which contains 16 non-zero elements in $\boldsymbol{w}$ is used first. The stability and our methods can both work well with zero FN and FP rates. When we use Dataset 2 and Dataset 3, which contain more correlated features, the stability method cannot work well while our method can work best. To check the applicability of our method, we work on a real fMRI dataset provided by HCP. We can observe that the stability method cannot work well especially when the number features in each subspace is large. This is because when more features are selected, it is more likely that correlated real features are simultaneously selected in the same subspace. As SBL only selects one feature out of a correlated group, some correlated real features are not selected. Our method, combining the accuracy method, can avoid returning an extremely small feature set. However, to achieve a good performance, we suggest choosing a large value of L and small value of M. A large L can make the whole feature space be covered by random subspaces.

Cross validation is a common approach to construct classification model when no independent validation dataset is available. In some studies, when we have an independent dataset, we can directly use the whole training dataset to construct one model and evaluate its performance using the independent dataset. However, if we want to avoid overfitting to the data, still we can consider applying the cross validation approach to construct a model with optimal feature set from the whole training dataset and then apply the model on the independent dataset.

The selected feature set, which is stable across folds of cross validation, can be regarded as the predictive map of a brain pattern. This is benefited from applying linear models that the weighted sums of features are used for prediction. As no non-linear transformations of

features are required, we can directly detect which features are selected by checking their weights in the model. The predictive map can potentially tell us which functional brain regions participate in a specific brain activity. Therefore, it is quite helpful for brain pattern interpretation.

In conclusion, this chapter brings a biomarker discovery approach to MVPA in fMRI. Rather than investigating sophisticated machine learning methods for classification, our focus is generating a stable feature set that can be used to build an accurate predictive model. With the application of our method, the MVPA study can be improved. Our method is a good attempt to bring techniques in other field (Bioinformatics) into the fMRI research.

7 Deriving connectivity: an elastic approach

7.1 Motivation

In a complex living system, various components are interacting to complete a task. Thus a common study is to investigate component interactions by building the connectivity of networks. In bioinformatics, protein-protein interaction networks, for instance, have significant importance in explaining biological processes. By measuring the concentration of proteins at discrete time points, we can deduce their dependences. Based on some prior knowledge of protein features, such as structure and sequence, the inferred connectivity can be further improved to reflect real physical interactions of proteins. Similarly, gene regulatory network can be constructed using quantitative gene expression time series dataset. In the network, each node represents a gene and the link between nodes shows the dependence between genes. In neuroscience, functional connectivity can help to explore relationships among different function regions of a brain in completion of a task and further understand complex brain activities. The functional connectivity is constructed by detecting the temporal correlation between spatially remote neurophysiological events.

Existing connectivity methods fall into two categories: methods aiming at detecting statistical dependence between two nodes without considering the influence of other nodes, and methods targeting at reducing indirect links by considering the effect from other nodes. In the first category, the connectivity is inferred by computing pairwise correlation, mutual information or other similarity metric between each pair of nodes. The resulting connectivity contains a massive amount of indirect links owing to transitive effects of correlations. Figure 7-1 (from (Barzel & Barabási 2013)) shows examples that observed connectivity contains spurious links caused by indirect effects .

As methods in the first category may lead to inaccurate connectivity, there are many studies investigating direct effect inference. Various methods are developed to identify true pairwise interactions. For example, partial correlation is used to characterize conditional dependences (de la Fuente et al. 2004; Hemelrijk 1990; Veiga et al. 2007); graphic models are used to detect direct information flow of networks (Weigt et al. 2009; Jordan & Wainwright 2008); network deconvolution methods formulate the inference problem in the framework of graph-theory (Barzel & Barabási 2013; Feizi et al. 2013).

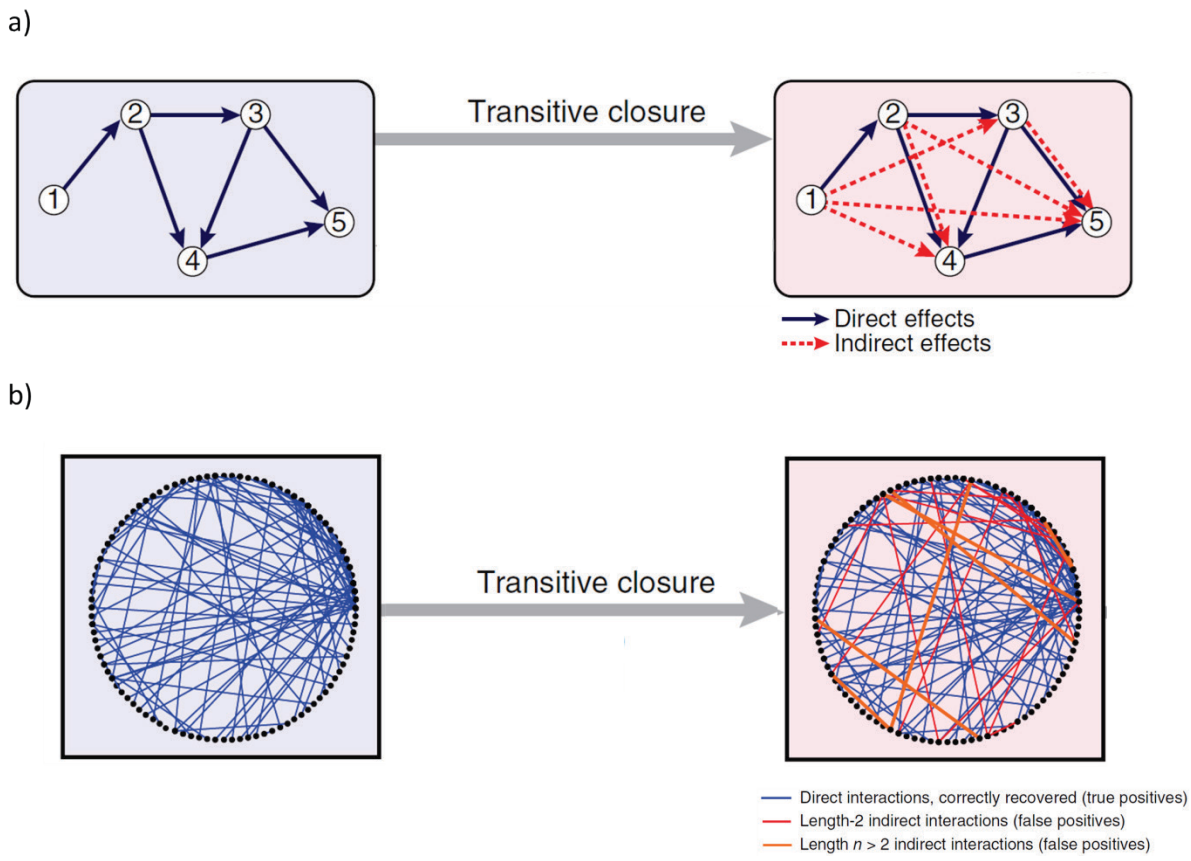


Figure 7-1 Examples of getting indirect links due to transitive effect. Subplots on the left are the true network, while those on the right are observed network with indirect effects. a) As results of transitive closure, direct edges in a network can lead to indirect links (dashed red arrows). b) A bigger network with more nodes. Direct links are represented by solid blue arrows, while indirect ones are shown by solid red and orange ones.

In this chapter, we focus on using the minimum value of partial correlation (denoted as minimum partial correlation) (Lauritzen 1996) as a metric to construct connectivity. The partial correlation between two variables is influenced by the controlling variable set. The minimum partial correlation (MPC) of a variable pair is the minimum of all absolute values of partial correlations obtained from all possible controlling sets. As we will discuss in the later part of this chapter, MPC is of significance in deducing true links of connectivity. However, calculating MPC is time-consuming. This is because the number of all possible controlling sets grows exponentially with the number of nodes. Several algorithms for causal inference can be adopted for fast computation of minimum partial correlation. Among them, the PC algorithm is widely used, which can lower the computational complexity by recursively removing spurious links from a complete graph (Pearl 2000).

In this chapter, we proposed a method, called the elastic PC algorithm, aiming to maximally approach to the minimum partial correlation within a given time constraint. We apply this method to generate functional connectivity using fMRI datasets. The proposed method can be potentially applied to other fields of datafied life research as well.

7.2 Background

Connectivity shows the dependence of components within a system. For example, gene regulatory network reflects interactions of regulators governing the gene expression levels; protein-protein interaction network shows interactions of proteins in a cellular system. This chapter focuses on generating connectivity using fMRI datasets. In fMRI analysis, functional connectivity studies the temporal dependences among brain regions (ROI), without resting on any generative models of distributed brain responses (Friston 2011). Although connectivity cannot give a comprehensive interpretation of brain functions, they can show important features of brain activities at the level of brain region interactions. For instance, they can inspire reasonable hypotheses of brain organizations (Buckner et al. 2013) and contribute to predictive model construction (Craddock et al. 2013), such as pathological state (Hawellek et al. 2011) and cognitive state (Shirer et al. 2012) classification. If the connectivity model can show the true and distinct characteristic of a specific brain activity, it can greatly help to generate an accurate prediction model. Therefore, it is always preferable to build the connectivity which is capable of reflecting true relationships of brain regions.

In this chapter, we only focus on investigating methods to infer the true skeleton of the connectivity. Thus, the direction (causality) and strength of connections are out of our scope. The skeleton of the connectivity can reflect system-level changes associated with different activities in the brain (Wang et al. 2010). There are many computational methods aiming to identify the existence of links between two ROIs (i.e. nodes). The study in (Smith et al. 2011) has systematically investigated the performance of most popular methods, where the best four methods as regularized inverse covariance (ICOV), fully partial correlation, Bayesian network and full correlation. Smith et al. (2011) introduces 28 simulated networks. Having designed 28 networks of various complexities, BOLD signals are generated according to the different structures of different networks. To mimic the characteristics of real fMRI datasets, dynamic causal models (Friston et al. 2003) are used to generate BOLD signals.

Among these four best methods, full correlation (i.e. Pearson correlation) is most commonly used to estimate the independences of two nodes. Many studies have demonstrated that the

full correlation method offers deep insights into hidden brain structures (Hermundstad et al. 2013) and powerful features for brain decoding (Turk-Browne 2013). Since it is a simple and parameter free approach, full correlation is usually the first choice to derive connectivity. However, we cannot ignore its potential limitations. The full correlation method has no means in reducing indirect links. For example, we get a simple 3 node network as $A \rightarrow B \rightarrow C$. As node A and node C are connected by node B, there is an information flow between these two nodes. Full correlation may therefore wrongly find a direct link between node A and node C. We regard this kind of link as indirect link.

Having noticed the problem of full correlation, some studies attempt to remove indirect links using fully partial correlation (Marrelec et al. 2006). The fully partial correlation method estimates the dependence of nodes by conditioning them on all other nodes. When the number of nodes in the network grows very large, it is intractable to directly calculate fully partial correlation. Lee et al. (2011) proposed a method based on the sparse linear modelling by introducing L1-norm regularization. To improve sparsity and sensitivity, Ryali et al. (2012) further developed a method based on both L1- and L2-norm regularisation. Although it reduces indirect links to some degree, it sometimes generates wrong connections.

ICOV tends to eliminate non-zero elements of the correlation matrix by introducing regularization (Friedman et al. 2008; Varoquaux et al. 2010). With an appropriate setting of regularization parameter, ideally ICOV can return the connectivity matrix reflecting true connections of nodes. However, it is usually not possible to define the optimal parameter beforehand, which therefore limits the applicability of this method. Although cross validation is a usual way to define the regularization parameter, we are not confident with its estimation. This is because a value of the parameter which can give good prediction of BOLD signals is not necessarily same as the one to accurately recover the connectivity.

In this chapter, we suggest minimum partial correlation as a measure of functional connectivity in fMRI. The minimum partial correlation between two nodes is the minimum of all partial correlations by controlling all possible subsets of other nodes. According to the theory of graphical causal models, there is no direct link between two nodes if the observations of the nodes are assumed to follow a multivariate Gaussian distribution and their minimum partial correlation is zero (Spirtes et al. 2000). Conversely, non-zero minimum partial correlation indicates a direct effect under the assumption of “stability” (Pearl 2000) or “faithfulness” (Spirtes et al. 2000). Although minimum partial correlation is a reasonable way

to detect node dependences, it is quite time-consuming. The number of all possible conditional subsets grows exponentially with the number of nodes. Thus, approximation methods are required. Several algorithms for causal inference can be adopted to calculate minimum partial correlation, such as SGS (Pearl 2000), PC (Pearl 2000), TPDA (Cheng et al. 2001), MMHC (Tsamardinos et al. 2006) and TPMB (Wang & Chan 2012). Among them, the PC method is most widely used.

The PC algorithm can lower the computational complexity of minimum partial correlation by recursively removing spurious links from a complete graph. One step in PC algorithm is based on the graph reduced by previous steps. The more edges are removed, the smaller the search space is. This algorithm enjoys polynomial time if the degree of the graph is finite (Spirtes et al. 2000), but it still requires exponential time in worst cases. Although (Iyer et al. 2013) demonstrates that the PC algorithm achieves a satisfied accuracy in simulation datasets, this recursive style suffers from one issue in practice. That is, the existence of an edge is determined by the significance level. If an edge is deleted incorrectly, this search space is improperly reduced, which affects the next steps of the PC algorithm. With a larger significant level, less computation time is required but it is more likely to wrongly remove real links. Thus, an appropriate choice of significant level is quite important for the PC algorithm.

In this chapter, we propose a new method, called the elastic PC algorithm, to approximate minimum partial correlation. Unlike the original PC algorithm, our method is not relied on one specific significant level. Within a given time constraint, our method approaches to the minimum partial correlation as possible. The following sections will introduce this algorithm in detail. A set of simulation datasets from (Smith et al. 2011) and resting-state fMRI datasets provided by the Human Connectome Project (Van Essen et al. 2013; Stephen M Smith et al. 2013) are used to demonstrate our methods. Its performance in terms of both accuracy and efficiency is evaluated. The derived connectivity for real fMRI dataset has good interpretability.

7.3 Method

7.3.1 Minimum partial correlation

Suppose there is a set of N ROIs that is denoted as $\boldsymbol{R} = \{r_1, r_2, \dots\dots, r_N\}$. Their time series of BOLD signals are represented by $\boldsymbol{X}_i = \{x_i^1, x_i^2, \dots\dots, x_i^T\}$ for $i \in \{1, \dots, N\}$. Given two ROIs

r_i and r_j, as well as a conditional set $\boldsymbol{Z} \subseteq \boldsymbol{R}\backslash\{r_i, r_j\}$, the partial correlation can be estimated from samples as follows:

$$\hat{\rho}_{i,j\cdot\boldsymbol{Z}} = \frac{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})(\varepsilon_{j\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})}{\sqrt{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})^2 \sum_{t=1}^{T}(\varepsilon_{j\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})^2}} \quad 7.1$$

$\{\varepsilon_{i\cdot\boldsymbol{Z}}^{t}\}$ is a set of the residuals of the linear regression with r_i as the response variable and the ROIs in $\boldsymbol{Z}$ as predictor variables. So is $\{\varepsilon_{j\cdot\boldsymbol{Z}}^{t}\}$.

The minimum partial correlation $\omega_{i,j}$ between two ROIs r_i and r_j calculated from samples is the minimum of all absolute values of partial correlations by controlling all possible subsets of other nodes, which is defined as:

$$\widehat{\omega}_{i,j} = \min\{|\hat{\rho}_{i,j\cdot\boldsymbol{Z}}|\} \, for \; all \; \boldsymbol{Z} \subseteq \boldsymbol{R}\backslash\{r_i, r_j\} \quad 7.2$$

MPC is of significance in deducing the connectivity. We can explain its importance by the theory of graphical causal models (or causal Bayes nets). Given a directed acyclic graph (DAG) $\mathcal{C} = (\mathbf{V}, \mathbf{E})$, which consists of a node set $\mathbf{V}$ and an edge set $\mathbf{E}$. Each node denotes a random variable and each directed edge represents a causal effect from one random variable to another one. A causal model represented by $\mathcal{C}$ generates a joint probability distribution over the nodes, which is denoted as $P(\boldsymbol{V})$. The DAG $\mathcal{C}$ and the distribution P satisfy the causal Markov condition if and only if for every node $r \in \boldsymbol{V}$, x is independent of $\boldsymbol{V}\backslash(Descendant(r) \cup Parents(r))$ given $Parents(r)$ (Spirtes et al. 2000).

When the Markov condition is met, the conditional independence relations in the generated distribution P are entailed in the DAG $\mathcal{C}$ using the concept of d-separation (Pearl 2000). A path is d-separated by a set $\boldsymbol{Z}$ if: a) the path contains a chain $r_1 \rightarrow r_2 \rightarrow r_3$ and r_2 is in $\boldsymbol{Z}$; b) the path contains a fork $r_1 \leftarrow r_2 \rightarrow r_3$ and r_2 is in $\boldsymbol{Z}$; c) the path contains a collider $r_1 \rightarrow r_2 \leftarrow r_3$ and r_2 along with its all descendants are not in $\boldsymbol{Z}$. Two nodes are d-separated by $\boldsymbol{Z}$ if and only if all paths between these two nodes are d-separated by $\boldsymbol{Z}$. If r_i and r_j are d-separated by a node set $\boldsymbol{Z}$, then r_i and r_j are conditionally independent in the distribution P given $\boldsymbol{Z}$. Here we only emphasis that there is no edge between nodes r_i and r_j if and only if they can be d-separated by some node set. That is, r_i and r_j are conditionally independent in the distribution P given some node set, if there is no edge in DAG $\mathcal{C}$ between these two nodes. The reverse statement is also true when the assumption of "stability" (Pearl 2000) or "faithfulness" (Spirtes et al. 2000) are satisfied. A distribution P is faithful to a DAG $\mathcal{C}$ if all and only the

conditional independence relations in P are entailed by $\mathcal{C}$. In other words, there is an edge between two nodes only if they are conditionally dependent in P given any node set.

According to (Lauritzen 1996), a controlling set $\boldsymbol{Z} \subseteq \boldsymbol{R}\backslash\{r_i, r_j\}$ gives $\hat{\rho}_{i,j\cdot\boldsymbol{Z}} = 0$ if and only if r_i and r_j are conditionally independent on $\boldsymbol{Z}$. Therefore, we can get the following proposition: if the distribution P of $\boldsymbol{V}$ in a DAG $\mathcal{C}$ is faithful to $\mathcal{C}$, there is an edge between two nodes r_i and r_j in $\mathcal{C}$ if and only if the minimum partial correlation $\omega_{i,j} \neq 0$. Thus, MPC is a measure of existence of connectivity. Assuming a brain network can be simplified as a graphical causal model which satisfied the Markov condition, there is a direct link between two nodes (i.e. ROIs) if and only if their minimum partial correlation is significantly non-zero.

Let $|\boldsymbol{Z}|$ denote the cardinality of the controlling node set $\boldsymbol{Z}$. From (Fisher 1924), we can know that the distribution of $\hat{\rho}_{i,j\cdot\boldsymbol{Z}}$ follows the same distribution of full correlation estimated from $T - |\boldsymbol{Z}|$ samples. Usually, the correlation is assumed to follow a Gaussian distribution after Fisher z-transformation (Fisher 1921; Fisher & Fisher 1915), which gives

$$\tilde{\rho}_{i,j\cdot\boldsymbol{Z}} = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{i,j\cdot\boldsymbol{Z}}}{1-\hat{\rho}_{i,j\cdot\boldsymbol{Z}}}\right)\sqrt{T - |\boldsymbol{Z}| - 3} \quad \sim N(0,1) \tag{7.3}$$

where $\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}$ is the z-score of estimated partial correlation $\hat{\rho}_{i,j\cdot\boldsymbol{Z}}$. If we want to reject the null hypothesis that partial correlation is zero with the significant level of α, then $\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}$ should be larger than $N^{-1}(1-\frac{\alpha}{2})$, where $N^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function. In this chapter, we approximate the absolute value of the z-score of the estimated minimum partial correlation as

$$\widetilde{\omega}_{i,j} = \min\left\{\left|\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}\right|\right\} for\ all\ \boldsymbol{Z} \subseteq \boldsymbol{R}\backslash\{r_i, r_j\} \tag{7.4}$$

Given $\boldsymbol{X}_i = \{x_i^1, x_i^2, \ldots\ldots, x_i^T\}$ for all $i \in \{1, \ldots, N\}$, we can construct a symmetric matrix $\boldsymbol{\Omega}$ containing $\widetilde{\omega}_{i,j}$ for all pairs of ROIs. The value of an element in $\boldsymbol{\Omega}$ indicates the existence of a link for the corresponding nodes.

7.3.2 PC algorithm

To calculate the minimum partial correlation, we need to enumerate all possible controlling sets, which is quite time consuming. Several algorithms for causal inference can be easily modified to calculate minimum partial correlation. Among them, the PC algorithm is widely used. The PC algorithm is composed of two main steps: skeleton graph inference and edge direction determination. Given a directed graph, its skeleton is an undirected graph by

stripping away all arrowheads from its links and getting rid of redundant undirected links. This chapter only focuses on generating functional connectivity without caring orientations of edges. Thus we concentrate on introducing the first step of the PC algorithm (Spirtes & Glymour 2000). There are two versions of PC: one is called the oracle version where the perfect conditional independence information is at hand; the other one is the sample version, which copes with a more realistic situation that conditional independence relationships have to be learned from data. In this chapter, we only discuss the sample version.

The PC algorithm uses an important feature of Bayesian network to achieve fast computation. That is, if two nodes r_i and r_j in an undirected graph can be d-separated, then there is a subset of their adjacent nodes d-separate them (Spirtes & Glymour 2000). Therefore, we do not need to enumerate all possible controlling sets. Instead we only need to consider subsets of adjacent sets. The PC algorithm is shown in Algorithm 7.1. It starts with a complete graph $\mathcal{C}$ and subsequently depletes edges from $\mathcal{C}$ in the loop on lines 4-20 in Algorithm 7.1. First, when ℓ is 0, all ordered pairs of (r_i, r_j) are checked for marginal dependence. After finding all marginal independent pairs and removing their corresponding edges in $\mathcal{C}$, the algorithm proceeds to the next iteration with $\ell = 1$. When $\ell = 1$, for any ordered pair (r_i, r_j) that is still adjacent in $\mathcal{C}$, check whether partial correlations on all possible node sets $\boldsymbol{Z}$ of size $\ell = 1$ are significant. If at least one partial correlation is not significant, then deplete $r_i - r_j$ from $\mathcal{C}$ and save the minimum partial correlation value to $\boldsymbol{\Omega}(\mathrm{i}, \mathrm{j})$ and $\boldsymbol{\Omega}(\mathrm{j}, \mathrm{i})$. After all pairs of adjacent nodes have been investigated for conditional independence given subsets of size ℓ of their adjacent sets, the algorithm goes to the next iteration with $\ell = 2$. This process continues until all adjacent sets in the current graph are smaller than ℓ. The resulted graph $\mathcal{C}$ shows the skeleton and $\boldsymbol{\Omega}$ contains the minimum partial correlation of node pairs.

Algorithm 7.1	The PC algorithm
Input	Signal $\boldsymbol{X}_i = \{x_i^1, x_i^2, \dots\dots, x_i^T\}$ of ROI r_i for all $i \in \{1, \dots, N\}$, significant threshold α
Output	Matrix $\boldsymbol{\Omega}$ containing z-scored MPC for all pairs of ROIs
1	Form the complete undirected graph $\mathcal{C}$ on the vertex set $\boldsymbol{R} = \{r_1, r_2, \dots\dots, r_N\}$
2	$\ell \leftarrow -1$
3	Initialize $\boldsymbol{\Omega}$ with a $N \times N$ matrix, where is entry is one

4 **do**

5 $\ell \leftarrow \ell + 1$

// the step ℓ uses the results from the step $\ell - 1$

6 **for each** ordered pair of (r_i, r_j) that are adjacent in $\mathcal{C}$ and satisfy $\left|adj(\mathcal{C}, r_i)\backslash\{r_j\}\right| \geq \ell$

// if the number of adjacent nodes of r_i excluding r_j is larger than ℓ

7 **for each** controlling set $\boldsymbol{Z} \subseteq adj(\mathcal{C}, r_i)\backslash\{r_j\}$ with $|\boldsymbol{Z}| = \ell$

8 $\hat{\rho}_{i,j\cdot\boldsymbol{Z}} \leftarrow \frac{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})(\varepsilon_{j\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})}{\sqrt{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})^2 \sum_{t=1}^{T}(\varepsilon_{j\cdot\boldsymbol{Z}}^{t} - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})^2}}$, where $\{\varepsilon_{i\cdot\boldsymbol{Z}}^{t}\}$ is a set of the residuals of the linear regression with r_i as the response variable and the ROIs in $\boldsymbol{Z}$ as predictor variables.

9 $\tilde{\rho}_{i,j\cdot\boldsymbol{Z}} \leftarrow \frac{1}{2}\log(\frac{1+\hat{\rho}_{i,j\cdot\boldsymbol{Z}}}{1-\hat{\rho}_{i,j\cdot\boldsymbol{Z}}})\sqrt{T - |\boldsymbol{Z}| - 3}$

10 **end**

11 $\tilde{\omega}_{i,j} \leftarrow \min\{\left|\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}\right|\}$

12 **if** $\tilde{\omega}_{i,j} \leq N^{-1}(1 - \frac{\alpha}{2})$

13 Delete edge $r_i - r_j$ from $\mathcal{C}$

14 **end**

15 **if** $\boldsymbol{\Omega}(\mathrm{i}, \mathrm{j}) > \tilde{\omega}_{i,j}$ // matrix $\boldsymbol{\Omega}$ is symmetric

16 $\boldsymbol{\Omega}(\mathrm{i}, \mathrm{j}) \leftarrow \tilde{\omega}_{i,j}$

17 $\boldsymbol{\Omega}(\mathrm{j}, \mathrm{i}) \leftarrow \tilde{\omega}_{i,j}$

18 **end**

19 **end**

20 **while** exist at least one pair of adjacent nodes (r_i, r_j) in $\mathcal{C}$ satisfying $\left|adj(\mathcal{C}, r_i)\backslash\{r_j\}\right| > \ell$

A modification of the PC algorithm has been proposed in (Colombo & Maathuis 2012), which is called PC-stable. Algorithm 7.2 shows the process of the PC-stable algorithm. The differences between Algorithm 7.2 and Algorithm 7.1 are written in red. Specifically, on lines 6-8 of Algorithm 7.2, the adjacent sets of all nodes are computed and stored at the beginning of each iteration. Therefore, the removal of edges on line 16 does not affect the calculation of partial correlation of other nodes at this level of ℓ. Algorithm 7.2 only records which edges

should be removed and for the purpose of finding adjacent sets it removes edges only when it goes to the next value of ℓ. These changes make the PC algorithm order-independent.

Algorithm 7.2	The PC-stable algorithm
Input	Signal $\boldsymbol{X}_i = \{x_i^1, x_i^2, \dots\dots, x_i^T\}$ of ROI r_i for all $i \in \{1, \dots, N\}$, significant threshold α
Output	Matrix $\boldsymbol{\Omega}$ containing z-scored MPC for all pairs of ROIs
1	Form the complete undirected graph $\mathcal{C}$ on the vertex set $\boldsymbol{R} = \{r_1, r_2, \dots\dots, r_N\}$
2	$\ell \leftarrow -1$
3	Initialize $\boldsymbol{\Omega}$ with a $N \times N$ matrix, where is entry is one
4	**do**
5	$\ell \leftarrow \ell + 1$
	// the step ℓ uses the results from the step $\ell - 1$
6	**for each** node r_i in $\mathcal{C}$
7	$a(r_i) \leftarrow adj(\mathcal{C}, r_i)$
8	**end**
9	**for each** ordered pair of (r_i, r_j) that are adjacent in $\mathcal{C}$ and satisfy $\left\|a(r_i)\backslash\{r_j\}\right\| \geq \ell$
	// if the number of adjacent nodes of r_i excluding r_j is larger than ℓ
10	**for each** controlling set $\boldsymbol{Z} \subseteq a(r_i)\backslash\{r_j\}$ with $\|\boldsymbol{Z}\| = \ell$
11	$\hat{\rho}_{i,j\cdot\boldsymbol{Z}} \leftarrow \frac{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^t - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})(\varepsilon_{j\cdot\boldsymbol{Z}}^t - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})}{\sqrt{\sum_{t=1}^{T}(\varepsilon_{i\cdot\boldsymbol{Z}}^t - \bar{\varepsilon}_{i\cdot\boldsymbol{Z}})^2 \sum_{t=1}^{T}(\varepsilon_{j\cdot\boldsymbol{Z}}^t - \bar{\varepsilon}_{j\cdot\boldsymbol{Z}})^2}}$, where $\{\varepsilon_{i\cdot\boldsymbol{Z}}^t\}$ is a set of the residuals of the linear regression with r_i as the response variable and the ROIs in $\boldsymbol{Z}$ as predictor variables.
12	$\tilde{\rho}_{i,j\cdot\boldsymbol{Z}} \leftarrow \frac{1}{2}\log(\frac{1+\hat{\rho}_{i,j\cdot\boldsymbol{Z}}}{1-\hat{\rho}_{i,j\cdot\boldsymbol{Z}}})\sqrt{T - \|\boldsymbol{Z}\| - 3}$
13	**end**
14	$\tilde{\omega}_{i,j} \leftarrow \min\{\left\|\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}\right\|\}$
15	**if** $\tilde{\omega}_{i,j} \leq N^{-1}(1 - \frac{\alpha}{2})$
16	Delete edge $r_i - r_j$ from $\mathcal{C}$
17	**end**

18 **if** $\boldsymbol{\Omega}(\mathrm{i},\mathrm{j}) > \widetilde{\omega}_{i,j}$ // matrix $\boldsymbol{\Omega}$ is symmetric

19 $\boldsymbol{\Omega}(\mathrm{i},\mathrm{j}) \leftarrow \widetilde{\omega}_{i,j}$

20 $\boldsymbol{\Omega}(\mathrm{j},\mathrm{i}) \leftarrow \widetilde{\omega}_{i,j}$

21 **end**

22 **end**

23 **while** exist at least one pair of adjacent nodes (r_i, r_j) in $\mathcal{C}$ satisfying $\left|a(r_i)\backslash\{r_j\}\right| > \ell$

7.3.3 Our method: elastic PC algorithm (ePC)

Our method is derived from the PC-stable algorithm. As we have discussed, the PC algorithm proceeds in a way that the skeleton of $\mathcal{C}$ at ℓ is dependent on the skeleton graph at $\ell - 1$. Therefore, if the previous step wrongly discards some edges, then these mistakes will propagate to the following steps. The edge depletion is highly relied on the choice of the significant level α. With a smaller significant level of α, more edges $r_i - r_j$ can satisfy $\widetilde{\omega}_{i,j} \leq N^{-1}(1 - \frac{\alpha}{2})$. Thus more edges are depleted with smaller α, increasing the chance that some edges are wrongly discarded. To increase the accuracy of edge removal at each ℓ, a larger α is required. However, the increase of α will increase the calculation time as more edges are remained at each ℓ. Therefore, α controls the accuracy and complexity at the same time.

It is quite difficult to optimally determine α. A common choice of α is 0.05. We can first set α to 0.05 and gradually increase its value to obtain more results with higher accuracy. This can avoid the situation that the PC algorithm runs for an unexpected long time with a large choice of α. For example, we would like to choose α from the set $\{0.05, 0.1, 0.15, 0.2, 0.25\}$. We first carry out the PC method with $\alpha = 0.05$ and then with larger values. If the algorithm keeps running for a long while with $\alpha = 0.2$, then we can just record the results with $\alpha = 0.15$ which is the most accurate results that we can obtain with a plausible execution time.

The above approach brings in extra computation load, as the results with small values of α are not used in the calculation with higher levels. To save computation effort, we propose our own method, the elastic PC algorithm, taking the concept of elasticity in cloud computing (Han et al. 2013). Our method automatically increases the significant threshold within a given time limit to maximally approach the minimum partial correlation. It implicitly avoids

repeatedly calculating same partial correlation that has been done in previous steps with lower significant threshold. The elastic PC algorithm is shown in Algorithm 7.3. In the elastic PC algorithm, we introduce a matrix $\mathbb{S}$ that contains the minimum partial correlation values with different ℓ. As there are N nodes in the graph $\mathcal{C}$ and ℓ should be smaller than $\left|adj(\mathcal{C}, r_i)\backslash\{r_j\}\right|$ for any pair of (r_i, r_j), the maximum value of ℓ is $N-2$. Thus, ℓ can vary from 0 to $N-2$. The dimension of $\mathbb{S}$ is $N \times N \times (N-1)$, where the kth slice $\mathbb{S}(:,:,k)$ contains minimum partial correlation calculated with $\ell = k-1$. We call this matrix as s-cube for short.

Algorithm 7.3	The elastic PC algorithm
Input	Signal $\boldsymbol{X}_i = \{x_i^1, x_i^2, \dots\dots, x_i^T\}$ of ROI r_i for all $i \in \{1, \dots, N\}$, significant threshold α, previous significant threshold β, previous s-cube $\mathbb{S}_\beta$
Output	s-cube $\mathbb{S}_\alpha$ for α
1	**for each** unordered pair of (r_i, r_j)
2	$\mathbb{S}_\alpha(i,j,:) \leftarrow \left\|\tilde{\rho}_{i,j\cdot\emptyset}\right\|$
3	$\mathbb{S}_\alpha(j,i,:) \leftarrow \left\|\tilde{\rho}_{i,j\cdot\emptyset}\right\|$
4	**end**
5	**for** $k = 2{:}\,(N-1)$
	// get the skeleton of $k-1$ using $\mathbb{S}_\alpha(:,:,k-1)$ and $\mathbb{S}_\beta(:,:,k-1)$
6	Initialize the reference skeleton $\mathcal{C}_\alpha$and $\mathcal{C}_\beta$ as the complete undirected graph on the vertex set $\boldsymbol{R} = \{r_1, r_2, \dots\dots, r_N\}$
7	**for each** unordered pair of (r_i, r_j)
8	**if** $\mathbb{S}_\alpha(i,j,k-1) < N^{-1}(1-\frac{\alpha}{2})$
9	Delete edge $r_i - r_j$ from $\mathcal{C}_\alpha$
10	**end**
11	**if** $\mathbb{S}_\beta(i,j,k-1) < N^{-1}(1-\frac{\beta}{2})$
12	Delete edge $r_i - r_j$ from $\mathcal{C}_\beta$
13	**end**
14	**end**
15	$\mathbb{S}_\alpha(:,:,k) \leftarrow \mathbb{S}_\beta(:,:,k)$ // initialize the results of $\mathbb{S}_\alpha$
16	$\ell \leftarrow (k-1)$

for each ordered pair of (r_i, r_j) that are adjacent in $\mathcal{C}_\alpha$ and satisfy $\left|adj(\mathcal{C}_\alpha, r_i)\backslash\{r_j\}\right| \geq \ell$

  for each controlling set $\boldsymbol{Z} \subseteq adj(\mathcal{C}_\alpha, r_i)\backslash\{r_j\}$ with $|\boldsymbol{Z}| = \ell$

    if $\boldsymbol{Z} \cup \{r_j\} \notin adj(\mathcal{C}_\beta, r_i)$

      Calculate $\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}$

    end

  end

  $\widetilde{\omega}_{i,j} \leftarrow \min\{|\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}|\}$

  if $\mathbb{S}_\alpha(i,j,k) > \widetilde{\omega}_{i,j}$

    $\mathbb{S}_\alpha(i,j,k\colon(N-1)) \leftarrow \widetilde{\omega}_{i,j}$

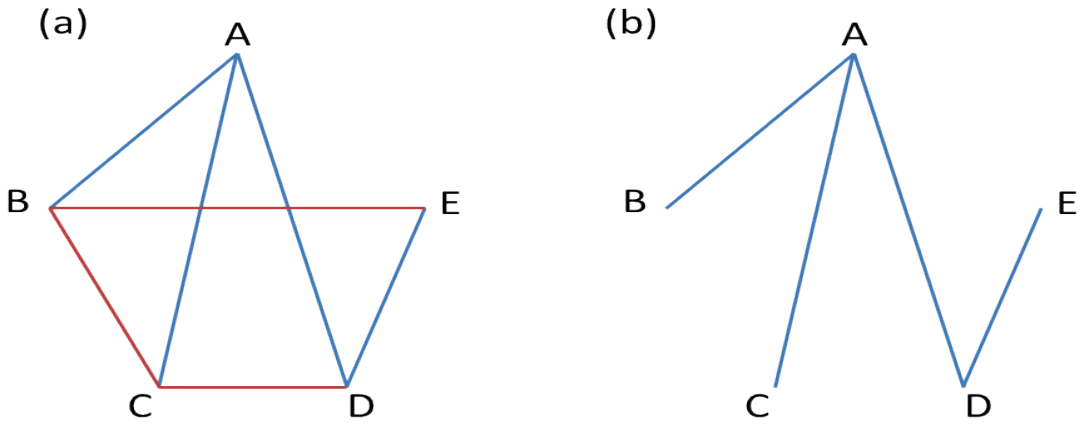
    $\mathbb{S}_\alpha(j,i,k\colon(N-1)) \leftarrow \widetilde{\omega}_{i,j}$

  end

end

end

The inputs of the elastic PC algorithm include the s-cube $\mathbb{S}_\beta$ from the previous significant threshold β. At beginning, marginal independence between any pair of (r_i, r_j) is calculated and initializes $\mathbb{S}_\alpha$. Then in the kth iteration, we first derive the graph $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$ using the $(k-1)$th slice of $\mathbb{S}_\alpha$ and $\mathbb{S}_\beta$. Afterwards, we move on to the calculation of the kth slice of $\mathbb{S}_\alpha$. For an ordered pair of (r_i, r_j) that is adjacent in $\mathcal{C}_\alpha$ but not adjacent in $\mathcal{C}_\beta$, the minimum partial correlation should be calculated again as it is not included in $\mathbb{S}_\beta$. Moreover, for an ordered pair of (r_i, r_j) that is adjacent in both $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, some steps of minimum partial correlation calculation have been done in $\mathbb{S}_\beta$ and we can directly reuse their results. That is, if a subset of the adjacent nodes $\boldsymbol{Z}$ for (r_i, r_j) in $\mathcal{C}_\alpha$ is also a subset of the adjacent nodes for (r_i, r_j) in $\mathcal{C}_\beta$, then we do not need to calculate $\tilde{\rho}_{i,j\cdot\boldsymbol{Z}}$. In this way, we can save computational effort by avoiding repeated calculations. The final results of the confidence matrix $\widehat{\boldsymbol{\Omega}}$ is $\mathbb{S}_\alpha(:,:,(N-1))$. An example of the elastic PC algorithm is illustrated in Figure 7-2.



(c)

#	Partial correlation	Need calculation?	#	Partial correlation	Need calculation?
1	$\tilde{\rho}_{A,B\cdot\{C\}}$	No	**11**	$\tilde{\rho}_{B,C\cdot\{E\}}$	Yes
2	$\tilde{\rho}_{A,B\cdot\{D\}}$	No	**12**	$\tilde{\rho}_{B,E\cdot\{A\}}$	Yes
3	$\tilde{\rho}_{A,B\cdot\{E\}}$	Yes	**13**	$\tilde{\rho}_{B,E\cdot\{C\}}$	Yes
4	$\tilde{\rho}_{A,C\cdot\{B\}}$	No	**14**	$\tilde{\rho}_{B,E\cdot\{D\}}$	Yes
5	$\tilde{\rho}_{A,C\cdot\{D\}}$	No	**15**	$\tilde{\rho}_{C,D\cdot\{A\}}$	Yes
6	$\tilde{\rho}_{A,D\cdot\{B\}}$	No	**16**	$\tilde{\rho}_{C,D\cdot\{B\}}$	Yes
7	$\tilde{\rho}_{A,D\cdot\{C\}}$	No	**17**	$\tilde{\rho}_{C,D\cdot\{E\}}$	Yes
8	$\tilde{\rho}_{A,D\cdot\{E\}}$	No	**18**	$\tilde{\rho}_{D,E\cdot\{A\}}$	No
9	$\tilde{\rho}_{B,C\cdot\{A\}}$	Yes	**19**	$\tilde{\rho}_{D,E\cdot\{B\}}$	Yes
10	$\tilde{\rho}_{B,C\cdot\{D\}}$	Yes	**20**	$\tilde{\rho}_{D,E\cdot\{C\}}$	Yes

Figure 7-2 An example of the elastic PC algorithm. Subplot (a) and (b) display skeletons of $\mathcal{C}_\alpha$and $\mathcal{C}_\beta$ at $\ell = 0$, satisfying $\alpha > \beta$. The table in (c) shows calculations required at $\ell = 1$ with the significant threshold of α. The column 'need calculation?' tells us whether the corresponding partial correlation needs to be calculated or can be directly obtained from $\mathbb{S}_\beta$. For node pairs (B, C), (C, D) and (B, E) (whose edges are in red in subplot(a)), all their partial correlations need to be calculated as $\mathcal{C}_\beta$ in subplot (b) does not contain the edges between them. For the node pair (A, B), partial correlation on set $\{E\}$ needs to be calculated. This is because in $\mathcal{C}_\beta$, $\{E\}$ is not a subset of adjacent nodes for (A, B) (same for the node pair (D, E)).

7.4 Experimental results

In this section, we use a synthetic dataset from (Smith et al. 2011) and also the resting-state fMRI datasets provided by the Human Connectome Project (Van Essen et al. 2013; Stephen M Smith et al. 2013) to demonstrate our method. The performance of the elastic PC-algorithm is compared with full correlation, fully partial correlation, regularized inverse covariance (ICOV) (Friedman et al. 2008), network deconvolution (ND) (Feizi et al. 2013) and global silencing (GS) (Barzel & Barabási 2013). A brief description of these methods is as follows:

- Full correlation: The full correlation of two ROIs r_i and r_j can be calculated using Equation 7.1 by setting $\boldsymbol{Z} = \emptyset$.
- Fully partial correlation: The fully partial correlation of two ROIs r_i and r_j can be calculated using Equation 7.1 by setting $\boldsymbol{Z} = \boldsymbol{R}\backslash\{r_i, r_j\}$, where $\boldsymbol{R} = \{r_1, r_2, \dots\dots, r_N\}$ is the whole set of ROIs.
- ICOV: Let $\boldsymbol{\Sigma}_{\text{sample}}$ be the sample covariance matrix, the ICOV method finds an optimal matrix $\boldsymbol{\Omega}$ whose elements reflect the independence of nodes:

$$\boldsymbol{\Omega} = \mathbf{argmin}_{\Omega>0}\text{tr}\left(\boldsymbol{\Omega}\boldsymbol{\Sigma}_{\text{sample}}\right) - \log\det\boldsymbol{\Omega} + \lambda\|\boldsymbol{\Omega}\|_{\ell_1} \tag{7.5}$$

where $\|\cdot\|_{\ell_1}$ is the element-wise ℓ_1 norm of coefficients in the matrix $\boldsymbol{\Omega}$. Optimization methods can be used to solve this equation. ICOV is a parameter dependent methods and the value of the regularization parameter λ greatly influences the results. Cross-validation is commonly used for parameter estimation.

- ND: Let $\boldsymbol{\Sigma}_{\text{sample}}$ be the sample covariance matrix. In ND, $\boldsymbol{\Sigma}_{\text{sample}}$ is modelled as the sum of direct dependence $\boldsymbol{\Omega}$ and indirect dependence due to indirect paths of increasing length in $\boldsymbol{\Omega}^2$, $\boldsymbol{\Omega}^3$ and others. $\boldsymbol{\Omega}$ is derived from:

$$\boldsymbol{\Omega} = \frac{\boldsymbol{\Sigma}_{\text{sample}}}{(\boldsymbol{I} + \boldsymbol{\Sigma}_{\text{sample}})} \tag{7.6}$$

where $\boldsymbol{I}$ is the identity matrix.

- GS: In the GS method, the relationship between $\boldsymbol{\Omega}$ and $\boldsymbol{\Sigma}_{\text{sample}}$ is modelled as

$$\boldsymbol{\Omega} = \frac{\boldsymbol{\Sigma}_{\text{sample}} - \boldsymbol{I} + \varphi((\boldsymbol{\Sigma}_{\text{sample}} - \boldsymbol{I})\boldsymbol{\Sigma}_{\text{sample}})}{\boldsymbol{\Sigma}_{\text{sample}}} \tag{7.7}$$

where $\varphi(\cdot)$ sets the off-diagonal terms of matrix to zero.

7.4.1 Simulation dataset analysis

For the elastic PC-algorithm, the initial significant threshold is 0.05 and the step size is 0.05. Ten steps are executed. For ICOV, the regularisation parameter is set to 5 and 100 (Smith et al. 2011). As we know the true skeleton of the simulation dataset, the performance of each method is represented by c-sensitivity (Smith et al. 2011). Suppose that each element of the symmetric matrix $\widehat{\boldsymbol{\Omega}}$ represents the estimated "confidence" about whether there is a link between the corresponding nodes. All elements of matrix $\widehat{\boldsymbol{\Omega}}$ are assumed to be nonnegative. The distribution of true positive (TP) is from the elements in the matrix $\widehat{\boldsymbol{\Omega}}$ whose corresponding links exist in the skeleton. Similarly, the distribution of false positive (FP) is from the elements in the matrix $\widehat{\boldsymbol{\Omega}}$ whose corresponding links do not exist in the skeleton. The c-sensitivity is defined as the fraction of true positives that have higher value in the matrix $\widehat{\boldsymbol{\Omega}}$ than the 95^{th} percentile of the false positive distribution. It shows the ability of the matrix $\widehat{\boldsymbol{\Omega}}$ for differentiating true links and spurious links. If there is no overlap between the distributions of TP and FP, the c-sensitivity is 100%. It is worth noting that diagonal elements are not used and only upper triangular elements are used in the calculation.

7.4.1.1 Data acquisition

Smith et al. (2011) generates rich and realistic fMRI datasets for a wide range of underlying networks, experimental protocols and problematic confounder in the data. We use these datasets (labelled as NetSim) to compare the performance of our method with other methods. The BOLD signals are generated using dynamic causal modelling (DCM) with nonlinear balloon model for the vascular dynamics (Friston et al. 2003). All network topologies are constructed based on a same building block: a 5-node ring. The ring is a variation of a chain, the head of which is pointed to its tail. The networks with more than 5 nodes are built by linking several 5-node rings together. 28 networks are designed for different purposes of evaluation. The details of these networks are shown in Table 1 of Smith et al. (2011). For each network, 50 simulated subjects are generated by slightly changing the values of parameters. The total number of time points is set to 100.

7.4.1.2 Results

We first investigate the performance of different methods by estimating the resulting c-sensitivity values. The results are summarized in Table 7-1, where each value is the mean value of 50 simulations. ICOV-5 and ICOV-100 represent the ICOV method with the regularization parameter being set to 5 and 100 respectively. This setting gives optimal

results of ICOV according to (Smith et al. 2011). The maximum c-sensitivity value for each network is written in red. From Table 7-1, we can find the elastic PC algorithm has a much better performance than the others. 24 out of 28 networks are best estimated by the elastic PC algorithm. Full correlation and ICOV-100 only work best for one network, while fully partial correlation achieves the maximum c-sensitivity values for two networks. ICOV-100 and ND return the maximum values for three networks, while GS works best for four networks. For the results obtained from elastic PC, 19 networks have c-sensitivity values larger than 85%. The results from all methods for the 11th network are lower than 20%. This is because the simulation dataset for this network is generated in a way that ROIs signals greatly influence all others (Smith et al. 2011).

Table 7-1 Values of c-sensitivity (%) for 28 simulation networks using different methods. The maximum value for each network is written in red.

Network ID	**Full correlation**	**Fully partial correlation**	**ICOV-5**	**ICOV-100**	**ND**	**GS**	**Elastic PC**
1	84.00	92.40	93.60	89.60	92.80	92.00	95.60
2	80.00	86.73	88.36	88.18	91.45	88.18	93.82
3	80.67	82.56	84.78	88.00	88.00	71.22	89.44
4	91.51	77.02	82.95	91.48	91.34	45.97	90.16
5	97.20	99.60	100.00	99.60	100.00	100.00	100.00
6	94.00	99.64	99.82	99.27	99.82	99.64	100.00
7	98.80	100.00	100.00	100.00	100.00	100.00	100.00
8	47.20	65.20	65.20	58.40	62.40	60.40	67.20
9	57.60	81.20	81.20	69.20	74.80	76.00	80.00
10	80.00	96.80	96.80	95.20	95.20	90.80	97.20
11	14.91	11.45	12.91	13.64	13.82	15.27	12.18
12	76.00	83.64	84.55	86.00	87.64	84.73	88.55
13	61.20	61.20	62.00	61.20	64.00	63.20	65.20
14	81.20	94.00	94.40	90.00	93.20	94.80	94.40
15	59.20	89.20	90.00	84.80	80.00	78.00	95.20
16	69.14	85.71	86.57	80.57	86.86	83.14	86.86
17	87.64	92.18	93.27	96.73	96.73	80.18	97.45
18	81.60	91.60	91.60	89.20	92.40	91.60	94.40

19	86.00	94.00	94.40	94.80	95.20	94.80	96.40
20	87.20	95.60	97.20	95.20	98.00	96.00	98.80
21	82.40	89.60	90.40	87.60	90.80	89.20	92.80
22	61.20	74.00	73.60	68.40	72.80	66.80	76.80
23	46.40	73.20	78.00	63.20	70.40	69.20	80.40
24	32.00	41.20	41.60	24.80	38.40	42.40	45.60
25	65.60	68.00	67.20	69.60	72.00	71.60	73.60
26	51.20	53.60	54.00	57.60	57.60	52.80	60.00
27	65.20	68.00	69.20	71.20	74.00	61.60	75.60
28	74.40	83.20	84.00	84.00	84.40	83.20	87.60

We further check the relationship between c-sensitivity and significant level for the elastic PC algorithm. From Figure 7-3, we can see that with the increasing value of significant level most networks can have better performance. Although curves in Figure 7-3 have some local fluctuations, the overall increasing trend can still be clearly observed. An obvious increase for a large fraction of networks occurs at significant level equal to 0.1.

As we have introduced in the method section, the elastic PC algorithm can save computation efforts by avoiding repeated calculations. Some results at the latter significant level can be directly obtained from results at the previous level. Thus, we can investigate how much computation effort has been saved at each significant level. Here, the saved computation effort isrepresented as the proportion of times that line 19 in Algorithm 7.3 is not parsed. Figure 7-4 shows the relationship between saved computation effort and significant level for different networks. We can see that the maximum proportion can reach to 100%. For some networks, the proportion values become 100% at some significant levels. It means that the elastic PC algorithm returns stable results when significant level is above a certain threshold. Hence all results at the latter levels can be directly obtained from previous results. The computation effort is 100% saved in this case.

The proportion of saved efforts varies from 23% to 100% with the mean equal to 84.3%. When the significant level becomes larger than 0.15, 27 out of 28 networks have the proportion values larger than 50%. The largest increase of proportion value for most networks occurs when significant level increases from 0.05 to 0.1. As we have pointed out, there is an observable increase of c-sensitivity in Figure 7-3 occurring at this place. Therefore, the elastic PC algorithm can increase the performance a lot with low cost of computation

efforts. All these observations demonstrate that our algorithm can dramatically save computation efforts while maintaining performance.

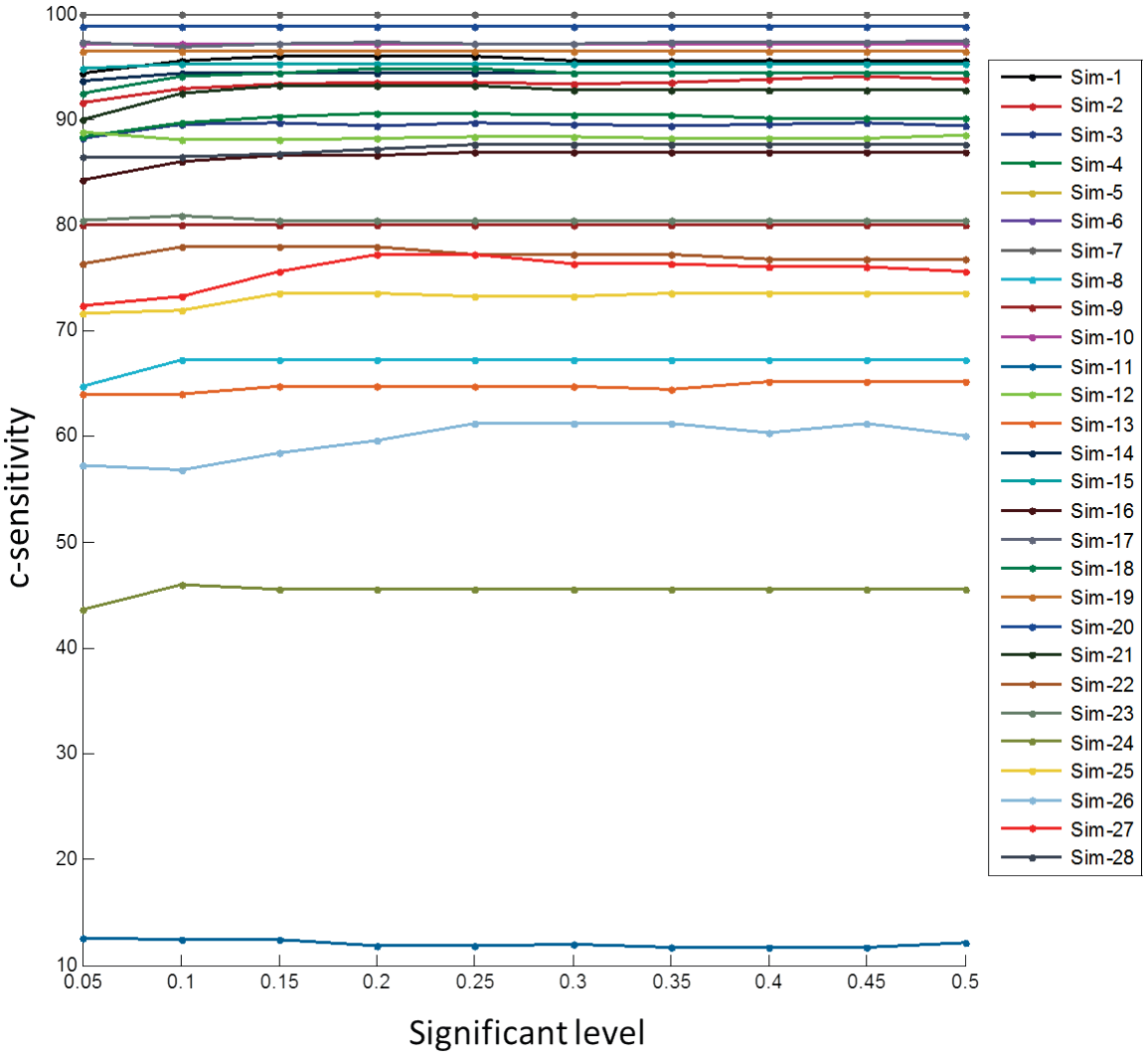


Figure 7-3 Values of c-sensitivity obtained at different significant levels from the elastic PC algorithm for 28 simulated networks.

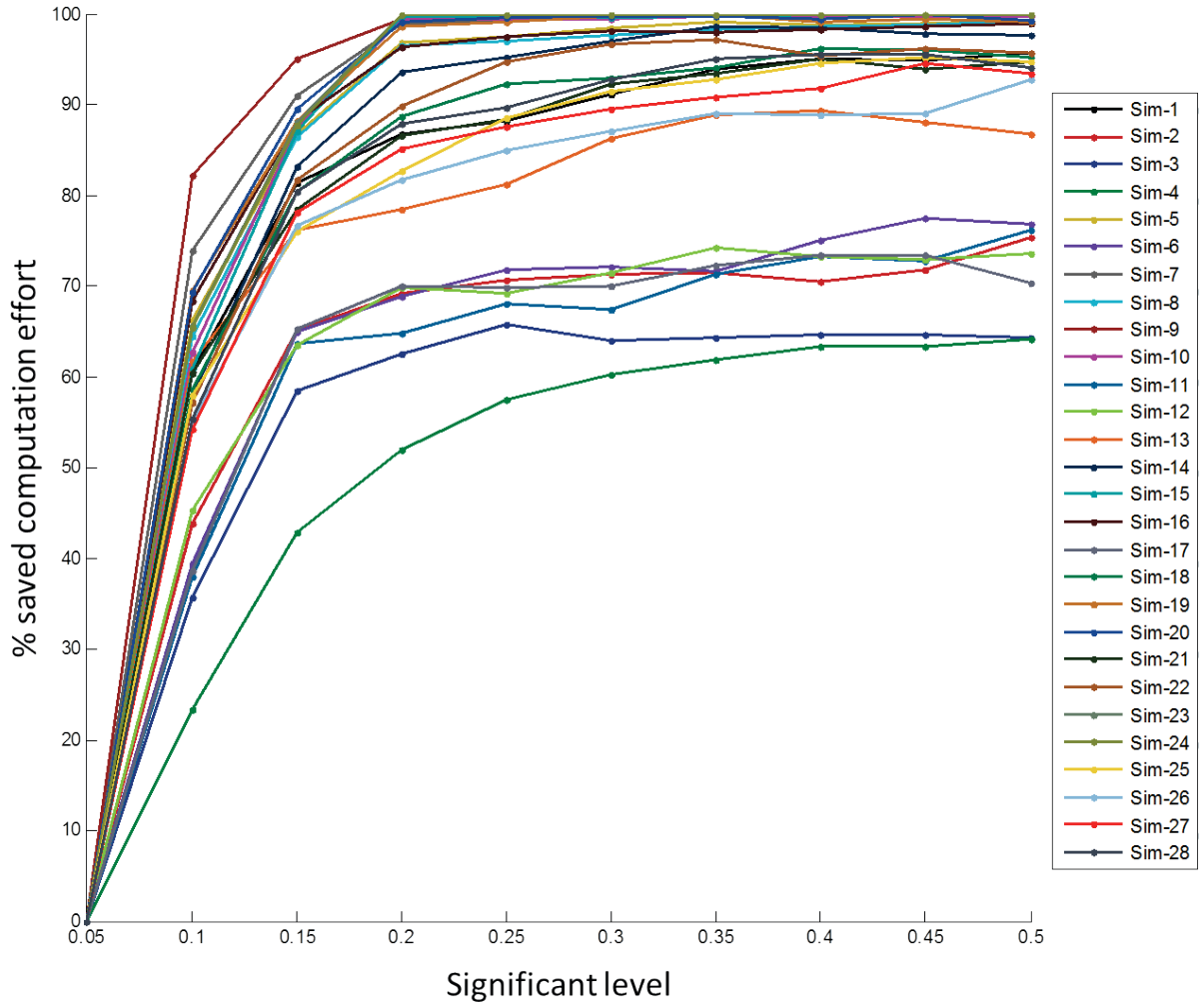


Figure 7-4 The proportion of saved computation efforts at different significant levels from the elastic PC algorithm for 28 simulated networks.

7.4.2 Real dataset analysis

In real dataset analysis, we also compare the elastic PC algorithm with four other methods as we listed above. For the elastic PC algorithm, the initial significant threshold is 0.05 and the step size is 0.05. Ten steps are executed for single subject analysis. For multiple subject analysis, to save computation time only three steps are executed. Thus the significant level ranges from 0.05 to 0.15. For ICOV, the regularisation parameter is set to 6.5, obtained from cross-validation.

7.4.2.1 Data acquisition

We also illustrate our method using the resting-state fMRI data of 10 unrelated subjects from the Human Connectome Project (Van Essen et al. 2013; Stephen M Smith et al. 2013). For

each subject, the functional data are acquired in four approximate-15-minute runs, which are carried out in two separate sessions. The functional images are acquired using a gradient-echo EPI sequence with the following parameters: repetition time (TR) = 720ms, echo time (TE) = 33.1ms, flip angle = 52^o, FOV = 208×180mm, slice thickness = 2.0mm, 72 slices, frames per run = 1200, run duration = 14:33. The data is processed using FSL (Jenkinson et al. 2012) , FreeSurfer (Dale et al. 1999) and Connectome Workbench (Marcus et al. 2013) image analysis suites. The data is de-noised using an independent component analysis based method (Salimi-Khorshidi et al. 2014; Griffanti et al. 2014). The details of the pipelines are presented in (Glasser et al. 2013). For more discussions of resting-state fMRI in the Human Connectome Project please refer to (Stephen M Smith et al. 2013).

The AAL atlas (Tzourio-Mazoyer et al. 2002) is used to parcel a whole brain into to 116 ROIs, which are 90 regions in cerebrum and 26 regions in cerebellum. The time series of a ROI are defined to be the average time series of all voxels in this ROI. The mean of each ROI signal across all time points is centred to zero.

7.4.2.2 Results of single subject analysis

We first randomly select a subject to compare the performance of our method with other methods. As for the real dataset we do not know the ground truth, we cannot use c-sensitivity to evaluate the performance. Therefore, we first show the images of resulting confidence matrices $\widehat{\boldsymbol{\Omega}}$ from different methods. All $\widehat{\boldsymbol{\Omega}}$ matrices are normalized in the same way that all elements range from 0 to 1. Images shown in Figure 7-5 can give us a straightforward overview of different results. In Figure 7-5, 10 images in the first two columns show results of elastic PC at different significant levels; and 5 images in the last column display results of 5 other methods.

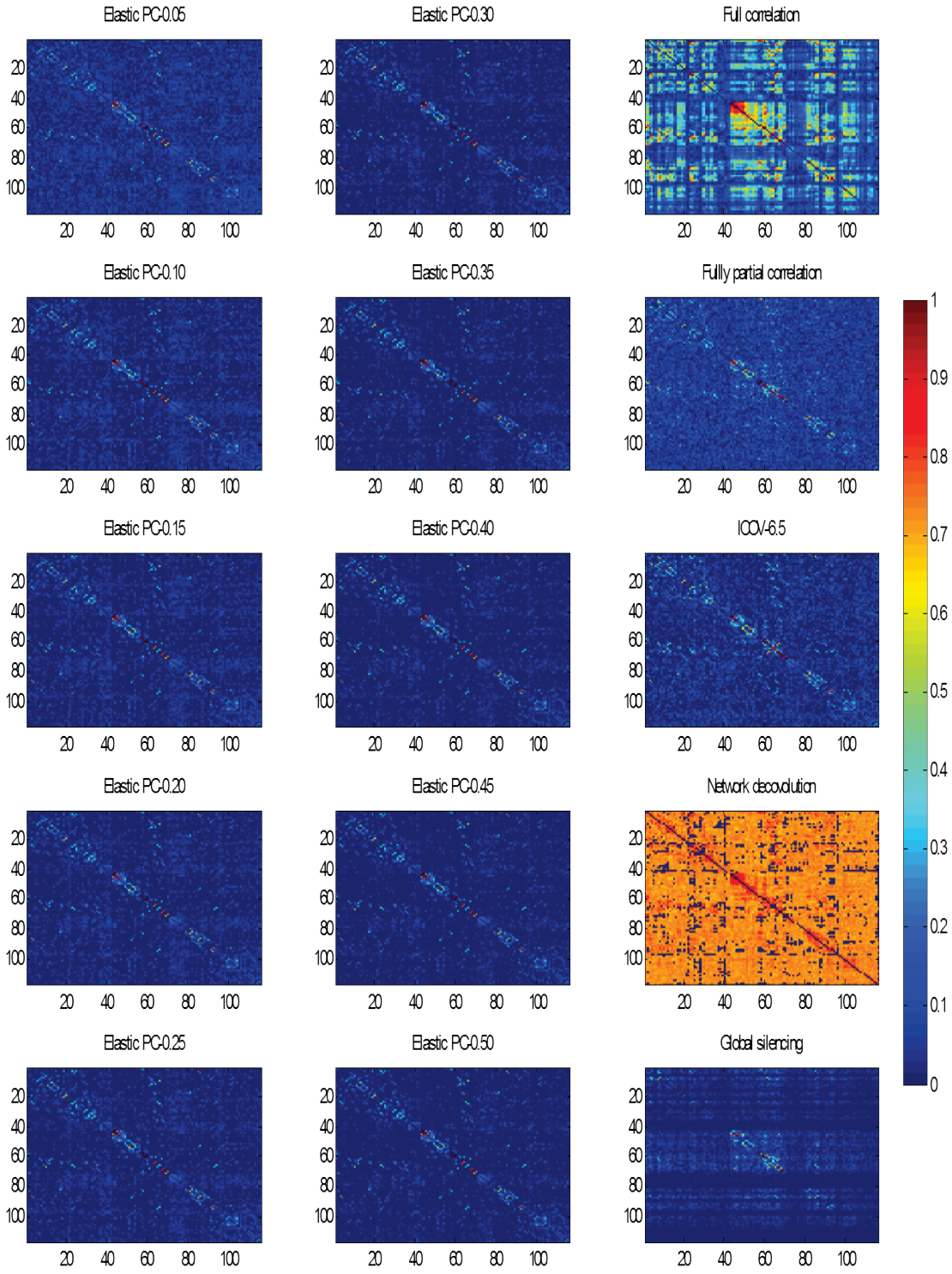


Figure 7-5 Images of matrix $\widehat{\boldsymbol{\Omega}}$ from different methods. All elements in matrices range from 0 and 1. The colour map is shown on the right. Elements with small values are assigned by a cold colour, while elements with large values are assigned by a warm colour.

We can see that with the increasing significant level, more elements in the results approach to zero. The resulting matrix with the significant level of 0.5 is sparsest. There are a lot of

elements consistently large in all elastic PC results, especially elements next to the diagonal. It means that two neighbouring labelled ROIs are correlated. This observation is consistent with the fact that two homotypic regions in the left and right hemispheres are given two adjacent labels in the AAL atlas. For example, labels of the lingual regions in the left and right hemispheres are 47 and 48 respectively. Two homotypic regions are more likely to highly correlate with each other.

Compared with other methods including full correlation, fully partial correlation, ICOV-6.5, ND and GS, results from elastic PC method with significant level larger than 0.1 are sparser than results from all other methods. The ND method works badly in regards of sparsity, while GS method generates unreliable results whose pattern is quite different from others. Fully partial correlation and ICOV-6.5 return results in the similar pattern of elastic PC but less sparse. All these observations can tell us results from the elastic PC algorithm is reasonable and reliable than results from other methods.

Having investigated the results of the elastic PC algorithm, we move on to studying the relationship between the saved computation effort and significant level. As it is shown in Figure 7-6, the proportion of saved computation effort varies from 9.2% to 72%. When the significant level increases from 0.05 to 0.1, only 9.2% computation effort is saved. It means that the elastic PC algorithm does a lot of re-calculations at 0.1. When the significant level further increases above 0.1, the proportion of saved effort is always maintained at a level larger than 50%. Thus, the elastic PC method reaches a relatively stable state, where the number of re-calculations is limited. Figure 7-6 hence shows the efficiency of our method.

Figure 7-5 helps us to visualize the results of the elastic PC algorithm with significant level ranges from 0.05 to 0.5. We can see that the results are similar but become sparser with larger significant level. Now, let us estimate changes between results with two adjacent significant levels. The amounts of changes between two neighbouring significant levels, calculated as the root mean square difference of elements. From Figure 7-7, we can see that the changes at 0.1 is as high as 0.17. When the significant level increases above 0.2, the changes remain at a constant level lower than 0.02. It tells that the results become relatively stable with significant level larger than 0.2. This is consistent with the observation from Figure 7-6 that the proportion of saved computation effort maintains high when the significant level is larger than 0.2.

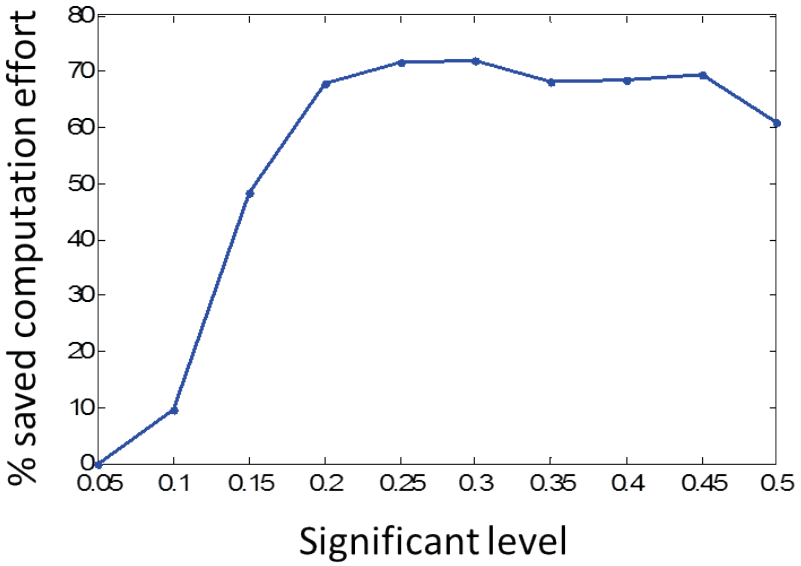


Figure 7-6 The proportion of saved computation efforts at different significant levels from the elastic the PC algorithm for a real fMRI dataset.

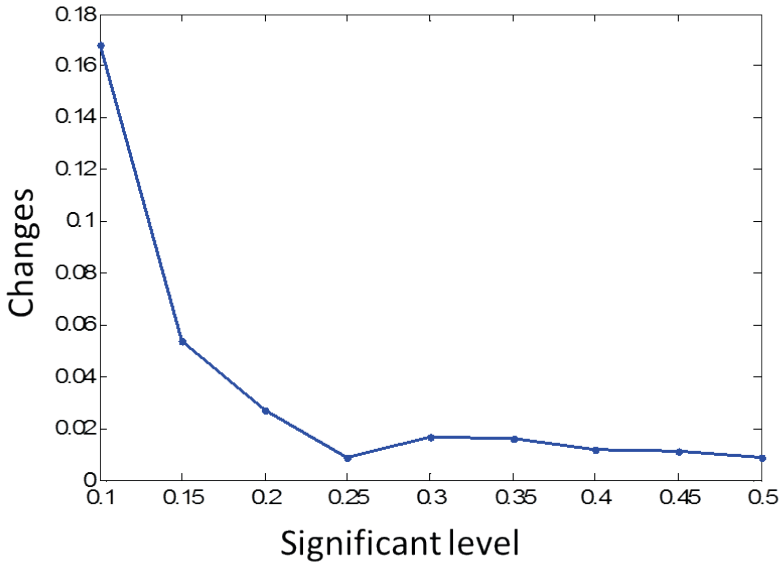


Figure 7-7 The changes between results with two adjacent significant levels from the elastic PC algorithm.

7.4.2.3 Results of multiple subjects analysis

We apply different methods to infer matrices $\hat{\boldsymbol{\Omega}}$ for 10 subjects. All $\hat{\boldsymbol{\Omega}}$ matrices are first normalized that all elements range from 0 to 1. Then we average $\hat{\boldsymbol{\Omega}}$ obtained from each method across 10 subjects. Therefore, we get 6 matrices for 6 different methods including full correlation, fully partial correlation, ICOV, ND and GS. To compare these 6 different results,

we select the top 20 largest elements of each matrix and find their corresponding links. Thus, each method returns 20 most confident links. A lot of links are coherently detected by different methods. There are in total 40 links selected by at least one method.

Table 7-2 shows the details of these links. There are three categories, where links are between: (I) two homotypic regions in the left and right hemispheres; (II) two heterotypic regions in the same hemisphere; (III) two heterotypic regions in different hemispheres. In the table, the last character in 'ROI Name' indicates the hemisphere, where 'L' means the left and 'R' means the right. If a link is found by a certain method, then the corresponding value in the table is set to 1; otherwise it is set to 0. From Table 7-2, we can see that there are 16 Type I links, 22 Type II links and 2 Type III links. We can observe that links discovered by our method are similar with those detected by fully partial correlation and ICOV. In the literature, fully partial correlation and ICOV are the most popular methods to infer direct links. Therefore, our method can basically generate reliable results. Moreover, the results of full correlation are only partly consistent with all other methods. This is because, except full correlation, all other methods aim to detect direct links. The full correlation method has no means in reducing indirect links. Thus its results are less reliable and different from other methods.

Table 7-2 Links detected by different methods. 'Full', 'FP' and 'ePC' represent full correlation, fully partial correlation and elastic PC for short. Links that are coherently detected by all methods are written in red.

Type	ROI Name	ROI Name	Full	FP	ICOV	ND	GS	ePC
I	Frontal_Mid_L	Frontal_Mid_R	0	0	0	0	1	0
	Frontal_Sup_Medial_L	Frontal_Sup_Medial_R	0	1	1	0	0	1
	Frontal_Med_Orb_L	Frontal_Med_Orb_R	0	0	0	0	0	1
	Calcarine_L	Calcarine_R	1	1	1	1	1	1
	Cuneus_L	Cuneus_R	1	1	1	1	1	1
	Lingual_L	Lingual_R	1	0	0	0	0	0
	Occipital_Sup_L	Occipital_Sup_R	1	0	0	0	0	0
	Occipital_Mid_L	Occipital_Mid_R	1	1	1	1	1	1
	Occipital_Inf_L	Occipital_Inf_R	0	1	1	1	0	1
	Postcentral_L	Postcentral_R	1	1	1	1	1	1
	Parietal_Sup_L	Parietal_Sup_R	0	1	1	1	1	1
	SupraMarginal_L	SupraMarginal_R	1	1	1	1	1	1

	Angular_L	Angular_R	0	1	1	0	1	0
	Precuneus_L	Precuneus_R	1	1	1	1	1	1
	Paracentral_Lobule_L	Paracentral_Lobule_R	0	1	1	1	0	1
	Temporal_Sup_L	Temporal_Sup_R	0	1	1	1	0	1
II	Precentral_R	Postcentral_R	1	1	1	1	1	1
	Frontal_Sup_L	Frontal_Mid_L	0	1	1	1	1	1
	Frontal_Sup_R	Frontal_Mid_R	0	1	1	1	1	1
	Frontal_Mid_R	Parietal_Inf_R	1	0	0	1	0	1
	Frontal_Inf_Tri_L	Frontal_Inf_Orb_L	0	0	1	0	0	1
	Calcarine_L	Cuneus_L	1	0	0	0	0	0
	Calcarine_L	Lingual_L	1	0	0	0	0	0
	Calcarine_R	Lingual_R	1	0	0	0	0	0
	Cuneus_L	Lingual_L	1	0	0	0	0	0
	Cuneus_L	Occipital_Sup_L	1	1	0	1	1	0
	Cuneus_R	Lingual_R	1	0	0	0	0	0
	Cuneus_R	Occipital_Sup_R	1	0	0	0	1	0
	Occipital_Sup_L	Occipital_Mid_L	0	0	0	0	1	0
	Occipital_Sup_R	Occipital_Mid_R	1	0	0	0	1	0
	Occipital_Mid_L	Occipital_Inf_L	0	0	0	1	0	0
	Occipital_Mid_R	Parietal_Sup_R	0	0	0	0	1	0
	Parietal_Sup_L	Parietal_Inf_L	0	1	1	1	1	1
	Parietal_Inf_R	Angular_R	0	1	1	1	0	0
	Angular_L	Temporal_Mid_L	0	0	0	0	1	0
	Angular_R	Temporal_Mid_R	0	0	0	0	1	0
	Cerebelum_Crus1_L	Cerebelum_Crus2_L	0	1	1	1	0	1
	Cerebelum_Crus1_R	Cerebelum_Crus2_R	0	1	1	1	0	1
III	Calcarine_L	Lingual_R	1	0	0	0	0	0
	Cuneus_R	Occipital_Sup_L	1	0	0	0	0	0

There are 6 Type I links that are coherently discovered by all 6 different methods, which occur in the following ROIs: Calcarine, Cuneus, Occipital_Mid, Postcentral, SupraMarginal and Precuneus. Compared to the other types, the fraction of Type I links that are consistently detected by different methods is relatively high. For Type II links, only one out of 22 links is discovered by all 6 methods, which occurs between the Precentral and Postcentral regions in the right hemisphere. 10 Type II links are only found by one method.

There are only two Type III links, which are solely found by the full correlation method. These two links occur between: Calcarine_L and Lingual_R, Cuneus_R and Occipital_Sup_L. Figure 7-8 plots these ROIs and also shows links among these regions detected by different methods. We can see that two Type I links appearing in Calcarine and Cuneus are detected by all 6 methods. These two links are quite reliable. Two type II links between Cuneus and Occipital_Sup in each hemisphere are detected by some methods expect elastic PC and ICOV. It may be possible that these two links are indirect links. There are 9 links that are only detected by full correlation. It is quite likely they are indirect links wrongly detected by the full correlation method. For example, there is a reliable link between Cuneus_L and Cuneus_R (detected by all methods) and also some observable effects between Cuneus_L and Occipital_Sup_L (detected by 4 methods). Therefore, there are some indirect effects between Cuneus_R and Occipital_Sup_L. Due to these indirect effects, full correlation may wrongly assign a link between these two ROIs.

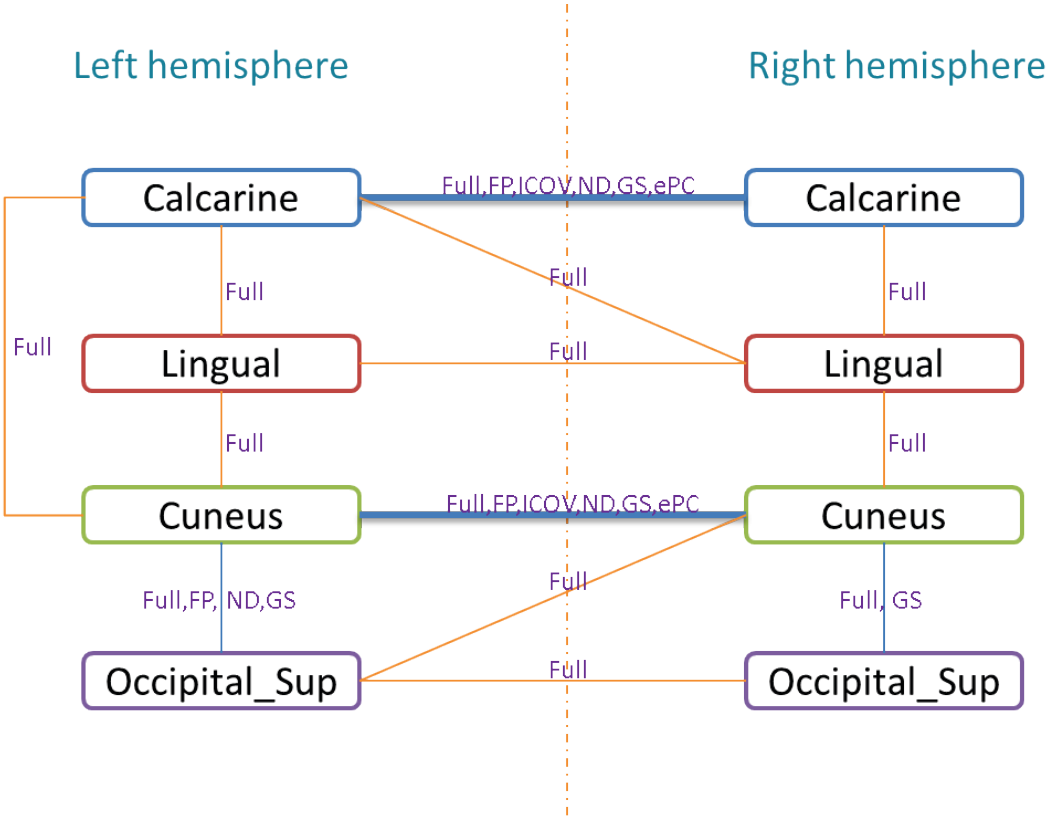


Figure 7-8 Links detected by various methods among the following ROIs: Calcarine, Lingual, Cuneus and Occipital_Sup.

We also investigate the distribution of top 1% links that are found by elastic PC. 67 links are selected with the corresponding elements in $\hat{\boldsymbol{\Omega}}$ ranging from 0.2 to 0.75. We plot these links

using BrainNet Viewer (Xia et al. 2013), as it is shown in Figure 7-9. There are 26 Type I links, 40 Type II links and 1 Type III link. It is interesting to observe that we only obtain a quite limited number of links between two heterotypic regions in different hemispheres. It suggests that in the resting state, the connectivity is majorly composed of Type I and Type II links. We can try to explain our observations as follows. When the brain is in the resting state, the connectivity is ideally in low energy cost mode while can still promptly pass information at the presence of a sudden external stimulus. With the existence of Type I and Type II links, ROIs can be directly connected or connected with one step jump. Therefore, by introducing limited Type III links and massive Type I and Type II links, the minimum path between any two ROIs is short while the degree of connectivity is maintained low, resulting in low energy cost.

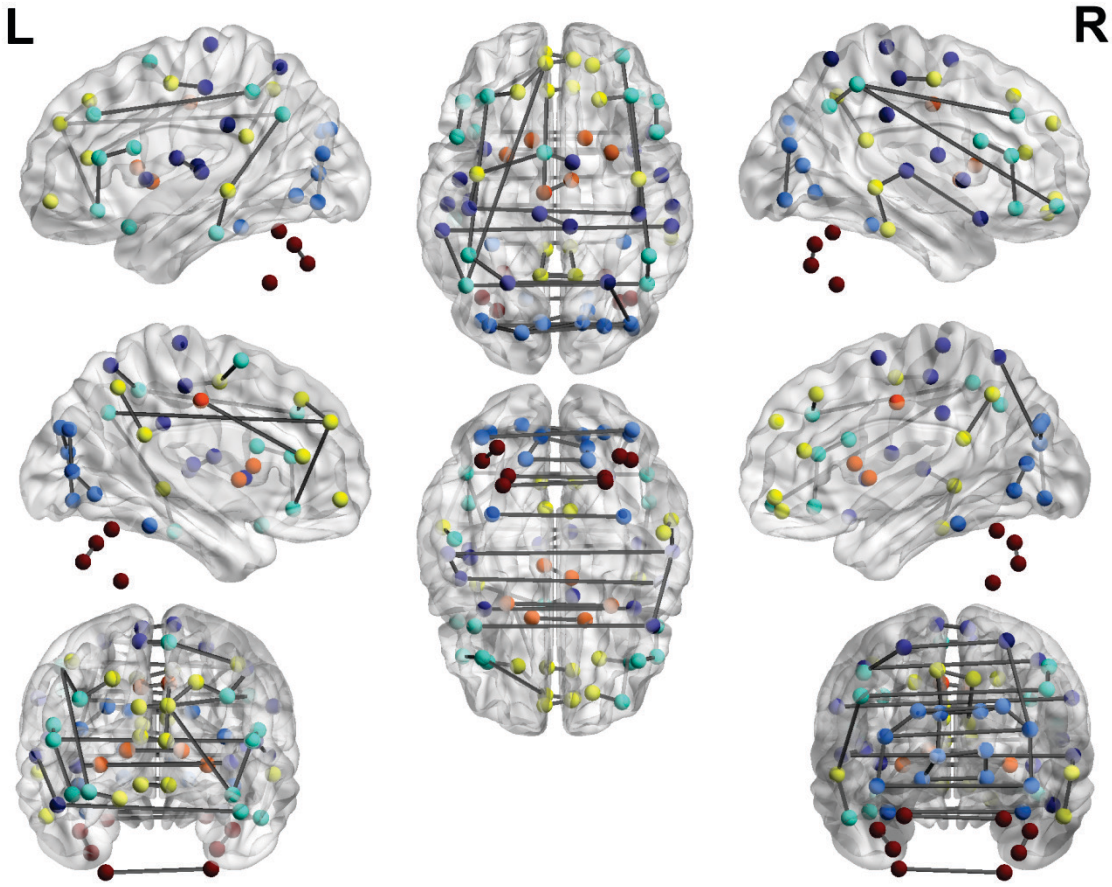


Figure 7-9 The top 1% links selected by the elastic PC algorithm. Nodes outside the cerebrum are in cerebellum. This figure is generated by BrainNet Viewer (Xia et al. 2013) to show all size of brain surfaces, including the lateral and medial sides of each hemisphere, the dorsal and ventral sides, and the anterior and posterior sides of the entire brain.

7.5 Discussion

We propose an elastic PC algorithm, which is derived from the PC method, to approximate minimum partial correlation for connectivity inference. The PC algorithm uses fundamental features of Bayesian network for fast computation: if two nodes can be d-separated by a node set, then these two nodes are conditionally independent; and the node set is a subset of their adjacent nodes. This is because if two nodes are d-separated, then there is no edge between them. Thus each node cannot be a descendent of the other node. One property of Bayesian network is that a node is independent of its nondescendents in the graph given its parents. That is, the node set separating these two nodes contains their parents, which are their adjacent nodes. Although the PC method assumes the graph to be acyclic, we can still apply it to a cyclic graph. For example, the 14^{th} simulation network used in the simulation dataset analysis has cyclic connections, where the c-sensitivity of elastic PC algorithm reaches to 94.4. Hence we could expect that our method can work on cyclic graphs as well.

The original PC algorithm is a parameter dependent method that the choice of significant level greatly determines the results. Ideally, the PC method can perfectly approach to minimum partial correlation with sufficiently large significant level. However, with the increasing significant level, computation cost increases dramatically. The elastic PC algorithm gradually increases the significant level until the up limit of computation time has been reached. Then it outputs the results with the current level. The beauty of the elastic PC algorithm is saving computation effort by reusing results from previous steps. We have shown the percentage of computations has been saved at each step of the elastic PC algorithm.

The aim of the elastic PC algorithm is inferring the dependence of nodes by calculating the minimum partial correlation. Values of MPC directly indicate the confident levels of links. However, there is a potential problem of estimating MPC. That is, the value of minimum partial correlation is always zero if the number of data points is less than or equal to the number of ROIs. This problem is rare in the practice of fMRI studies, because the number of time points of fMRI data is normally more than 300 and the number of ROI is usually equal to or less than 150. However, minimum partial correlation cannot be directly applied for inferring connectivity at voxel level. This is because the number of voxels is much larger than the number of time points (Leonardi et al. 2013; Hutchison et al. 2013). We need to reduce the number voxels before applying MPC. For example, having inferred the connectivity among ROIs, we can see which ROIs are dependent. When generating voxel-level

connectivity, only voxels in dependent ROIs are considered in the controlling set. Thus the size of controlling set can be greatly reduced, which gives a chance to use MPC.

In real fMRI analysis, we use the AAL atlas to define ROIs and apply a straightforward approach to obtain signals of ROIs. In our approach, the time series signal of each ROI is calculated as the mean signal of all voxels in that ROI. There is an argument that the structural atlas-based parcellation approach may not finely correspond to real functional boundaries in the data (S. M. Smith 2012). Some data-driven sophisticated parcellation methods are being used, such as high-dimensional ICA (Fornito et al. 2013). However, recent study shows that the AAL atlas have greater stability than other methods (Wee et al. 2014). Thus, we finally chose the AAL atlas to illustrate our algorithm for its simplicity and interpretability.

In conclusion, we propose the elastic PC algorithm to construct connectivity. The main idea of this method is using minimum partial correlation to estimate the dependence of nodes. The elastic PC algorithm is derived from PC while minimizing the influence of predefined significant level with given computation time. We use a set of simulation networks from the literature to demonstrate the accuracy of our method by calculating c-sensitivity. We also show the efficiency of our method by estimating the proportion of saved computations at each step of the elastic PC algorithm. The performance of our method is compared with 5 other methods. 24 out of 28 simulation networks obtain the best performance with our method. We also use real fMRI datasets collected from HCP to show the applicability of our method. The results of elastic PC are reliable in terms of sparsity.

8 Building linear dynamic models: a sparse learning approach

8.1 Motivation

Dynamic models, consisting of a set of equations describing a function of time and the state of the system, can reveal mechanisms of complex systems. Observation models contain the relationships between measurements and states of dynamic models. With the known observation model, various inference methods can be used to infer dynamic models from measurements. Dynamic model inference is of significant importance in diverse research areas. For example, mechanistic models of biological pathways can describe the time evolution of intracellular molecules in response to stimuli and give a detailed insight into pathway dynamics. Particularly, in drug discovery for diseases, such as cancer, diabetes and asthma, mechanistic models of pathways allow for a better understanding of biological processes of drug effects leading to more efficient treatments (Ribeiro & Pinto 2009).

Two main steps to construct dynamic models are: defining dynamic equations and estimating parameters. In this research, we focus on linear equations to describe the time evolution of state variables. The inferred transition matrix of linear dynamic model indicates the dynamic relationships of system elements. Having decided to use linear equations, the model inference is equivalent to parameter estimation, where each element in the transition matrix is a parameter.

There are a lot of inference methods available by employing technologies of traditional optimization methods (Mendes & Kell 1998), Bayes' inference (Jaakkola & Jordan 2000) and recursive estimators (e.g., Kalman filters) (Baker 2010; Simon & Simon 2003; Sun et al. 2008; Wang et al. 2009). However, these methods cannot return a unique set of parameters or an accurate estimation, when only a fraction of elements in the systems are measured. For example, the work in (Hendriks et al. 2008) can only return hundreds of possible parameters sets for the p38 pathway model. This is a typical underdetermined problem that the number of unknowns is larger than the number of measurements.

To give a solution to the underdetermined system of linear equations, we need to impose extra constraints as appropriate to force the solution of transition matrix only to have a small number of elements with non-zero values. For instance, the study in (Wee et al. 2012) incorporates sparse constraints into connectivity modelling in fMRI analysis. Another example is that (Dai et al. 2009) impose sparse constraints to estimate the expression of genes from combined measurements. In contrast to conventional DNA microarray, where

each genetic sensor responds to a single target, a compressive sensing microarray is designed to combined measure a set of targets.

This chapter proposes an idea that can be potentially benefiting the inference of dynamic models of signalling networks in biological systems. This work is motivated by our previous research in modelling an integrated glucocorticoid receptor (GR) and p38 mitogen activated protein kinase (MAPK) pathway model (Holehouse et al. 2012). In our previous work, we attempt to construct a dynamic model, which can explain the GR resistance of severe asthmatics. However, we cannot return a single model with specific parameters as the time-series measurements are limited. Therefore, it comes into an idea of using sparse learning methods to infer models under limited measurements. The following sections will introduce a plausible way of inferring signalling network inference.

8.2 Background

To understand the hidden mechanism behind complex signalling networks, it is necessary to build dynamic models (Wolkenhauer 2014). The model construction requires quantitative dataset, collected from wet-lab experiments, measuring the dynamic behaviour of networks. Two key issues are important in the aspect of obtaining experimental measurements: perturbation and observation (Hecker et al. 2009). Systematic perturbation of the signalling networks is required to understand dynamic behaviours. Usually, we investigate behaviours of networks by introducing different amounts/types of stimuli/inhibitors to networks. That is, the initial conditions of networks vary under different perturbations. For example, the dynamics of the p38 MAPK pathway, such as Cytokine levels and HSP27 phosphorylation over time, can be detected by investigating its response to LPS (lipopolysaccharide). Thus, different amounts of LPS are introduced to perturb the p38 MAPK pathway. Measurements in each perturbation are carried out at discrete time points. Species that are predefined to be involved in the target network are typically quantified by immunoprecipitation (Bonifacino et al. 2001). The number of time-course measurements and the interval between two sequential measures are crucial in the network modelling. Lack of measurements or too long observation gap will result in impaired observation of real network dynamics and hence incorrect models. Therefore, to obtain a sound network model, we need to carry out sufficient measurements. However, we should take the cost of measurements into consideration as well.

Another aspect of research concerns the inference of crosstalk of local networks. Previously, researchers focus on investigating models of local networks. Local networks are

independently studied by ignoring possible interference from others. Although local networks and simple combinations of local networks can explain and predict some biochemical phenomenon, they lack predictability with respect to the emergent properties and phenotypes. Complex biological processes, such as cell division, differentiation and apoptosis, which are corporately mediated by multiple pathways, cannot be well understood by studying local networks in isolation. With the knowledge of the existence of crosstalk among local networks, the modelling approach orients to construct a more complicated model for an integrated network containing some potentially connected local networks. For example, the GR pathway is essential for regulating anti-inflammatory and immunosuppressive processes. However, post-translational modification of GR, a potential substrate for p38 MAPK, affects nuclear retention of GR as well as transactivation. In some inflammatory diseases, such as severe asthma, the effect of GR pathway as an anti-inflammatory regulator is dramatically impaired when the p38 MAPK pathway is over-activated. Thus it gives rise to the hypothesis that there is crosstalk between the p38 MAPK and GR pathways, and that the study of the integrated p38 MAPK and GR networks can potentially explain the reduced functionality of glucocorticoids in chronic inflammation at the molecular level. Therefore, there are some attempts to construct models which integrating p38 MAPK and GR networks (Holehouse et al. 2012).

Nowadays, there are emerging works to study cell-scale networks that captures emergent property of cellular systems. Recently, a large scale study of crosstalk in a whole-cell model of Mycoplasma genitalium has been generated by incorporating the majority of networks and their interactions at various levels of granularity (Karr et al. 2012). This model is built from over 900 primary sources, reviews, books, and databases. The whole system is divided into 28 subsystems which are all independently built with the assumption that they are all independent on short timescales. Simulations are performed through running submodels independently at each step with some variables of a submodel depended on the value of other submodels at the previous time step. This reductionist approach to the reconstruction of biological systems within a cell is not ideal as it artificially divides the system into many small-scale local networks. Local networks are assumed to act independently from each other, and hence they are reconstructed individually. Crosstalk between local networks is then explored to link all the local networks together to build up a cell-scale network. Such an approximation of the system brings inaccuracy to the modelling. Also the scalability issues arise using this approach as the complexity and size of the networks increase.

Starting from the philosophy of holism, the problem of cell-scale network inference can be considered from a completely different viewpoint based on the fact that the biological system of a cell is indivisible. In the holistic approach, a cell-scale network will be measured and modelled as a whole without artificial divisions. Figure 8-1 compares reductionist and holistic approaches. In Figure 8-1 (a), networks are inferred following the reductionist approach. Suppose three pathways are contained in one cell-scale network. Three different wet-lab experiments are carried out on three pathways independently. Having got three individual models, then two more experiments are used to discover crosstalk among pathways. Afterwards, an integrated model can be constructed. In Figure 8-1 (b), the holistic approach is used. The whole network is measured by a single wet-lab experiment, from which a complete model is inferred directly. This model contains submodels for three pathways and also crosstalk among them. No extra experiments are designed to find a target crosstalk. Instead, it will automatically discover crosstalk that we did not expect to have.

Although the holistic approach can theoretically give a more accurate model than the reductionist approach, there are three practical issues in existing inference methods: 1) scalability issue, 2) measurement issue and 3) overfitting issue. Firstly, there are a huge number of molecular species involved in a cell-scale network. For instance, there are 518 kinases (Manning et al. 2002) and approximately 150 phosphatases (Forrest et al. 2003) working together to mediate the signal transduction network in a human cell. Exhaustively measuring all species in a cell via immunoprecipitation is extremely expensive or even impossible. Although there are high-throughput techniques to measure proteins in signalling networks, they cannot precisely quantify a large number of proteins and especially their post-translational modifications (Hecker et al. 2009). For example, the measurement of post-translational modified proteins in signalling networks using the emerging mass spectrometry technique is highly dependent on enrichment methods whose performance is influenced by various factors (Dunn et al. 2010). Secondly, it is impractical to measure all individual molecules in a cell-scale network due to their various post-translational modifications and complex formation. For example, unphosphorylated STAT5, tyrosine phosphorylated monomeric STAT5 and tyrosine phosphorylated dimeric STAT5 in the JAK-STAT signalling pathway are difficult to assess individually, which affects pathway integrity (Swameye et al. 2003). Thirdly, there are a limited number of studies attempted to provide cell-scale networks. The structural prior knowledge of cell-scale networks is scarce. It has been reported that the

existing inference methods are likely to over-fit for experimental data without structural constraints (Xiong & Choe 2008).

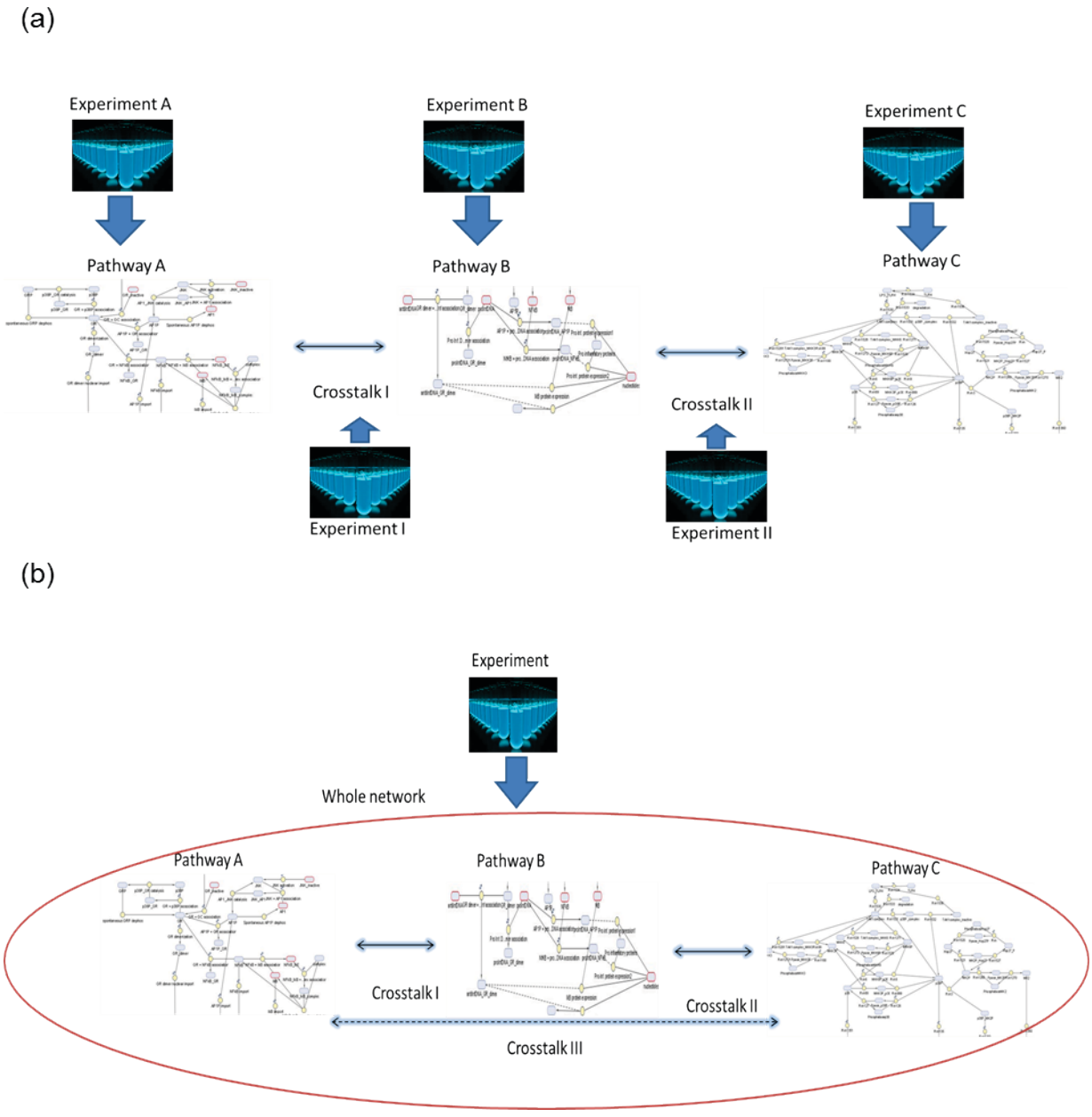


Figure 8-1 Comparison of reductionist and holistic approaches. a) the reductionist approach to study networks; b) the holistic approach to study network. In b) Crosstalk I and II are crosstalk we expect to get; Crosstalk III is the one we may get, benefitted from the holistic approach.

Due to the above issues, the methodology of measuring and inferring cell-scale networks requires disruptive changes. The work proposed in this chapter is exactly to develop such a

novel methodology. Our new methodology is inspired from the following two key observations:

1. Sparsity of variations can be obtained by separation of timescales. As different biological reactions occur over various timescales, the concentrations of only a few species in a cell vary significantly in each specific timescale whilst a large fraction of species remain stable. For example, the receptor internalization (10^2 s) process triggers phosphorylation and catalysis of proteins (<1s) that in turn translocate into cell nucleus and induce their target gene expression; the transcriptional regulation process (10^2 s), acting as a linkage point, stimulates signal cascading of other signalling pathways (Papin & Palsson 2004). As a result, the concentrations of species on fast timescales are maintained in an approximate steady-state level over slower timescales, and vice versa. Let us define the variations of concentrations as the differences between concentrations of adjacent time points. Then a large number of variations of concentrations are zero or approximate zero from a measurement taken under a specific timescale. We call this property as "sparsity of variations".
2. Combined-measurement can be achieved via cross-reactivity. Molecules which share some structural similarities can be combined measured together. Due to the cross-reactivity of an antibody, the antibody may bind not only the targeted protein but also other proteins, such as its various molecular forms or to other proteins in complex with the target protein (Frank 2002). This phenomenon frequently affects measurements of the concentration of the target protein in an immunoprecipitation assay. Traditionally, the ideal way is to use an antibody with a high specific affinity under stringent binding conditions in order to obtain accurate results. In contrast to require highly specific antibodies, we attempt to make use of its cross-reactivity in order to measure the aggregated concentrations of several proteins in one go. We call this experimental method a "combined-measurement".

These two key observations motivate us to use sparse signal recovery methods (e.g.,(Candès et al. 2006; Donoho 2006; Candes & Tao 2006)) to infer cell-scale networks. These methods aim at finding the sparsest solution of underdetermined linear systems, based on the following two conditions: (I) the signal is sparse in some domain; (II) a measurement is a weighted linear combination of several points of the signal. Figure 8-2 gives an illustration of sparse signal recovery.

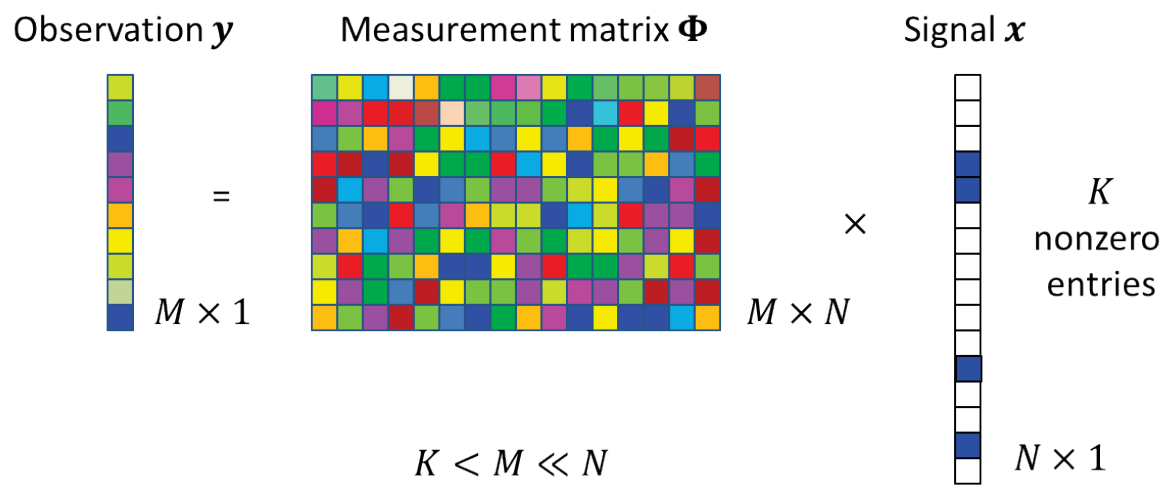


Figure 8-2 An illustration of sparse signal recovery. The signal $\boldsymbol{x}$ has the dimension of $N \times 1$, where the number of non-zero elements K is far less than N. The matrix $\boldsymbol{\Phi}$ is the measurement matrix with the dimension of $M \times N$. The inner product of the signal $\boldsymbol{x}$ and the measurement matrix $\boldsymbol{\Phi}$ gives M weighted sums, represented by the observation vector $\boldsymbol{y}$. With the sparse constraint, the N dimensional signal $\boldsymbol{x}$ can be recovered by using only M observations.

Sparsity of variations and combined-measurement meet the two prerequisites of sparse signal recovery respectively. To obtain the sparsity of variations, we tend to investigate cell-scale networks at different timescales separately, such as seconds, minutes, hours, days and weeks. In each timescale, the combined-measurement is designed by considering the cross-activity of antibodies. The model inferred in each timescale explains the observed dynamic relationships among molecules. In practise, for example, a network model for severe asthmatics in the timescale of minutes can describe and predict patients' immediate response to drugs. The network model in the timescale of days can explain patients' prolonged response to drugs. These two models can tell us whether asthma symptoms can be alleviated with the introduction of drugs and how long will the effectiveness of drugs last for. The idea of investigating networks at different timescale is similar to the concept of adiabatic approximation in quantum mechanics, which states that the reaction of a physical system to a perturbation depends on the timescale of the perturbation (Griffiths 2005).

In this chapter, we focus on investigating the method for network inference under a specific timescale. The proposed method, which is called CCELL, is based on a sparse signal recovery method, sparse Bayesian learning (Tipping 2001; Ji et al. 2008). The following part

of this chapter will explain this method in detail and then show some simulation results to demonstrate our method.

8.3 Method

In our work, networks are modelled by linear equations. Rather than investigating various kinetic equations to model biological functions, we simplify the network model using linear equations and focus on inferring parameters in the linear model. The reason of using simple linear models in our work is that a more complex modelling makes much harder to optimise parameters. It is because the size of searching space increases exponentially with the number unknowns in the model. This is a popular issue in modelling, called the problem of dimensionality. By using a less complex model, such as the linear model, the searching space can be significantly reduced and deterministic models are more likely to be obtained.

In our model, the real-valued observation at discrete time $t \in \{1,..,T\}$ is labelled as an $M \times 1$ vector $\boldsymbol{y}_t$, where the hidden state variables (such as concentration of proteins) are represented by an $N \times 1$ vector $\boldsymbol{x}_t$. The dimension of $\boldsymbol{x}_t$ equals to the number of species in the network, where the dimension of $\boldsymbol{y}_t$ is equivalent to the number of measurements carried out at each discrete time point. The relationship between $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ is:

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + N(0, \sigma_s^2 \boldsymbol{I}) \quad \text{for} \quad t \in \{2,..,T\} \tag{8.1}$$

and

$$\boldsymbol{y}_t = \boldsymbol{C}\boldsymbol{x}_t + N(0, \sigma_m^2 \boldsymbol{I}) \quad \text{for} \quad t \in \{1,..,T\} \tag{8.2}$$

where the transition matrix $\boldsymbol{A}$ is an $N \times N$ matrix that we want to infer, and σ_s^2 and σ_m^2 are the covariance for the state and measurement noises. The measurement matrix is denoted by the $M \times N$ matrix $\boldsymbol{C}$, where each row indicates one combined measurement. For example, if a row of $\boldsymbol{C}$ has the value of $(0,1,0,1,1,0,0)$, it means that the 2nd, 4th and 5th molecules are measured together.

According to our model, concentrations of molecules at the previous time point $t-1$ determines the concentrations at the next time point t via the transition matrix $\boldsymbol{A}$. The linear model in Equation 8.1 is of the similar form with the multivariate autoregressive model (MAR) by simply setting L in the following equation to be 1.

$$\boldsymbol{x}_t = \sum_{\tau=1}^{L} \boldsymbol{A}\boldsymbol{x}_{t-\tau} + N(0, \sigma_s^2 \boldsymbol{I}) \quad \text{for} \quad t \in \{2,..,T\} \tag{8.3}$$

MAR models quantify the linear dependence of one molecule upon all others in the network. It characterizes dependences within the data, especially in terms of historical influence that one variable has on another (Harrison et al. 2003). When $\boldsymbol{x}_t$ for $t \in \{1,..,T\}$ is known, our aim is to infer the transition matrix $\boldsymbol{A}$. We add sparse constraint on $\boldsymbol{A}$ in order to get the majority of elements in the inferred $\boldsymbol{A}$ to be zero. Suppose $a_{j,i}$ be the element of $\boldsymbol{A}$ in the jth row and ith column. In our inference results, if $a_{j,i}$ is significantly nonzero, it can indicate that the information contained in the ith state variable can be used to predict the jth state variable, while this information cannot be obtained from other variables. We can roughly say non-zero elements in $\boldsymbol{A}$ represent the corresponding state variables having irreducible effect on others.

Now, let us introduce the CCELL method. The whole inference process is depicted in Figure 8-3. We first jointly infer the changes in concentration of species at all time points using observations $\boldsymbol{y}_t$ for $t \in \{1,..,T\}$ and the measurement matrix $\boldsymbol{C}$. By checking the changes in concentration values we can find which species are changing over time. We call them as active or time-variant species. Then we only focus on inferring the transition matrix for active species, $\widehat{\boldsymbol{A}}$. Afterwards, we can derive the complete transition matrix $\boldsymbol{A}$ from $\widehat{\boldsymbol{A}}$.

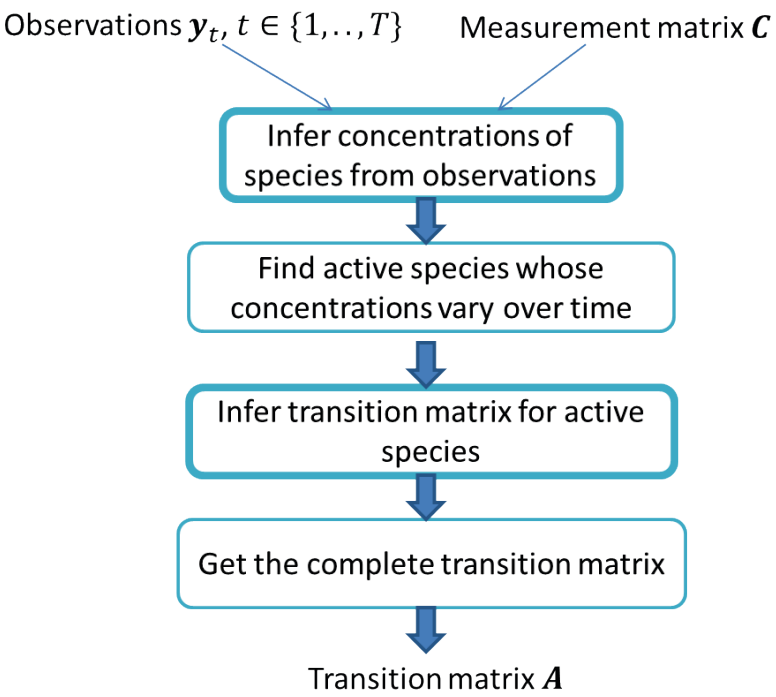


Figure 8-3 The flowchart of the whole inference process.

Please note that the whole inference process is carried out in two steps. After the first step has been finished, we can obtain the changes in concentration of species. In this chapter, we adopt linear difference equations to model the system. In the future, we can use more complicated model (such as ordinary differential equations) and apply any other inference methods as we have discussed in subsection 3.6.2 to infer the model using the concentrations inferred in the first step. This chapter uses the linear model as an example. The proposed process of inference can incorporate modifications of model and changes of inference method.

8.3.1 Inferring concentrations of species from observations

The first step of the CCELL method is inferring $\boldsymbol{x}_t$ from $\boldsymbol{y}_t$ and $\boldsymbol{C}$ for all $t \in \{1, .., T\}$. As we have mentioned in the introduction part, one important feature of a network under a specific timescale is the sparsity of variations. That is, the difference of concentrations between two subsequent time points is sparse. Let us denote the variations of species by $\Delta\boldsymbol{x}_t$:

$$\Delta\boldsymbol{x}_t = \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \tag{8.4}$$

and

$$\Delta\boldsymbol{y}_t = \boldsymbol{y}_t - \boldsymbol{y}_{t-1} \tag{8.5}$$

for $t \in \{2, .., T\}$.

Then we reformulate Equation 8.2 as follows

$$\Delta\boldsymbol{y}_t = \boldsymbol{C}\Delta\boldsymbol{x}_t + N(0, 2\sigma_m^2\boldsymbol{I}) \tag{8.6}$$

For simplicity, let $\beta_t^{-1} = 2\sigma_m^2$ for the following notations. As $\Delta\boldsymbol{x}_t$ is sparse, sparse Bayesian learning can be used. We will first describe the SBL algorithm, from which we derive our own method, Group SBL.

8.3.1.1 SBL: independently infer $\Delta\boldsymbol{x}_t$ for $\boldsymbol{t} \in \{\boldsymbol{2}, .., \boldsymbol{T}\}$

SBL can be directly applied to individually infer $\Delta\boldsymbol{x}_t$ for each time point t from $\Delta\boldsymbol{y}_t$. At each discrete time t, we define a Gaussian prior to $\Delta\boldsymbol{x}_t$ with zero mean as

$$p(\Delta\boldsymbol{x}_t|\boldsymbol{\theta}^t) = \prod_{n=1}^{N}(2\pi)^{-\frac{1}{2}}(\theta_n^t)^{\frac{1}{2}}\exp\left\{-\frac{\theta_n^t}{2}(\Delta x_t^n)^{\boldsymbol{2}}\right\} \tag{8.7}$$

where Δx_t^n is the nth element of $\Delta\boldsymbol{x}_t$ and $\boldsymbol{\theta}^t = \{\theta_1^t, \ldots, \theta_N^t\}$ contains inverse variance of Δx_t^n.

For any t, the likelihood of $\Delta\boldsymbol{y}_t$ conditioned on $\Delta\boldsymbol{x}_t$ and $\boldsymbol{C}$ also follows a Gaussian distribution as:

$$p(\Delta \boldsymbol{y}_t|\Delta \boldsymbol{x}_t, \beta_t) = (2\pi)^{-\frac{M}{2}}(\beta_t)^{\frac{M}{2}} \exp\left\{-\frac{\beta_t}{2}\|\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{x}_t\|^2\right\} \quad 8.8$$

According to Bayes' rule, the posterior distribution of $\Delta \boldsymbol{x}_t$ over parameters is

$$p(\Delta \boldsymbol{x}_t|\Delta \boldsymbol{y}_t, \boldsymbol{\theta}^t, \beta_t) = \frac{p(\Delta \boldsymbol{y}_t|\Delta \boldsymbol{x}_t, \beta_t)p(\Delta \boldsymbol{x}_t|\boldsymbol{\theta}^t)}{p(\Delta \boldsymbol{y}_t|\beta_t, \boldsymbol{\theta}^t)} \quad 8.9$$

As $p(\Delta \boldsymbol{x}_t|\boldsymbol{\theta}^t)$ and $p(\Delta \boldsymbol{y}_t|\Delta \boldsymbol{x}_t, \beta_t)$ are conjugate, the posterior distribution is also Gaussian (Berger 1985). Suppose the posterior distribution of $\Delta \boldsymbol{x}_t$ over parameters is

$$p(\Delta \boldsymbol{x}_t|\Delta \boldsymbol{y}_t, \boldsymbol{\theta}^t, \beta_t) = N(\boldsymbol{\mu}_{\Delta x_t}, \boldsymbol{\Sigma}_{\Delta x_t}) \quad 8.10$$

where $\boldsymbol{\mu}_{\Delta x_t}$ and $\boldsymbol{\Sigma}_{\Delta x_t}$ are the mean and covariance of $\Delta \boldsymbol{x}_t$. To get $\boldsymbol{\mu}_{\Delta x_t}$ and $\boldsymbol{\Sigma}_{\Delta x_t}$, we can carry out the following inference:

$$\begin{aligned}
&\ln p(\Delta \boldsymbol{x}_t|\Delta \boldsymbol{y}_t, \boldsymbol{\theta}^t, \beta_t) \\
&= \ln p(\Delta \boldsymbol{y}_t|\Delta \boldsymbol{x}_t, \beta_t) + \ln p(\Delta \boldsymbol{x}_t|\boldsymbol{\theta}^t) + const \\
&= -\frac{\beta_t}{2}\|\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{x}_t\|^2 - \sum_{n=1}^{N}\frac{1}{2}\theta_n^t(\Delta x_t^n)^2 + const \\
&= -\tfrac{\beta_t}{2}(\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{x}_t)'(\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{x}_t) - \tfrac{1}{2}\Delta \boldsymbol{x}_t{}'\mathrm{diag}(\boldsymbol{\theta}^t)\Delta \boldsymbol{x}_t + const \\
&= -\tfrac{\beta_t}{2}(\Delta \boldsymbol{y}_t{}' - \Delta \boldsymbol{x}_t{}'\boldsymbol{C}')(\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{x}_t) - \tfrac{1}{2}\Delta \boldsymbol{x}_t{}'\mathrm{diag}(\boldsymbol{\theta}^t)\Delta \boldsymbol{x}_t + const \\
&= \\
&-\tfrac{\beta_t}{2}(\Delta \boldsymbol{y}_t{}'\Delta \boldsymbol{y}_t - \Delta \boldsymbol{y}_t{}'\boldsymbol{C}\Delta \boldsymbol{x}_t - \Delta \boldsymbol{x}_t{}'\boldsymbol{C}'\Delta \boldsymbol{y}_t + \Delta \boldsymbol{x}_t{}'\boldsymbol{C}'\boldsymbol{C}\Delta \boldsymbol{x}_t) - \tfrac{1}{2}\Delta \boldsymbol{x}_t{}'\mathrm{diag}(\boldsymbol{\theta}^t)\Delta \boldsymbol{x}_t + \\
&const \\
&= -\tfrac{1}{2}\Delta \boldsymbol{x}_t{}'\big(\beta_t\boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)\big)\Delta \boldsymbol{x}_t + \tfrac{1}{2}(\beta_t\Delta \boldsymbol{y}_t{}'\boldsymbol{C}\Delta \boldsymbol{x}_t + \beta_t\Delta \boldsymbol{x}_t{}'\boldsymbol{C}'\Delta \boldsymbol{y}_t) - \\
&\tfrac{\beta_t}{2}\Delta \boldsymbol{y}_t{}'\Delta \boldsymbol{y}_t + const \\
&= -\tfrac{1}{2}\Delta \boldsymbol{x}_t{}'\big(\beta_t\boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)\big)\Delta \boldsymbol{x}_t + \tfrac{1}{2}(\beta_t\Delta \boldsymbol{y}_t{}'\boldsymbol{C}\Delta \boldsymbol{x}_t + \Delta \boldsymbol{x}_t{}'\beta_t\boldsymbol{C}'\Delta \boldsymbol{y}_t) + const
\end{aligned} \quad 8.11$$

The operator $(.)'$ means the transpose of a matrix. Let us recall that the logarithm of the multivariate Gaussian distribution is of the following form:

$$\begin{aligned}
&-\frac{1}{2}(\Delta \boldsymbol{x}_t - \boldsymbol{\mu}_{\Delta x_t})'\boldsymbol{\Sigma}_{\Delta x_t}{}^{-1}(\Delta \boldsymbol{x}_t - \boldsymbol{\mu}_{\Delta x_t}) \\
&= -\frac{1}{2}\Delta \boldsymbol{x}_t'\boldsymbol{\Sigma}_{\Delta x_t}{}^{-1}\Delta \boldsymbol{x}_t + \frac{1}{2}(\boldsymbol{\mu}_{\Delta x_t}{}'\,\boldsymbol{\Sigma}_{\Delta x_t}{}^{-1}\Delta \boldsymbol{x}_t + \Delta \boldsymbol{x}_t'\boldsymbol{\Sigma}_{\Delta x_t}{}^{-1}\boldsymbol{\mu}_{\Delta x_t}) + const
\end{aligned} \quad 8.12$$

By comparing Equation 8.11 with 8.12, we get

$$\boldsymbol{\mu}_{\Delta x_t} = \beta_t \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{C}' \Delta \boldsymbol{y}_t$$
$$\boldsymbol{\Sigma}_{\Delta x_t} = (\beta_t \boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t))^{-1} \quad 8.13$$

Equation 8.10 and 8.13 show how to get the posterior distribution of variations of species $\Delta \boldsymbol{x}_t$ with defined parameter β_t (inverse variance of the noise) and $\boldsymbol{\theta}^t$ (inverse variance of the prior distribution of $\Delta \boldsymbol{x}_t$)**.**

Now, let us discuss how to get β_t and $\boldsymbol{\theta}^t$. SBL chooses the optimal parameter by maximising $p(\Delta \boldsymbol{y}_t | \beta_t, \boldsymbol{\theta}^t)$, which represents the probability of observation. $p(\Delta \boldsymbol{y}_t | \beta_t, \boldsymbol{\theta}^t)$ is obtained by integrating $\Delta \boldsymbol{x}_t$:

$$p(\Delta \boldsymbol{y}_t | \beta_t, \boldsymbol{\theta}^t) = \int p(\Delta \boldsymbol{y}_t | \Delta \boldsymbol{x}_t, \beta_t) p(\Delta \boldsymbol{x}_t | \, \boldsymbol{\theta}^t) d \, \Delta \boldsymbol{x}_t \quad 8.14$$

Because this represents the convolution of two Gaussians, it is readily evaluated to give the log from as:

$$\begin{aligned} \ln p(\Delta \boldsymbol{y}_t | \beta_t, \boldsymbol{\theta}^t) &= \ln N(\Delta \boldsymbol{y}_t | \boldsymbol{0}, \boldsymbol{\Omega}) \\ &= -\frac{1}{2}\{M \ln(2\pi) + \ln|\boldsymbol{\Omega}| + \Delta \boldsymbol{y}_t{}' \boldsymbol{\Omega}^{-1} \Delta \boldsymbol{y}_t\} \end{aligned} \quad 8.15$$

To get the expression of the covariance matrix $\boldsymbol{\Omega}$, we do the following inference:

$$\begin{aligned} p(\Delta \boldsymbol{y}_t | \beta_t, \boldsymbol{\theta}^t) &= \int p(\Delta \boldsymbol{y}_t | \Delta \boldsymbol{x}_t, \beta_t) p(\Delta \boldsymbol{x}_t | \, \boldsymbol{\theta}^t) d \, \Delta \boldsymbol{x}_t \\ &= \int (2\pi)^{-\frac{M}{2}} (\beta_t)^{\frac{M}{2}} \exp\left\{-\frac{\beta_t}{2} \| \Delta \boldsymbol{y}_t \right. \\ &\quad \left. - \boldsymbol{C} \Delta \boldsymbol{x}_t \|^2 \right\} \cdot \prod_{n=1}^{N} (2\pi)^{-\frac{1}{2}} (\theta_n^t)^{\frac{1}{2}} \exp\left\{-\frac{\theta_n^t}{2} (\Delta x_t^n)^2\right\} d \Delta \boldsymbol{x}_t \\ &= \left(\frac{\beta_t}{2\pi}\right)^{\frac{M}{2}} \prod_{n=1}^{N} \left(\frac{\theta_n^t}{2\pi}\right)^{\frac{1}{2}} \int \exp\{-E(\Delta \boldsymbol{x}_t)\} \, d \Delta \boldsymbol{x}_t \end{aligned} \quad 8.16$$

where we have defined

$$\begin{aligned} E(\Delta \boldsymbol{x}_t) &= \frac{\beta_t}{2} \| \Delta \boldsymbol{y}_t - \boldsymbol{C} \Delta \boldsymbol{x}_t \|^2 + \frac{1}{2} \Delta \boldsymbol{x}_t{}' \mathrm{diag}(\boldsymbol{\theta}^t) \Delta \boldsymbol{x}_t \\ &= \tfrac{1}{2} \Delta \boldsymbol{x}_t{}' \big(\beta_t \boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)\big) \Delta \boldsymbol{x}_t - \tfrac{1}{2} (\beta_t \Delta \boldsymbol{y}_t{}' \boldsymbol{C} \Delta \boldsymbol{x}_t + \beta_t \Delta \boldsymbol{x}_t{}' \boldsymbol{C}' \Delta \boldsymbol{y}_t) + \tfrac{\beta_t}{2} \Delta \boldsymbol{y}_t{}' \Delta \boldsymbol{y}_t \end{aligned} \quad 8.17$$

Let us reformat Equation 8.17 into the form of the square over $\Delta \boldsymbol{x}_t$ as follows:

$$E(\Delta \boldsymbol{x}_t) = E(\boldsymbol{m}) + \frac{1}{2} (\Delta \boldsymbol{x}_t - \boldsymbol{m})' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} (\Delta \boldsymbol{x}_t - \boldsymbol{m}) \quad 8.18$$

$$= \frac{1}{2}\Delta \boldsymbol{x}_t{}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \Delta \boldsymbol{x}_t - \frac{1}{2}\left(\boldsymbol{m}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \Delta \boldsymbol{x}_t + \Delta \boldsymbol{x}_t{}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \boldsymbol{m}\right) + \frac{1}{2}\boldsymbol{m}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \boldsymbol{m} + E(\boldsymbol{m})$$

Comparing Equation 8.17 and 8.18, we get

$$\boldsymbol{m} = \beta_t \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{C}' \Delta \boldsymbol{y}_t \tag{8.19}$$

and

$$E(\boldsymbol{m}) = \frac{\beta_t}{2}\Delta \boldsymbol{y}_t{}' \Delta \boldsymbol{y}_t - \frac{1}{2}\boldsymbol{m}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \boldsymbol{m} \tag{8.20}$$

Using Equation 8.18, we can evaluate the integral in Equation 8.16

$$\int \exp\{-E(\Delta \boldsymbol{x}_t)\}\, d\Delta \boldsymbol{x}_t = \exp\{-E(\boldsymbol{m})\}\, (2\pi)^{\frac{N}{2}} |\boldsymbol{\Sigma}_{\Delta x_t}|^{1/2} \tag{8.21}$$

Now we only need to deal with $\exp\{-E(\boldsymbol{m})\}$, where $E(\boldsymbol{m})$ can be obtained as follows:

$$\begin{aligned} E(\boldsymbol{m}) &= \frac{1}{2}(\beta_t \Delta \boldsymbol{y}_t{}' \Delta \boldsymbol{y}_t - \boldsymbol{m}' \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \boldsymbol{m}) \\ &= \frac{1}{2}(\beta_t \Delta \boldsymbol{y}_t{}' \Delta \boldsymbol{y}_t - \beta_t \Delta \boldsymbol{y}_t{}' \boldsymbol{C} \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{\Sigma}_{\Delta x_t}{}^{-1} \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{C}' \Delta \boldsymbol{y}_t \beta_t) \\ &= \frac{1}{2}\Delta \boldsymbol{y}_t{}' (\beta_t \boldsymbol{I} - \beta_t \boldsymbol{C} \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{C}' \beta_t) \Delta \boldsymbol{y}_t \\ &= \frac{1}{2}\Delta \boldsymbol{y}_t{}' \left\{\beta_t \boldsymbol{I} - \beta_t \boldsymbol{C}\left(\beta_t \boldsymbol{C}' \boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)\right)^{-1} \boldsymbol{C}' \beta_t\right\} \Delta \boldsymbol{y}_t \end{aligned} \tag{8.22}$$

Since

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \tag{8.23}$$

we can get

$$E(\boldsymbol{m}) = \frac{1}{2}\Delta \boldsymbol{y}_t{}' (\beta_t{}^{-1} \boldsymbol{I} + \boldsymbol{C}\left(\mathrm{diag}(\boldsymbol{\theta}^t)\right)^{-1} \boldsymbol{C}')^{-1} \Delta \boldsymbol{y}_t \tag{8.24}$$

Let

$$\boldsymbol{\Omega} = \beta_t{}^{-1} \boldsymbol{I} + \boldsymbol{C}\left(\mathrm{diag}(\boldsymbol{\theta}^t)\right)^{-1} \boldsymbol{C}' \tag{8.25}$$

then,

$$E(\boldsymbol{m}) = \frac{1}{2}\Delta \boldsymbol{y}_t{}' \boldsymbol{\Omega}^{-1} \Delta \boldsymbol{y}_t \tag{8.26}$$

This gives the last term of Equation 8.15. Now let us show how to get the first two terms in Equation 8.15. From Equation 8.16, 8.21 and 8.26, we can get

$$\begin{aligned}\ln p(\Delta \boldsymbol{y}_t|\beta_t, \boldsymbol{\theta}^t) &= \ln\left\{\left(\frac{\beta_t}{2\pi}\right)^{\frac{M}{2}}\prod_{n=1}^{N}\left(\frac{\theta_n^t}{2\pi}\right)^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\Delta \boldsymbol{y}_t{}'\boldsymbol{\Omega}^{-1}\Delta \boldsymbol{y}_t\right\}(2\pi)^{\frac{N}{2}}\left|\boldsymbol{\Sigma}_{\Delta x_t}\right|^{\frac{1}{2}}\right\} \\ &= \frac{1}{2}(-M\ln 2\pi\ + M\ln\beta_t\ + \ln|\mathrm{diag}(\boldsymbol{\theta}^t)| + \ln\left|\boldsymbol{\Sigma}_{\Delta x_t}\right|) \\ &-\frac{1}{2}\Delta \boldsymbol{y}_t{}'\boldsymbol{\Omega}^{-1}\Delta \boldsymbol{y}_t\end{aligned} \tag{8.27}$$

As we have

$$\left|\beta_t{}^{-1}\mathbf{I}\right||\beta_t \boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)| = |\mathrm{diag}(\boldsymbol{\theta}^t)|\left|\beta_t{}^{-1}\mathbf{I} + \boldsymbol{C}\mathrm{diag}(\boldsymbol{\theta}^t)^{-1}\boldsymbol{C}'\right| \tag{8.28}$$

which gives

$$\begin{aligned}\ln\left|\beta_t{}^{-1}\mathbf{I} + \boldsymbol{C}\mathrm{diag}(\boldsymbol{\theta}^t)^{-1}\boldsymbol{C}'\right| &= \ln\left|\beta_t{}^{-1}\mathbf{I}\right| + \ln|\beta_t \boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}^t)| - \ln|\mathrm{diag}(\boldsymbol{\theta}^t)| \\ &= -M\ln\beta_t\ - \ln\left|\boldsymbol{\Sigma}_{\Delta x_t}\right| - \ln|\mathrm{diag}(\boldsymbol{\theta}^t)|\end{aligned} \tag{8.29}$$

Equation 8.27 can be turned into

$$\ln p(\Delta \boldsymbol{y}_t|\beta_t, \boldsymbol{\theta}^t) = \frac{1}{2}(-M\ln 2\pi\ - \ln|\boldsymbol{\Omega}|) - \frac{1}{2}\Delta \boldsymbol{y}_t{}'\boldsymbol{\Omega}^{-1}\Delta \boldsymbol{y}_t \tag{8.30}$$

Therefore, $p(\Delta \boldsymbol{y}_t|\beta_t, \boldsymbol{\theta}^t)$ is of the following form:

$$\begin{aligned}p(\Delta \boldsymbol{y}_t|\beta_t, \boldsymbol{\theta}^t) = (2\pi)^{-M/2}|\beta_t{}^{-1}\mathbf{I} + \boldsymbol{C}\mathrm{diag}(\boldsymbol{\theta}^t)^{-1}\boldsymbol{C}'|^{-1/2}\exp\{-\frac{1}{2}\Delta \boldsymbol{y}_t{}'(\beta_t{}^{-1}\mathbf{I} \\ + \boldsymbol{C}\mathrm{diag}(\boldsymbol{\theta}^t)^{-1}\boldsymbol{C}')^{-1}\Delta \boldsymbol{y}_t\}\end{aligned} \tag{8.31}$$

From (Bishop & Tipping 2000), the closed form of β_t and $\boldsymbol{\theta}^t$ cannot be derived that expectation-maximisation based re-estimates are used. In the maximisation step, the goal is to find values of β_t and $\boldsymbol{\theta}^t$ that maximize Equation 8.31. They can be obtained by differentiating $\ln(\mathbb{E}(p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)))$ and setting to zero, where $\mathbb{E}(.)$ denotes expectation over $\Delta \boldsymbol{x}_t$. Use Equation 8.31, 8.30, 8.22, 8.18, 8.17, and the formula

$$\mathbb{E}(\mathbf{X}\mathbf{X}') = \boldsymbol{\Sigma} + \mathbb{E}(\mathbf{X})\mathbb{E}(\mathbf{X}) \tag{8.32}$$

the term in $\ln(\mathbb{E}(p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)))$ related to $\ln(\theta_n^t)$ is

$$\frac{1}{2}[\ln\theta_n^t - \theta_n^t((\mu_{\Delta x_t}^n)^2 + \boldsymbol{\Sigma}_{\Delta x_t}^{n,n})] \tag{8.33}$$

where $\mu_{\Delta x_t}^n$ is the nth element of $\boldsymbol{\mu}_{\Delta x_t}$, and $\boldsymbol{\Sigma}_{\Delta x_t}^{n,n}$ is the element of $\boldsymbol{\Sigma}_{\Delta x_t}$ in nth row and nth column.

Differentiate Equation 8.33 with respect to $\partial \ln(\theta_n^t)$ gives

$$\frac{\partial l \, \mathrm{n}\left(\mathbb{E}\left(\, p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)\right)\right)}{\partial \ln(\theta_n^t)} = \frac{1}{2}[1 - \theta_n^t((\mu_{\Delta \boldsymbol{x}_t}^n)^2 + \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}^{n,n})] \tag{8.34}$$

In (Bishop & Tipping 2000), following the definition of a quantity γ_n^t in (MacKay 1992):

$$\gamma_n^t = 1 - \theta_n^t \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}^{n,n} \tag{8.35}$$

Equation 8.34 becomes:

$$\frac{\partial l \, \mathrm{n}\left(\mathbb{E}\left(\, p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)\right)\right)}{\partial \ln(\theta_n^t)} = \frac{1}{2}[-\theta_n^t(\mu_{\Delta \boldsymbol{x}_t}^n)^2 + \gamma_n^t)] \tag{8.36}$$

γ_n^t can be interpreted as a measure of how 'well-determined' its corresponding parameter is by the data. By setting Equation 8.36 to be zero, we get the equation to iteratively update θ_n^t as

$$\theta_n^{t\,new} = \frac{\gamma_n^t}{(\mu_{\Delta x_t}^n)^2} \tag{8.37}$$

Similarly, we get the equation to update β_t by differentiating Equation 8.31 with respect to $\partial \ln(\beta_t)$. Using the Equation 8.29 and 8.17, the term in $\ln \mathbb{E}((\, p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)))$ related to $\ln(\beta_t)$ is

$$\frac{1}{2} M \ln \beta_t \; - \frac{\beta_t}{2} \|\Delta \boldsymbol{y}_t - \boldsymbol{C}\Delta \boldsymbol{\mu}_t\|^2 - \frac{\beta_t}{2} tr(\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t} \boldsymbol{C}'\boldsymbol{C}) \tag{8.38}$$

Differentiate Equation 8.38 with respect to $\partial \ln(\beta_t)$ gives

$$\frac{\partial \ln(\mathbb{E}\left(\, p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}^t, \beta_t)\right))}{\partial \ln(\beta_t)} = \frac{1}{2}(M - \beta_t \left\|\Delta \boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{\mu}_{\Delta \boldsymbol{x}_t}\right\|^2 - \beta_t tr(\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t} \boldsymbol{C}'\boldsymbol{C})) \tag{8.39}$$

$tr\left(\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t} \boldsymbol{C}'\boldsymbol{C}\right)$ can be written as ${\beta_t}^{-1} \sum_{n=1}^{N} \gamma_n^t$ according to the following transformation

$$\begin{aligned} \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}(\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t})^{-1} &= \boldsymbol{I} \\ \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}(\boldsymbol{C}'\boldsymbol{C} + {\beta_t}^{-1}\mathrm{diag}(\boldsymbol{\theta}^t)) &= {\beta_t}^{-1}\boldsymbol{I} \\ \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}\boldsymbol{C}'\boldsymbol{C} &= {\beta_t}^{-1}\boldsymbol{I} - \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}{\beta_t}^{-1}\mathrm{diag}(\boldsymbol{\theta}^t) \\ tr(\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}\boldsymbol{C}'\boldsymbol{C}) &= {\beta_t}^{-1}\sum_{n=1}^{N}(1 - \theta_n^t \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}_t}^{n,n}) \end{aligned} \tag{8.40}$$

By setting Equation 8.39 to zero, we can get

$$\beta_t^{new} = \frac{\left\|\Delta \boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{\mu}_{\Delta x_t}\right\|^2}{M - \sum_{n=1}^{N} \gamma_n^t} \tag{8.41}$$

In short, $\Delta \boldsymbol{x}_t$ can be obtained by iteratively calculating $\boldsymbol{\mu}_{\Delta x_t}$, $\boldsymbol{\Sigma}_{\Delta x_t}$, θ_n^t and β_t specified in Equation 8.13 and 8.37 and 8.41.

8.3.1.2 Group SBL: dependently infer $\Delta \boldsymbol{x}_t$ for all $\boldsymbol{t} \in \{\boldsymbol{2}, .., \boldsymbol{T}\}$

As SBL independently infer Δx_t, the value at neighbouring time points will not influence each other. It may happen that Δx_t^n is zero at most time points, but at one or two time points Δx_t^n is above zero due to noise or estimation error. This brings difficulties to decide whether the species n is time-variant or not. A wrong decision will be made if we occasionally obtain a significant non-zero value at one time point. Hence, we consider a way to avoid this situation by incorporating information from all time points. The Group SBL algorithm is proposed in this chapter to infer Δx_t for all $t \in \{2, .., T\}$ dependently. The main idea to let Δx_t for all $t \in \{2, .., T\}$ share the same prior distribution.

Let $\Delta \boldsymbol{X} = [\Delta \boldsymbol{x}_2, \dots, \Delta \boldsymbol{x}_T]$ and $\Delta \boldsymbol{Y} = [\Delta \boldsymbol{y}_2, \dots, \Delta \boldsymbol{y}_T]$. In SBL, following the definition in Equation 8.7, the prior distribution of $\Delta \boldsymbol{X}$ is $p(\Delta \boldsymbol{X}|\boldsymbol{\theta})$:

$$p(\Delta \boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=2}^{T} \prod_{n=1}^{N} (2\pi)^{-\frac{1}{2}} (\theta_n^t)^{\frac{1}{2}} \exp\left\{-\frac{\theta_n^t}{2} (\Delta x_t^n)^{\boldsymbol{2}}\right\} \tag{8.42}$$

In Group SBL, we let $\theta_n^t = \theta_n$ for all t. Then Equation 8.42 becomes:

$$p(\Delta \boldsymbol{X}|\boldsymbol{\theta}) = \prod_{t=2}^{T} \prod_{n=1}^{N} (2\pi)^{-\frac{1}{2}} (\theta_n)^{\frac{1}{2}} \exp\left\{-\frac{\theta_n}{2} (\Delta x_t^n)^{\boldsymbol{2}}\right\} \tag{8.43}$$

where $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_N\}$ is the parameter set shared by all time points. Following Equation 8.10, the posterior distribution of $\Delta \boldsymbol{X}$ over parameters is

$$p(\Delta \boldsymbol{X}|\Delta \boldsymbol{Y}, \boldsymbol{\theta}, \beta_t) = \prod_{t=2}^{T} N\left(\boldsymbol{\mu}_{\Delta x_t}, \boldsymbol{\Sigma}_{\Delta x_t}\right) \tag{8.44}$$

where $\boldsymbol{\mu}_{\Delta x_t}$ and $\boldsymbol{\Sigma}_{\Delta x_t}$ remains the same as in Equation 8.13. The marginal likelihood $p(\Delta \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta})$ is

$$p(\Delta \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_{t=2}^{T} p(\Delta \boldsymbol{y}_t|\boldsymbol{\theta}, \beta_t) \tag{8.45}$$

where $\boldsymbol{\beta} = \{\beta_2, \dots, \beta_T\}$.

To obtain the formula to update $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, we follow the idea of SBL to choose the optimal parameter by maximizing $p(\Delta \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta})$ via an expectation-maximisation manner. Using Equation 8.45 and 8.36, differentiating $\ln(p(\Delta \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta}))$ with respect to θ_n will give as

$$\frac{\partial l \, \mathrm{n}\left(\mathbb{E}\left(\, p(\Delta \boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\beta})\right)\right)}{\partial \ln(\theta_n)} = \frac{1}{2}\sum_{t=2}^{T}[-\theta_n\left(\mu_{\Delta x_t}^{n}\right)^2 + \gamma_n^t)] \tag{8.46}$$

Therefore, the formula to update θ_n is

$$\theta_n{}^{new} = \frac{\sum_{t=2}^{T}\gamma_n^t}{\sum_{t=2}^{T}(\mu_{\Delta x_t}^{n})^2} \tag{8.47}$$

where γ_n^t follows the same definition in Equation 8.35. The formula for updating β_t remains the same as it is in Equation 8.41.

The whole process of the Group SBL method is summarized in Algorithm 8.1.

Algorithm 8.1	The Group SBL algorithm
Input	Changes in observations $\Delta \boldsymbol{Y} = [\Delta \boldsymbol{y}_2, \dots, \Delta \boldsymbol{y}_T]$ and measurement matrix $\boldsymbol{C}$
Output	The distribution of $\Delta \boldsymbol{X} = [\Delta \boldsymbol{x}_2, \dots, \Delta \boldsymbol{x}_T]$
1	N ← number of species
2	Initialize $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_N\}$ and $\boldsymbol{\beta} = \{\beta_t, \dots, \beta_T\}$
3	**do**
4	**for** $t \leftarrow 2$ **to** T
5	$\boldsymbol{\Sigma}_{\Delta x_t} \leftarrow (\beta_t \boldsymbol{C}'\boldsymbol{C} + \mathrm{diag}(\boldsymbol{\theta}))^{-1}$
6	$\boldsymbol{\mu}_{\Delta x_t} \leftarrow \beta_t \boldsymbol{\Sigma}_{\Delta x_t} \boldsymbol{C}' \Delta \boldsymbol{y}_t$
7	**end**
8	**for** $n \leftarrow 1$ **to** N
9	**for** $t \leftarrow 2$ **to** T
10	$\gamma_n^t \leftarrow 1 - \theta_n \boldsymbol{\Sigma}_{\Delta x_t}^{n,n}$
11	**end**
12	$\theta_n{}^{new} \leftarrow \frac{\sum_{t=2}^{T}\gamma_n^t}{\sum_{t=2}^{T}(\mu_{\Delta x_t}^{n})^2}$
13	**end**
14	**for** $t \leftarrow 2$ **to** T
15	$\beta_t{}^{new} \leftarrow \frac{\left\|\Delta \boldsymbol{y}_t - \boldsymbol{C}\boldsymbol{\mu}_{\Delta x_t}\right\|^2}{M - \sum_{n=1}^{N}\gamma_n^t}$

16		**end**
17	**while**	$\boldsymbol{\mu}_{\Delta x_t}$ is not convergent

8.3.2 Finding active species whose concentrations vary over time

Having obtained the distribution of variations of species $\Delta \boldsymbol{x}_t$, the next step is to determine active species. The species whose mean variations in concentration over time is larger than a predefined threshold ε_x will be recorded as an active species; otherwise, it will be regarded as an inactive species since its concentration over time remains almost constant. That is, we define an active species whose concentration satisfies:

$$\frac{1}{\mathrm{T}-1}\sum_{t=2}^{T}|\mu_{\Delta x_t^n}| > \varepsilon_x \tag{8.48}$$

where ε_x is a small value. Let vector $\boldsymbol{S}^{active}$ record the index of active species. The choice of ε_x will greatly influence the selection of active species. As it will be shown in section 8.4, we choose the value of ε_x by the standard deviation of estimated measurement noise.

8.3.3 Inferring transition matrix for active species

Let $\widehat{\boldsymbol{x}}_t$ denote the concentrations of active species at time t. The total number of active species in the above step is $\widehat{N}$. Assume the transition matrix for active species is $\widehat{\boldsymbol{A}}$, which is a $\widehat{N} \times \widehat{N}$ matrix. We have

$$\Delta\widehat{\boldsymbol{x}}_t = \widehat{\boldsymbol{A}}\Delta\widehat{\boldsymbol{x}}_{t-1} + N(0,2\sigma_s^2\boldsymbol{I}) \tag{8.49}$$

for $t \in \{3,..,T\}$. For simplicity, let $\delta_t^{-1} = 2\sigma_s^2$ for the following notations.

Let $\Delta\widehat{\boldsymbol{x}}_t^j$ be the jth element of $\Delta\widehat{\boldsymbol{x}}_t$, and $\widehat{\boldsymbol{a}}'_j$ be the jth row of $\widehat{\boldsymbol{A}}$. From Equation 8.49, we can obtain:

$$\begin{aligned}\Delta\widehat{\boldsymbol{x}}_t^j &= \widehat{\boldsymbol{a}}'_j\Delta\widehat{\boldsymbol{x}}_{t-1} + N(0,\delta_t^{-1}) \\ &= \Delta\widehat{\boldsymbol{x}}'_{t-1}\widehat{\boldsymbol{a}}_j + N(0,\delta_t^{-1})\end{aligned} \tag{8.50}$$

Augmenting Equation 8.50 for all $t \in \{3,..,T\}$, we can get

$$\left[\Delta\widehat{\boldsymbol{x}}_3^j, \Delta\widehat{\boldsymbol{x}}_4^j, \dots, \Delta\widehat{\boldsymbol{x}}_T^j\right]' = [\Delta\widehat{\boldsymbol{x}}_2, \Delta\widehat{\boldsymbol{x}}_3, \dots, \Delta\widehat{\boldsymbol{x}}_{T-1}]'\widehat{\boldsymbol{a}}_j + N(0, \mathrm{diag}(\boldsymbol{\delta}_j^{-1})) \tag{8.51}$$

where $\mathrm{diag}(\boldsymbol{\delta}_j^{-1})$ is a matrix whose diagonal elements are set by the vector $\boldsymbol{\delta}_j^{-1}$. Let us assume $\boldsymbol{\delta}_j^{-1} = \delta_j^{-1}\boldsymbol{I}$, where δ_j is the inverse variance of noise for the jth row. The transpose of the jth row $\widehat{\boldsymbol{a}}_j$ is the signal to be inferred. The matrix $[\Delta\widehat{\boldsymbol{x}}_2, \Delta\widehat{\boldsymbol{x}}_3, \dots, \Delta\widehat{\boldsymbol{x}}_{T-1}]'$and column

vector $\left[\Delta\hat{\boldsymbol{x}}_3^j, \Delta\hat{\boldsymbol{x}}_4^j, \ldots, \Delta\hat{\boldsymbol{x}}_T^j\right]'$ can be viewed as a measurement matrix and the corresponding observation vector respectively. According to a widely accepted assumption that structures of biological networks are usually sparse (Hendriks et al. 2008)(August & Papachristodoulou 2009)(Yeung et al. 2002), SBL can be directly used to solve Equation 8.51.

We define a Gaussian prior to each row of $\widehat{\boldsymbol{A}}$ as:

$$p\left(\hat{\boldsymbol{a}}_j|\boldsymbol{\eta}^j\right) = \prod_{n=1}^{\hat{N}}(2\pi)^{-\frac{1}{2}}(\eta_n^j)^{\frac{1}{2}}\exp\{-\frac{\eta_n^j}{2}(\hat{a}_j^n)^{\mathbf{2}}\} \tag{8.52}$$

where $\boldsymbol{\eta}^j = \{\eta_1^j, \ldots, \eta_{\hat{N}}^j\}$ and $\hat{a}_j^n$ is the nth element of $\hat{\boldsymbol{a}}_j$.

Similar with Equation 8.13, the posterior distribution of $\hat{\boldsymbol{a}}_j$ over parameters is

$$p\left(\hat{\boldsymbol{a}}_j|\Delta\hat{\boldsymbol{x}}_t, \boldsymbol{\eta}^j, \delta_j\right) = N(\boldsymbol{\mu}_{\hat{\boldsymbol{a}}_j}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j}) \tag{8.53}$$

where

$$\begin{aligned}\boldsymbol{\mu}_{\hat{\boldsymbol{a}}_j} &= \delta_j\boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j}\sum_{t=2}^{T-1}\boldsymbol{\mu}_{\Delta\hat{\boldsymbol{x}}_t}\mu_{\Delta\hat{\boldsymbol{x}}_{t+1}^j} \\ \boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j} &= (\delta_j\sum_{t=2}^{T-1}\boldsymbol{\mu}_{\Delta\hat{\boldsymbol{x}}_t}\boldsymbol{\mu}_{\Delta\hat{\boldsymbol{x}}_t}{}' + \mathrm{diag}(\boldsymbol{\eta}^j))^{-\mathbf{1}}\end{aligned} \tag{8.54}$$

Let $\mu_{\hat{\boldsymbol{a}}_j}^n$ be the nth element of $\boldsymbol{\mu}_{\hat{\boldsymbol{a}}_j}$, and $\boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j}^{n,n}$ be the element of $\boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j}$ in nth row and column. $\boldsymbol{\mu}_{\Delta\hat{\boldsymbol{x}}_t}$ is the mean of $\Delta\hat{\boldsymbol{x}}_t$ obtained from the previous step.

Same as Equation 8.37 and 8.41, the formula to update η_n^j and δ_j are:

$$\eta_n^{j\,new} = \frac{\gamma_n^j}{(\mu_{\hat{\boldsymbol{a}}_j}^n)^2} \tag{8.55}$$

where

$$\gamma_n^j = 1 - \eta_n^j\boldsymbol{\Sigma}_{\hat{\boldsymbol{a}}_j}^{n,n} \tag{8.56}$$

and

$$\delta_j{}^{new} = \frac{\sum_{t=2}^{T-1}(\mu_{\Delta\hat{x}_{t+1}^j} - \boldsymbol{\mu}_{\Delta\hat{\boldsymbol{x}}_t}{}'\boldsymbol{\mu}_{\hat{\boldsymbol{a}}_j})^2}{(T-2) - \sum_{n=1}^{\hat{N}}\gamma_n^j} \tag{8.57}$$

The whole process of the SBL algorithm to infer $\hat{\boldsymbol{a}}_j$ is summarized in Algorithm 8.2.

Algorithm 8.2	The SBL algorithm to infer $\widehat{\boldsymbol{a}}_j$
Input	The distribution of $\Delta\widehat{\boldsymbol{x}}_t$ for $t \in \{2, .., T\}$
Output	The distribution of $\widehat{\boldsymbol{a}}_j$
1	$\widehat{N} \leftarrow$ number of active species
2	Initialize $\boldsymbol{\eta}^j = \{\eta_1^j, ..., \eta_{\widehat{N}}^j\}$ and δ_j
3	**do**
4	$\boldsymbol{\Sigma}_{\widehat{\boldsymbol{a}}_j} = (\delta_j \sum_{t=2}^{T-1} \boldsymbol{\mu}_{\Delta\widehat{\boldsymbol{x}}_t} \boldsymbol{\mu}_{\Delta\widehat{\boldsymbol{x}}_t}{}' + \mathrm{diag}(\boldsymbol{\eta}^j))^{-\mathbf{1}}$
5	$\boldsymbol{\mu}_{\widehat{\boldsymbol{a}}_j} = \delta_j \boldsymbol{\Sigma}_{\widehat{\boldsymbol{a}}_j} \sum_{t=2}^{T-1} \boldsymbol{\mu}_{\Delta\widehat{\boldsymbol{x}}_t} \mu_{\Delta\widehat{\boldsymbol{x}}_{t+1}^j}$
6	**for** $n \leftarrow 1$ **to** $\widehat{N}$
7	$\gamma_n^j = 1 - \eta_n^j \boldsymbol{\Sigma}_{\widehat{\boldsymbol{a}}_j}^{n,n}$
8	$\eta_n^{j\,new} = \dfrac{\gamma_n^j}{(\mu_{\widehat{\boldsymbol{a}}_j}^n)^2}$
9	**end**
10	$\delta_j^{new} = \dfrac{\sum_{t=2}^{T-1} (\mu_{\Delta\widehat{x}_{t+1}^j} - \boldsymbol{\mu}_{\Delta\widehat{\boldsymbol{x}}_t}{}' \boldsymbol{\mu}_{\widehat{\boldsymbol{a}}_j})^2}{(T-2) - \sum_{n=1}^{\widehat{N}} \gamma_n^j}$
11	**while** $\boldsymbol{\mu}_{\widehat{\boldsymbol{a}}_j}$ is not convergent

8.3.4 Obtaining the complete transition matrix

As the vector $\boldsymbol{S}^{active}$ specifies the index of active species, we can use it to derive the complete transition matrix $\boldsymbol{A}$ from $\widehat{\boldsymbol{A}}$. We can first initialize $\boldsymbol{A}$ to be an identity matrix $\boldsymbol{I}_{N\times N}$, which means that by default all species are inactive. Then we set elements of $\boldsymbol{A}$ by

$$a_{j'}^{n'} = \hat{a}_j^n \text{ for all } n \in \{1, \widehat{N}\} \text{ and } j \in \{1, \widehat{N}\} \tag{8.58}$$

where $\hat{a}_j^n$ is the element of $\widehat{\boldsymbol{A}}$ in the jth row and nth column, while $a_{j'}^{n'}$ is the element of $\boldsymbol{A}$ in the j'th row and n'th column. j' and n' are obtained from

$$\begin{aligned} j' &= S_j^{active} \\ n' &= S_n^{active} \end{aligned} \tag{8.59}$$

where S_j^{active} and S_n^{active} are the jth and nth elements of $\boldsymbol{S}^{active}$. The resulting transition matrix $\boldsymbol{A}$ reflects the dynamics of the whole network, where some species can be inactive while some others can be active.

8.3.5 Perturbation

The above steps infer transition matrix using measurements $\boldsymbol{y}_t$ under one perturbation. It can be easily extended to any number of perturbations by successively combining all measurements together. Suppose there are a number of P perturbations. Then we have the observation matrix as:

$$\overline{\Delta \boldsymbol{Y}} = [\Delta \boldsymbol{Y}^1, \dots, \Delta \boldsymbol{Y}^P] \tag{8.60}$$

where $\Delta \boldsymbol{Y}^p$ contains observations from the pth perturbation, and $\Delta \boldsymbol{Y}^p = \left[\Delta \boldsymbol{y}_2^p, \dots, \Delta \boldsymbol{y}_T^p\right]$. Then the matrix for the variations of species is

$$\overline{\Delta \boldsymbol{X}} = [\Delta \boldsymbol{X}^1, \dots, \Delta \boldsymbol{X}^P] \tag{8.61}$$

where $\Delta \boldsymbol{X}^p = \left[\Delta \boldsymbol{x}_2^p, \dots, \Delta \boldsymbol{x}_T^p\right]$ comprises of concentrations under the pth perturbation. Let $K = (T-1) * P$, then the dimensions of $\overline{\Delta \boldsymbol{Y}}$ and $\overline{\Delta \boldsymbol{X}}$ are $M \times K$ and $N \times K$, respectively. The kth columns of $\overline{\Delta \boldsymbol{Y}}$ and $\overline{\Delta \boldsymbol{X}}$ are denoted by vectors $\Delta \boldsymbol{y}_k$ and $\Delta \boldsymbol{x}_k$, respectively. Then the whole procedure of CCELL for transition matrix inference under P perturbations can be described as Algorithm 8.3.

Algorithm 8.3	The CCELL algorithm
Input	Changes in observations over perturbations $\overline{\Delta \boldsymbol{Y}}$ and measurement matrix $\boldsymbol{C}$
Output	Transition matrix $\boldsymbol{A}$
1	N ← number of species
2	K ← number of columns in $\overline{\Delta \boldsymbol{Y}}$
3	$\{(\boldsymbol{\mu}_{\Delta x_k}, \boldsymbol{\Sigma}_{\Delta x_k}) \mid k \in \{1, K\}$ ← GroupSBL($\overline{\Delta \boldsymbol{Y}}, \boldsymbol{C}$) //call Group SBL algorithm
4	**for** n ←1 **to** N // Find active species
5	**if** $\frac{1}{\mathrm{K}} \sum_{k=1}^{K} \lvert \mu_{\Delta x_k^n} \rvert > \varepsilon_x$
6	save n to $\boldsymbol{S}^{active}$
4	**end**
7	**end**
8	$\boldsymbol{\mu}_{\Delta \hat{\boldsymbol{x}}_k} \leftarrow \boldsymbol{\mu}_{\Delta x_k}(\boldsymbol{S}^{active})$ for $k \in \{1, K\}$// get the expectations of $\Delta \hat{\boldsymbol{x}}_k$ for active species
9	$\widehat{N}$ ← length($\boldsymbol{S}^{active}$) // number of active species
10	**for** n ←1 **to** $\widehat{N}$
11	$\boldsymbol{\mu}_{\hat{\boldsymbol{a}}_n} \leftarrow$ SBL($\{\boldsymbol{\mu}_{\Delta \hat{\boldsymbol{x}}_k}\}_{k=1}^{K}, n$) // estimate each row of $\widehat{\boldsymbol{A}}$

12 **end**

13 Initialize $\boldsymbol{A} \leftarrow \boldsymbol{I}_{N\times N}$

14 **for** $j \leftarrow 1$ **to** $\widehat{N}$ //each row

15 **for** $n \leftarrow 1$ **to** $\widehat{N}$ //each column

16 $j' \leftarrow \boldsymbol{S}^{active}(j)$

17 $n' \leftarrow \boldsymbol{S}^{active}(n)$

18 $\boldsymbol{A}(j', n') \leftarrow \widehat{\boldsymbol{A}}(j, n)$

19 **end**

20 **end**

8.4 Experiment and results

To demonstrate our method, we introduce designed large-scale networks. Inheriting from the feature of time sparsity, only a small part of the network shows variation within a specific time scope. We would like to apply our method to find those time-variant species and infer the dynamic model of the time-variant part of the large-scale network. In our experiment, we also investigate the performance of our method when different numbers of species in the network have dynamic changes. That is, the number of elements in $\Delta\boldsymbol{x}_t$ is different, and hence the sparsity of the transition matrix $\boldsymbol{A}$ varies. The performance of sparse learning method is therefore investigated for inferring matrices of different sparsity.

In the following part, we will first introduce the settings of the experiment in detail. Then the performance of our method on $\Delta\boldsymbol{X}$ and $\boldsymbol{A}$ inference will be presented. Root-mean-square deviation (RMSE), sensitivity, specificity and the area under the receiver operating characteristic (ROC) curve (AUROC) are used to evaluate the performance. RMSE is used to measure the standard deviations between the predicted and real values. The RMSE value for $\Delta\boldsymbol{X}$ inference is calculated by

$$RMSE(\Delta\boldsymbol{X}, \Delta\boldsymbol{X}^{est}) = \sqrt{\frac{\sum_{t=2}^{T}\sum_{n=1}^{N}[\Delta x_t^n - (\Delta x_t^n)^{est}]^2}{(T-1)*P*N}} \tag{8.62}$$

where $\Delta\boldsymbol{X}^{est}$ is the inferred results, T is the number of time points, N is the number of species and P is the number of perturbations. Similarly, the RMSE value for $\boldsymbol{A}$ inference is

$$RMSE(\boldsymbol{A}, \boldsymbol{A}^{est}) = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}[a_{j,i} - (a_{j,i})^{est}]^2}{N*N}} \tag{8.63}$$

where $a_{j,i}$ is the element of $\boldsymbol{A}$ in jth row and ith column, and $\boldsymbol{A}^{est}$ is the inferred results.

Along with RMSE, we also use sensitivity and specificity. As we have mentioned in Chapter 3, sensitivity measures the ratio of real positives that are correctly inferred, while specificity measures the ratio of correctly identifying real negatives. To evaluate $\Delta\boldsymbol{X}^{est}$, we check whether we have correctly discovered time-variant species. Let real time-variant species be positive. With a specific value of threshold ε_x, we can find a set of active species satisfying Equation 8.48. Then sensitivity and specificity can be calculated by comparing the real time-variant species and inferred active species. By varying ε_x, we will get a set of sensitivity and specificity pairs. Similarly, we can check whether we have inferred the real structure of the network. Let the non-zero entry $a_{j,i}$ in $\boldsymbol{A}$ be real positive, representing a link between jth and ith species in the network. With a specific threshold ε_a, we can set the (j,i)th element in $\boldsymbol{A}^{est}$ to be an inferred positive if

$$\left(a_{j,i}\right)^{est} > \varepsilon_a \tag{8.64}$$

ε_a can be regarded as a sparsity parameter, such that fewer elements in the $\boldsymbol{A}$ matrix will be regarded as present with a larger value of ε_a. Comparing real positives and inferred positives, we will get the sensitivity of detecting real network structure. Moreover, ROC curve will be shown by plotting real positive rates against false positive rates at various ε_a. AUROC measures the area under the ROC curve, representing the ability of detecting real network structure using a single value.

8.4.1 Experimental settings

First, we design a set of benchmark large-scale networks that we would like to infer in our experiment. All networks are supposed to contain 300 species in total, while the number of time-variant species in each network varies. That is, the dimension of transition matrix $\boldsymbol{A}$ for each network is 300×300, but the number of non-zero entries of $\boldsymbol{A}$ is different. For a time-invariant species j, the corresponding elements in the jth row of $\boldsymbol{A}$ are all zero except the jth element is set to 1; for the time-variant species j, the elements in the jth row of $\boldsymbol{A}$ have more than one non-zero values, showing that it participates in some reactions with other species. The elements in matrix $\boldsymbol{A}$ range from 0 to 1. To mimic the behaviour of biological networks, the transition matrix $\widehat{\boldsymbol{A}}$ for time-variant species is designed by taking the structure of 4-well studied signalling pathways: JAK-STAT (Swameye et al. 2003), GR (Necela & Cidlowski 2004), ERK (Kolch 2000), p38 (Hendriks et al. 2008), and the crosstalk amongst them (Shuto

et al. 2002; Wang et al. 2006; Necela & Cidlowski 2004). Table 8-1 lists the details of benchmark networks. There are 6 networks, where the amount of time-variant species ranges from 4 to 58 and the number of links between these time-variant species varies from 4 to 101.

Table 8-1 Details of 6 benchmark networks

ID	Components	# time-variant species	# links
n-4	JAK-STAT	4	4
n-11	ERK	11	20
n-39	p38	39	61
n-50	ERK and p38	50	83
n-53	GR, ERK and p38	53	93
n-58	GR, JAK-STAT, ERK and p38	58	101

Having defined matrix $\boldsymbol{A}$, the next step is to generate the concentrations of species, $\boldsymbol{x}_t$. It is interesting to investigate the performance of our method under different numbers of perturbations. In our experiment, the number of perturbations P varies from 2 to 8 as these values are frequently used in wet-lab experiments. In each perturbation, the time-series measurement is carried out at 10 sequential time points, that is $T = 10$. We randomly generate the initial concentration of all 300 species by sampling from a normal distribution $N(100, 30^2)$ under each perturbation. Thus in total, we generate P different initial concentrations and each one is used to generate $\boldsymbol{x}_t$ according to Equation 8.1.

After $\boldsymbol{x}_t$ has been generated, the following step is to generate measurement matrix $\boldsymbol{C}$. For all networks, whose total numbers of species are identical to 300, 150 combined measurements will be carried out at each discrete time point. Thus the dimension of the matrix $\boldsymbol{C}$ is 150×300. We use the same measurement matrix for all 6 benchmark network. It requires that the measurement should cover all species that each species will be combined measured at least once. As low-density parity-check code (LDPC) (Gallager 1962) is frequently used for the measurement matrix construction (Sarvotham et al. 2006), here we use it to generate the matrix $\boldsymbol{C}$. With defined $\boldsymbol{C}$ and $\boldsymbol{x}_t$, the time-series measurement $\boldsymbol{y}_t$ can be generated according to Equation 8.2. Our experiment only focuses on investigating the performance of our method when system noise and measurement noise maintain small. The standard deviation of measurement noise to generate $\boldsymbol{y}_t$ is 0.01.

We will use the above settings to carry out the experiment. With the known measurement matrix $\boldsymbol{C}$ and combined measurement $\boldsymbol{y}_t$, we will show the results of inferring $\Delta\boldsymbol{x}_t$ and $\boldsymbol{A}$. In the last part of results section, we will brief discuss the influence of introducing different levels of measurement noise, where the standard deviation varies from 0.01 to 5. Moreover, the effect of different number of measurement time points under each perturbation will also be investigated.

8.4.2 Results

8.4.2.1 $\Delta\boldsymbol{x}_t$ inference results

Our method is carried out in a two-step manner. The first step is inferring the time-series changes of concentrations for all species, $\Delta\boldsymbol{x}_t$. By comparing the difference between the inferred and real $\Delta\boldsymbol{x}_t$, we can get the performance of our method in the first step. Figure 8-4 depicts the RMSE values of $\Delta\boldsymbol{x}_t$ for 6 benchmark networks under different numbers of perturbations. It also compares the results of Group SBL against traditional SBL.

From Figure 8-4, we can observe that the RMSE values obtained from Group SBL are much lower than those values obtained using SBL. Using Group SBL, the mean RMSE for all networks under all different perturbations is as low as 0.019; while the mean RMSE achieved by SBL is 0.48. It is also clear that for networks containing more time-variant species, the RMSE value will be larger. Thus the performance is influenced by the degree of sparsity. Another interesting observation is that the RMSE value seems to be only slightly influenced by the number of perturbations. This is because, particularly for SBL, the time-series changes $\Delta\boldsymbol{x}_t$ at current time point t is largely determined by $\Delta\boldsymbol{y}_t$. For Group SBL, different number of perturbations may have more effect on the results, since the hyperparameter of $\Delta\boldsymbol{x}_t$ is shared across different time points. We can see from Figure 8-4, RMSE values for the n-58 network are higher than other networks using both methods. Under different number of perturbations, RMSE from the SBL method still remains high. However, when the number of perturbations increases above 3, RMSE achieved by Group SBL drops dramatically. Thus, the proposed method, Group SBL, can more accurately infer $\Delta\boldsymbol{x}_t$ when more perturbations are used.

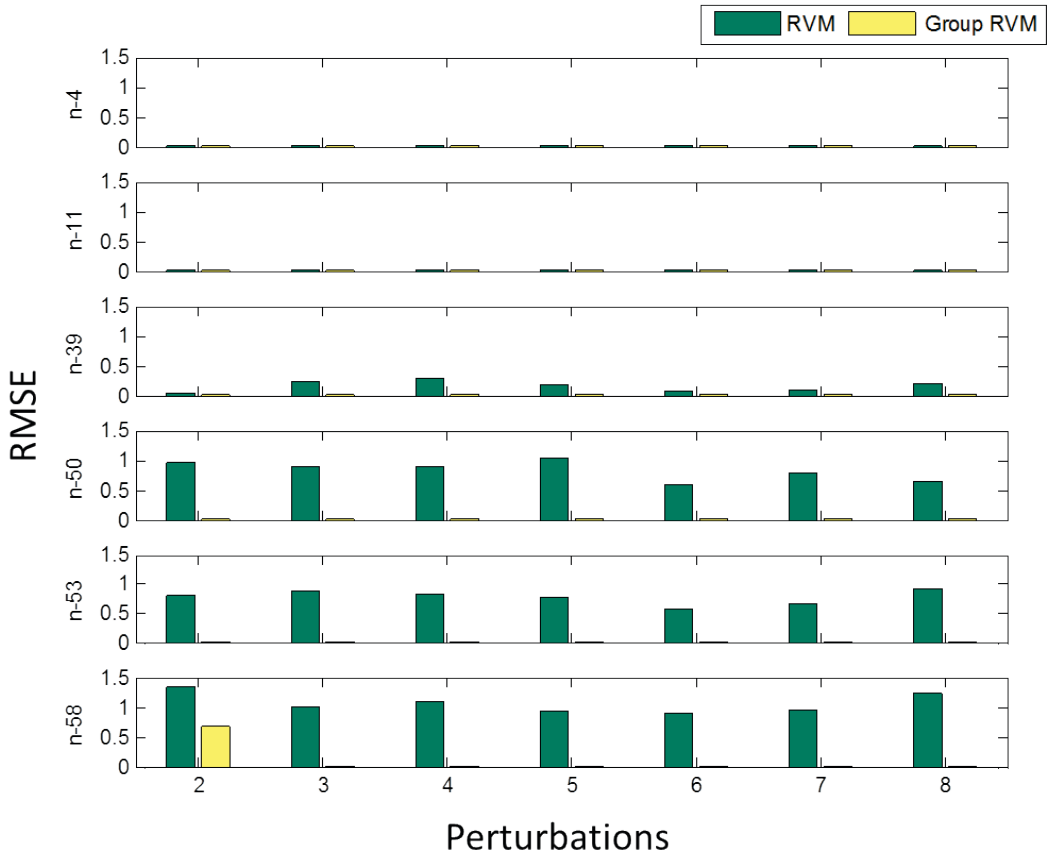


Figure 8-4 RMSE of $\Delta \boldsymbol{x}_t$ for all 6 benchmark networks under different number of perturbations using Group SBL and SBL. The Mean RMSE value of SBL is 0.48. For Group SBL the mean RMSE value is 0.019.

In order to further demonstrate that Group SBL is more robust than SBL, we plot the boxplot of estimation error for all Δx_t^n, where $t \in \{2, .., T\}$ and $n \in \{1, .., N\}$. Figure 8-5 and Figure 8-6 depict boxplots of estimation errors of Δx_t for all 6 benchmark networks under different perturbations using Group SBL and SBL respectively. Consistent with the observation in Figure 8-4, Group SBL gives smaller estimation error. Moreover, Group SBL shows stronger stability than SBL that the variation of errors is much smaller. For example, in Figure 8-5 the mean value of the 3rd quartile range of all boxplots is 0.0028, while in Figure 8-6 the mean value of the 3rd quartile range of all boxplots is 0.016. Therefore, we can conclude that from Figure 8-4, Figure 8-5 and Figure 8-6 Group SBL has better performance on inferring time-series changes of species in the aspect of estimation error. The error maintains at a very low level, where the mean of RMSE is close to 0.

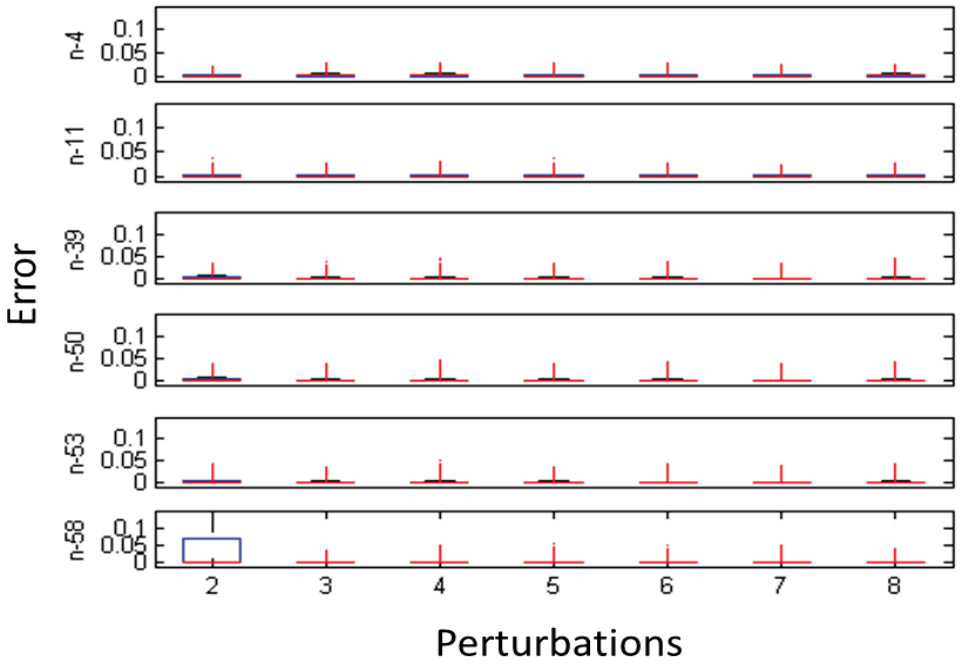


Figure 8-5 Boxplots of estimation errors of Δx_t for all 6 benchmark networks under different perturbations using Group SBL. The mean value of the 3rd quartile range of all boxplots is 0.0028.

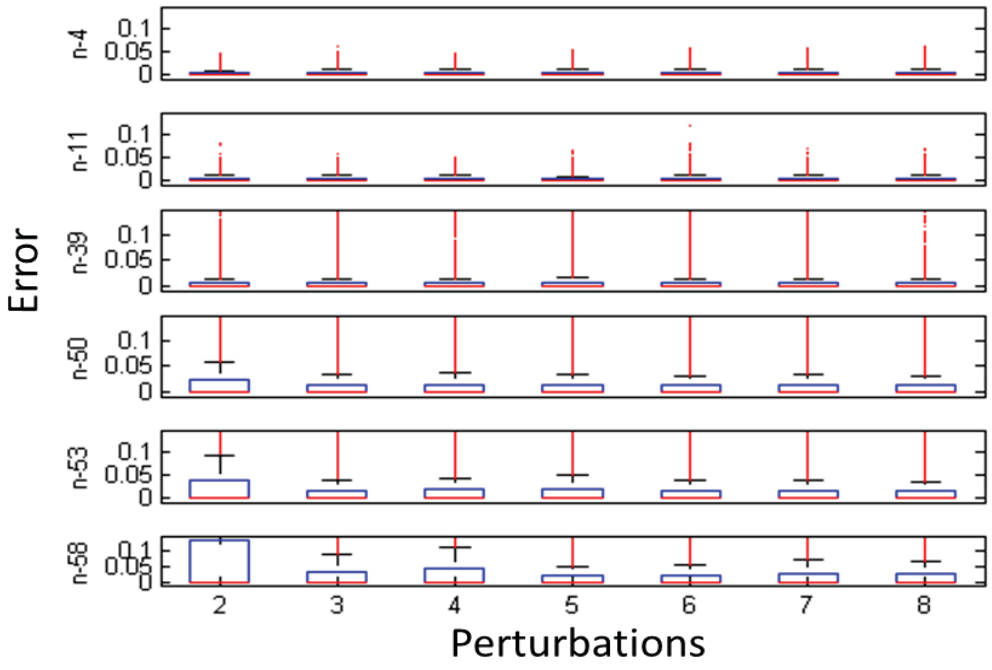


Figure 8-6 Boxplots of estimation errors of Δx_t for all 6 benchmark networks under different perturbations using SBL. The mean value of the 3rd quartile range of all boxplots is 0.016.

We can further demonstrate the robustness of Group SBL in another aspect. As we have mentioned previously, the main difference between Group SBL and SBL is that Group SBL forces the hyperparameter of Δx_t^n for the nth species to be identical across different discrete time points $t \in \{2, .., T\}$. Let us show an example to demonstrate the benefit of this difference. Take the inferred Δx_t for the n-58 network under 3 perturbations as an example. Figure 8-7 displays the image of Δx_t obtained from Group SBL; and Figure 8-8 shows the results from SBL. The colour of each pixel in the image is determined by the value of each element in Δx_t for $t \in \{2, .., T\}$. For a larger value, the corresponding pixel is darker; and for zero value, the corresponding pixel is white. In the n-58 network, the first 58 species are time-variant and the rest does not change. From Figure 8-7 and Figure 8-8, we can observe that the first 58 rows (representing the first 58 species) have a lot of dark pixels. By comparing Figure 8-7 with Figure 8-8, we can find that there are more dark pixels in other rows of Figure 8-8.This is because, the traditional SBL method independently infer $\Delta \boldsymbol{x}_t$ at different time point. At some time point, some elements in $\Delta \boldsymbol{x}_t$ maybe by chance become non-zero but at other time points these elements almost remain zero. In contrast, Group SBL infer $\Delta \boldsymbol{x}_t$ across all time points together. Thus, for a species, if most time it remains time-invariant and only has very tiny changes at some points from SBL, the time changes across all time points will be close to zero from Group SBL. To summary, Group SBL works better than SBL in inferring $\Delta \boldsymbol{x}_t$.

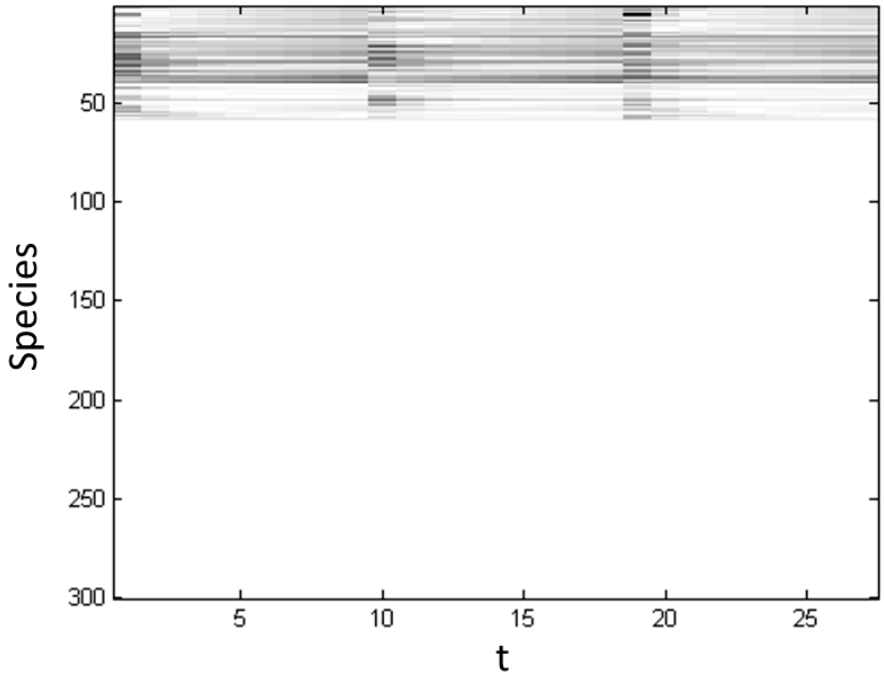


Figure 8-7 Image of $\Delta \boldsymbol{x}_t$ for the p58 network under 3 perturbations from Group SBL

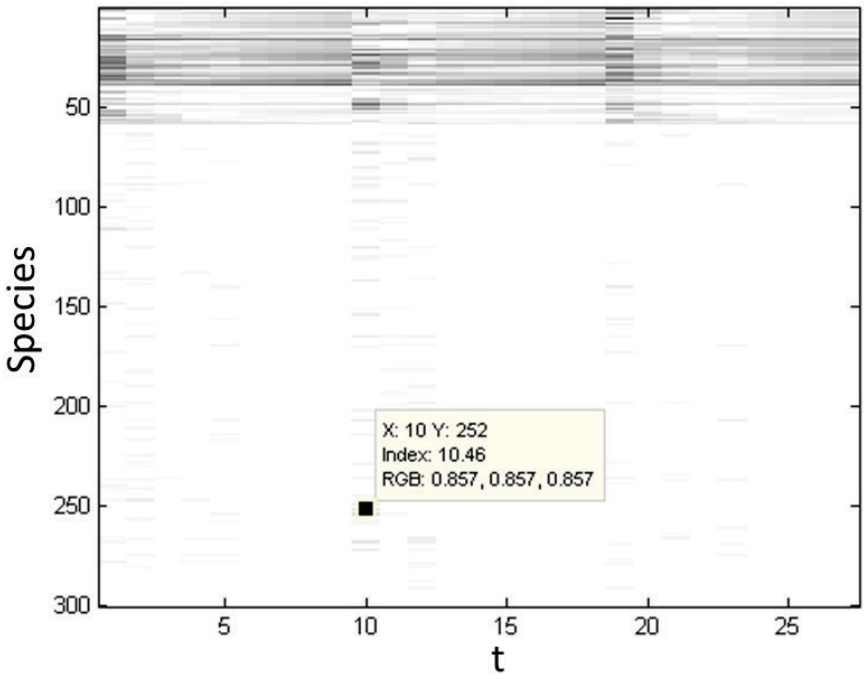


Figure 8-8 Image of $\Delta \boldsymbol{x}_t$ for the p58 network under 3 perturbations from SBL. There are also some dark pixels in the rows from 59 to 300. For example, there is a dark pixel in the 252nd row and the 10th column. It means that the time change of concentration for the 252nd species at discrete time point 10 is non-zero.

Having inferred $\Delta \boldsymbol{x}_t$, the next step is to determine the time-variant species. According to Equation 8.48, species are active when the mean changes over time are larger than the threshold ε_x. The choice of ε_x will influence the detection. Therefore, we now investigate a proper value of ε_x. As we have mentioned, real time-variant species are assumed to be positives. By varying the value of ε_x, we can get different values of sensitivity and specificity as it is shown in Figure 8-9. We can see that when the threshold varies from 0 to 0.04, the specificity values for all networks under all perturbations remain almost 1. It means that all are detected negatives are real time-invariant species. By looking at the subplots in the second row of Figure 8-9, we can observe that there is a sharp jump of the sensitivity values occurring around the threshold value equalling to 0.01. Thus, if $\varepsilon_x > 0.01$, then almost all time-invariant species will not be mis-classified as positives. This value exactly matches the standard deviation of artificial measurement noise. Therefore, if our method is applied for real network, the threshold ε_x is determined by estimating the noise in measurements. In our experiment, we choose $\varepsilon_x = 0.01$ to complete the following network inference task.

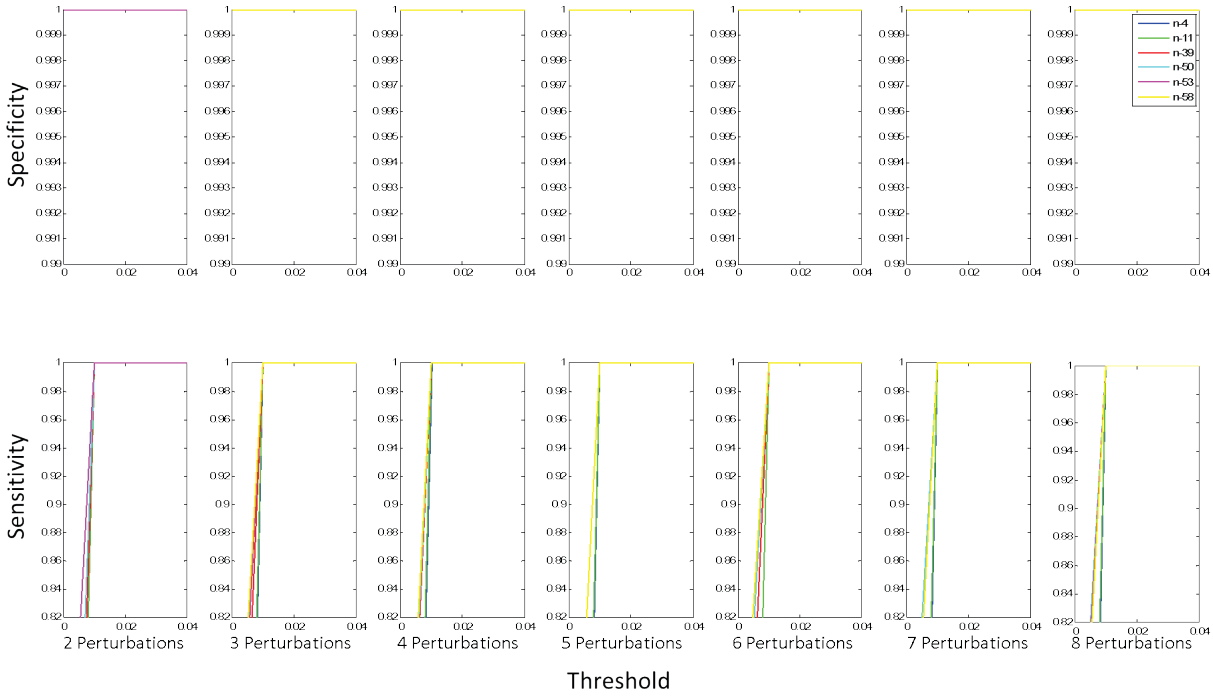


Figure 8-9 The specificity and sensitivity of detecting real time-variant species by changing the value of threshold ε_x. The threshold varies from 0 to 10 with step size of 0.01. In the figure, we only show the results where ε_x is within the range between 0 and 0.04.

8.4.2.2 Transition matrix inference

The second step of our method is using the time changes of active species $\Delta\widehat{\boldsymbol{x}}_t$ obtained from the Group SBL method to infer the transition matrix $\widehat{\boldsymbol{A}}$ for active species. Then from $\widehat{\boldsymbol{A}}$ we will get the complete transition matrix $\boldsymbol{A}$ for the whole network. Figure 8-10 illustrates the RMSE of transition matrix $\boldsymbol{A}$ for all 6 networks under different numbers of perturbations. Similar with the observations from Figure 8-4, the RMSE values are correlated with the sparse of variations. That is, a benchmark network, which contains less time-variant species, will have lower RMSE. Thus we can clearly see, from Table 8-2, the RMSE of the network n-58 is larger than other networks. This observation is particularly obvious when the number of perturbation is small.

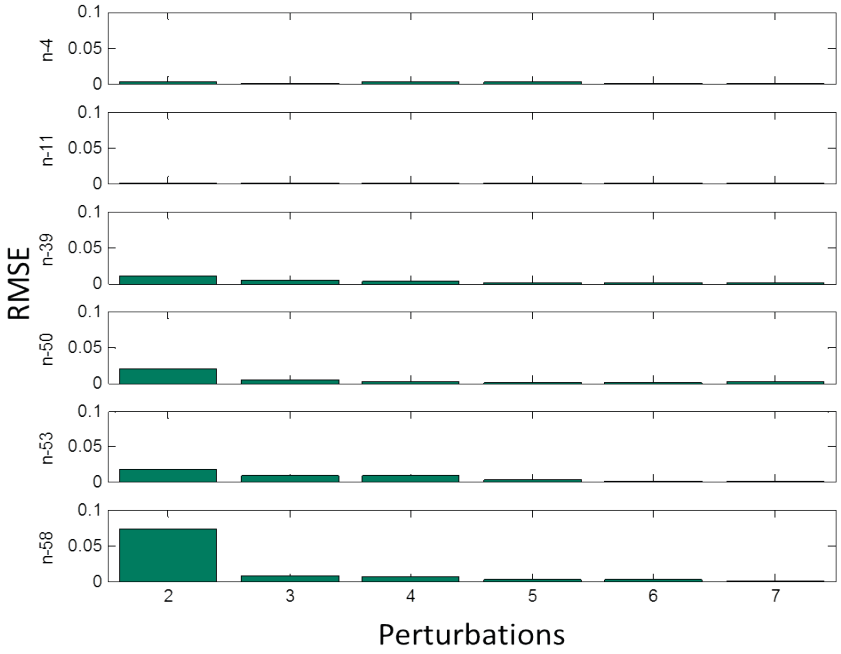


Figure 8-10 Bar charts of RMSE for transition matrix $\boldsymbol{A}$ inference. The mean RMSE for each network ranges from 0.0002 to 0.0137.

Table 8-2 RMSE of the matrix $\boldsymbol{A}$ for all 6 networks under different number of perturbations

	n-4	n-11	n-39	n-50	n-53	n-58
2 Perturbations	0.0027	0.0010	0.011	0.018	0.017	0.0735
3 Perturbations	$2.3*10^{-6}$	$1.3*10^{-5}$	0.0037	0.0051	0.0078	0.0092
4 Perturbations	0.0010	$7.1*10^{-6}$	0.0023	0.0034	0.0076	0.0061
5 Perturbations	0.0025	$6.6*10^{-6}$	$1.5*10^{-5}$	0.0018	0.0028	0.0024
6 Perturbations	$1.2*10^{-6}$	$6.6*10^{-6}$	0.0013	$1.2*10^{-5}$	0.0017	0.0022
7 Perturbations	$1.2*10^{-6}$	$6.4*10^{-6}$	0.0018	0.0020	$1.1*10^{-5}$	$1.2*10^{-5}$
8 Perturbations	0.0027	$5.7*10^{-6}$	0.0023	0.0023	0.0023	0.0027

It is interesting to find that the number of perturbations also influences the RMSE values of the transition matrix $\boldsymbol{A}$. According to Equation 8.51, each row of the transition matrix $\boldsymbol{A}$ is inferred by gathering all information across all discrete time points. Therefore, by increasing the number of perturbations, we will get more information to do the network inference. From Figure 8-10 and Table 8-2, RMSE values of the n-58 network are most greatly influenced by the number of perturbations, while the RMSE values of the n-4 network are least affected. This is because, when the network is very sparse, with a small number of perturbations (e.g., 2), the network can be well inferred using our method. Then increasing the number of

perturbations will not be able to further reduce the RMSE value, which is already very small. However, when the network contains more time-variant species, using a small number of perturbations, RMSE of the inference results is not low. Then increasing the number of perturbations will significantly reduce RMSE. For example, as it is shown in Table 8-2, the RMSE value of the n-58 network is 0.0735 when only 2 perturbations are used. By increasing the number of perturbations from 2 to 3, a significant reduction of RMSE can be obtained. The RMSE value jumps from 0.0735 to 0.0092 by the introduction of just one more perturbation. As 0.0092 is already very low, further increase of the number of perturbations will have trivial influence of RMSE.

From the above observations, we can conclude some suggestions for the real life network inference. Before starting the wet-lab experiments, we should make a rough estimation of the sparsity of the network. If we know the network may only contain a quite limited number of time-variant species within the investigated time scale, then we can only perturb the network with 2 more 3 times. If we think the network is less sparse, then we can introduce more perturbations to the network. When we cannot estimate the sparsity of an unknown network, then we can first perturb the system, for example, 3 times. Then we can compare the results obtained from 2 perturbations with 3 perturbations. If we find a significant difference, then we do more perturbations and check whether there is a significant difference between the results of 3 and 4 perturbations. We continue this step until the results becomes stable.

Along with calculating RMSE of $\boldsymbol{A}$, we also evaluate our method by checking whether it has correctly inferred the structure of a network. As we have mentioned before, we set the non-zero element of the real $\boldsymbol{A}$ (i.e. a link amongst time-variant species), $a_{j,i}$, to be positive. If $\left(a_{j,i}\right)^{est}$ from the estimation result is larger than the threshold ε_a, then it is an inferred positive. Comparison between real positives and inferred positives, we will get pairs of true positive rate and false positive rate. Figure 8-11 shows ROC curves by plotting real positive rates against false positive rates at various ε_a. From this figure, we can see that the increasing number of perturbations will give better ROC curves for all networks. Table 8-3 shows the AUROC values that measure the area under the ROC curves from Figure 8-11. We can find that the AUROC value ranges from 0.85 to 1. The smallest value, 0.85, is occurred for the n-58 network under 2 perturbations. When the number of perturbations increases to 3, the AUROC value increases to 0.96. Consistent with the observations from Figure 8-10 and

Table 8-2, the number of perturbations affects the performance of network structure. The influence is especially significant for less sparse network.

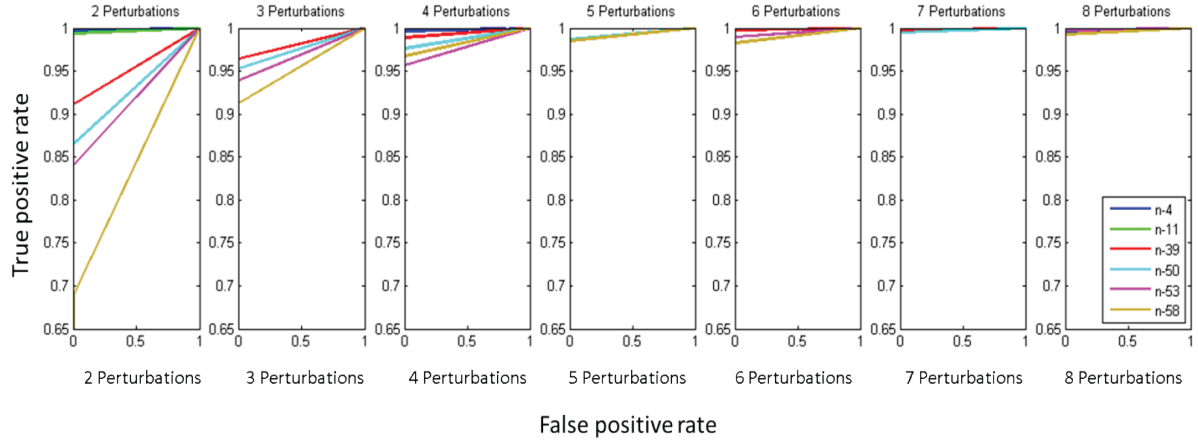


Figure 8-11 ROC curves of network structure inference. Each subplot contains results of all 6 benchmark networks under a specific number of perturbations.

Table 8-3 AUROC values of network structure inference for all 6 benchmark networks under different number of perturbations

	n-4	n-11	n-39	n-50	n-53	n-58
2 Perturbations	1	1	0.96	0.93	0.92	0.85
3 Perturbations	1	1	0.98	0.98	0.97	0.96
4 Perturbations	1	1	0.99	0.99	0.98	0.98
5 Perturbations	1	1	1	0.99	0.99	0.99
6 Perturbations	1	1	1	1	1	0.99
7 Perturbations	1	1	1	1	1	1
8 Perturbations	1	1	1	1	1	1

Now let us discuss how to choose the threshold ε_a. As the existence of a link between species j and i is detected by checking whether $\left(a_{j,i}\right)^{est}$ is larger than ε_a, the choice of ε_a is very important for network structure inference. We can investigate the way to choose ε_a by plotting specificity and sensitivity against ε_a. In Figure 8-12, the value of ε_a increases from 0 to 1 with the step size of 0.01. Subplots in the first row represent the specificities, which are the true negative ratios; and subplots in the second row of Figure 8-12 show the sensitivities of detecting real links. From the subplots for sensitivity, we can see that there is a sharp

decrease occurring around $\varepsilon_a = 0.2$. This significant change of specificity can be observed under all different numbers of perturbations; and it is particularly obvious for less sparse networks.

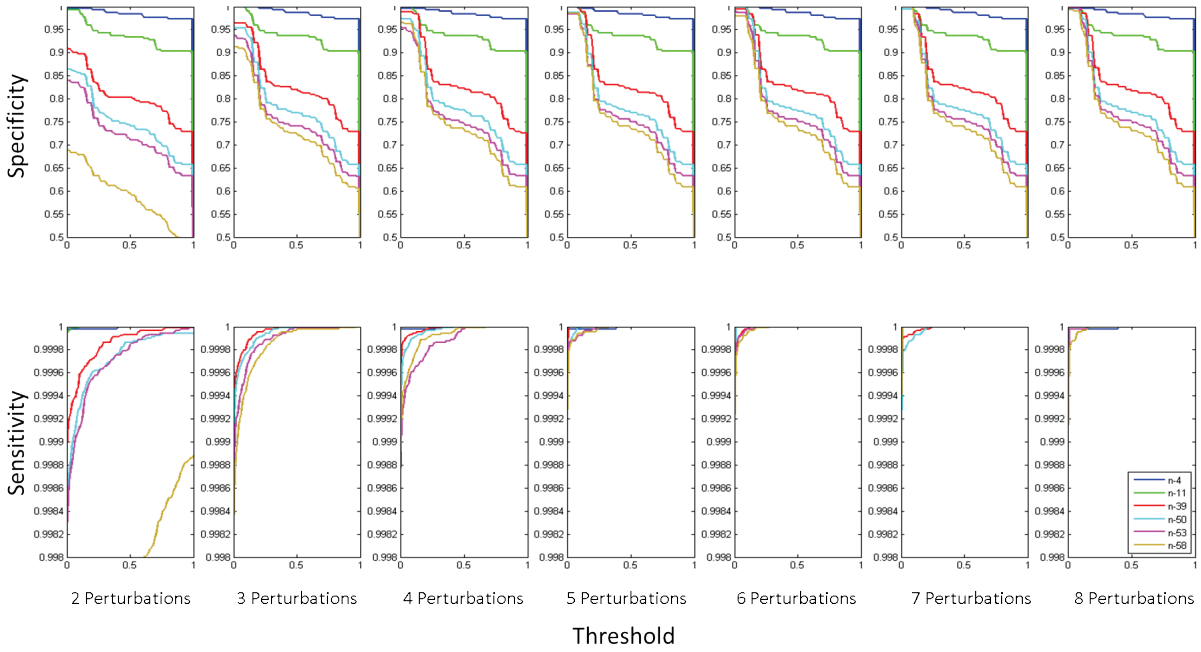


Figure 8-12 The specificity sensitivity of detecting real network structure by changing the value of threshold ε_a. The threshold varies from 0 to 1 with step size of 0.01.

From the subplots for sensitivity, we can find that the value of sensitivity maintains at a high level for most networks. With the increasing number of perturbations, the sensitivity values approach to 1 much quicker. As the sensitivity tells us the ability of identifying non-existing links, approaching to 1 faster means that a smaller value of ε_a is sufficient enough to detect false negatives. That is, the element of the estimated matrix $\boldsymbol{A}^{est}$ corresponding to zero entry of $\boldsymbol{A}$ is much closer to 0 with more perturbations. It tells us that in the estimation results the value of $\left(a_{j,i}\right)^{est}$ for $a_{j,i} = 0$ remains at a small level and the estimation error is small.

To make a balance between specificity and sensitivity, we can set the threshold ε_a smaller than 0.2 but sufficiently larger than 0. Within this range, the sensitivity is close to 1 and the specificity has not yet experienced a sharp jump. With more perturbations, we can choose a smaller threshold value, since the sensitivity values becomes 1 more quickly. We can find that in our experiment settings, the minimum non-zero value of $\boldsymbol{A}$ across all benchmark networks is around 0.2. Thus we suggest that the upper bound of ε_a is determined by the smallest non-zero value of the real transition matrix. Therefore, when our method is applied

to real network, we should determine ε_a by a rough estimation of the smallest transition rate or slightly larger than 0.

8.4.2.3 Influence of noise

In the above experiments, we fix the measurement noise to be very low. Now we investigate the performance of our method with different levels of measurement noise. Let us take the n-39 network as an example. We estimate the RMSE values of $\Delta \boldsymbol{x}_t$ and $\boldsymbol{A}$ under 6 perturbations. The standard deviation of measurement noise varies from 0.01 to 5. The results are shown in Figure 8-13 and Figure 8-14. Figure 8-13 compares the RMSE values of $\Delta \boldsymbol{x}_t$ obtained from SBL and Group SBL. It is clear that the RMSE values from Group SBL are much lower than those from SBL; and the mean value from Group SBL is 0.68 while the mean from SBL is 1.9. When the noise level increases, RMSE gradually increases as well. However, our method is more robust than SBL. Figure 8-14 shows the results of inferring $\boldsymbol{A}$. We can find that our method will have more estimation error with the increase of measurement noise. However, RMSE will not increase without bound. Instead, it will reach a stable state when the noise increases.

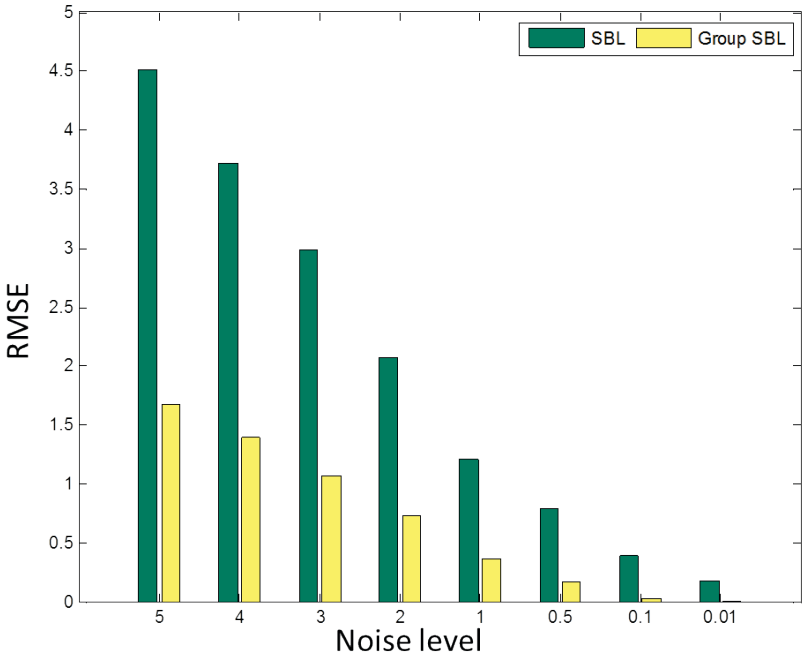


Figure 8-13 RMSE of Δx_t for the n-39 network under 6 perturbations with different levels of measurement noise. The Mean RMSE value of SBL is 1.9. For Group SBL the mean RMSE value is 0.68.

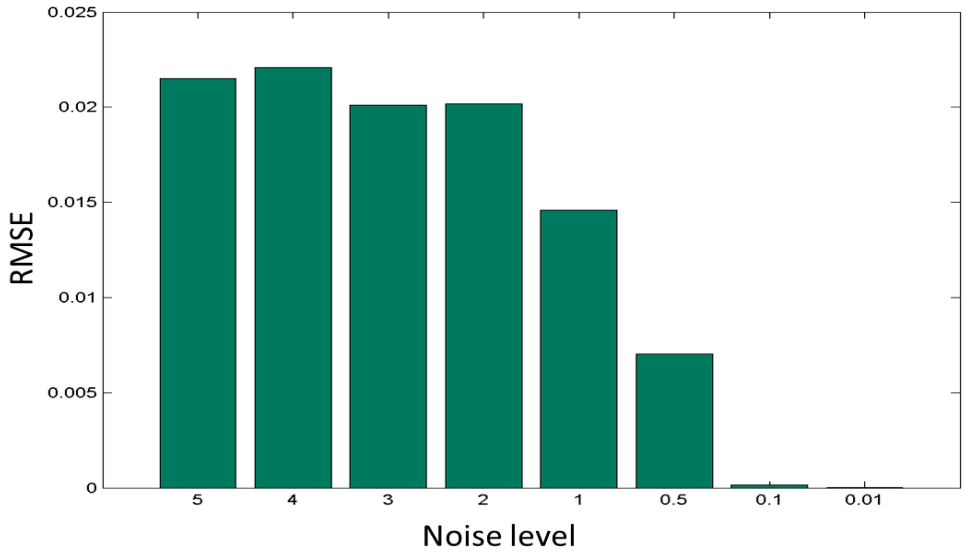


Figure 8-14 RMSE of $\boldsymbol{A}$ for the n-39 network under 6 perturbations with different levels of measurement noise. The RMSE value ranges from $1.1 * 10^{-5}$ to 0.022.

8.4.2.4 Influence of different value of T

In the above experiments, the number of time-series measurements carried out in each perturbation is fixed to 10, that is $T = 10$. Now we would also like to study the influence of different number of T. Let us also take the n-39 network as an example, where only 6 perturbations are introduced. The number of measurements varies from 4 to 12. Figure 8-16 shows the RMSE values of $\Delta \boldsymbol{x}_t$ obtained from SBL and Group SBL. It is interesting to see that the estimation error can be maintained low using Group SBL; and the RMSE values from SBL is much larger and have significant variations. This observation is quite exciting. It means that the proposed method in this chapter will be able to maintain a good performance with reduced number of measurements. Figure 8-16 shows the RMSE values of $\boldsymbol{A}$ obtained from our method. With the variation of T, RMSE can still remain at a low level, ranging from $8.2 * 10^{-6}$ to 0.053. Therefore, we can conclude that our method can reduce the number of T. As the total measurements need to be carried out is $T \times P \times M$, reducing T can save the number of measurements.

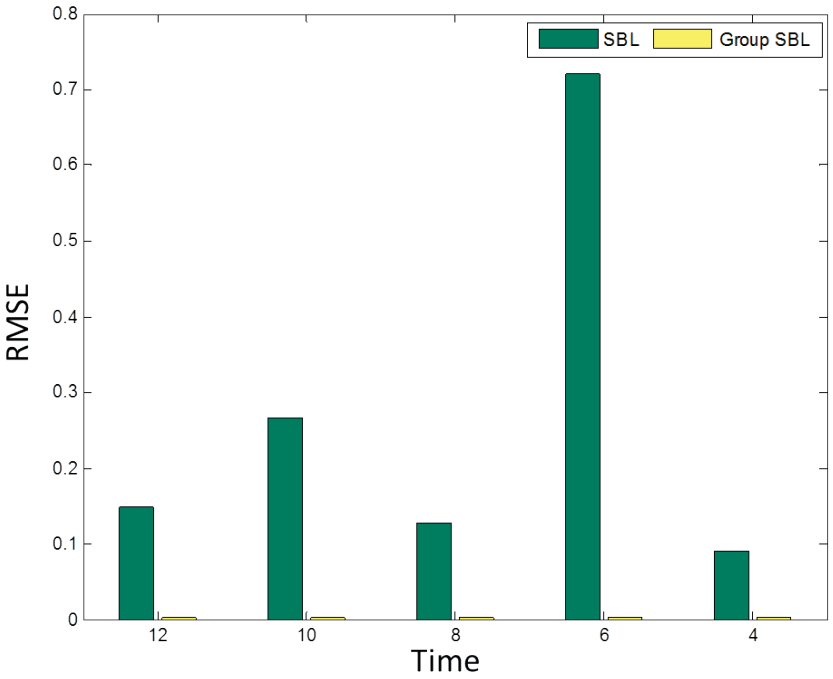


Figure 8-15 RMSE of Δx_t for the n-39 network under 6 perturbations with different number of measurements. The Mean RMSE value of SBL is 0.27. For Group SBL the mean RMSE value is 0.0036.

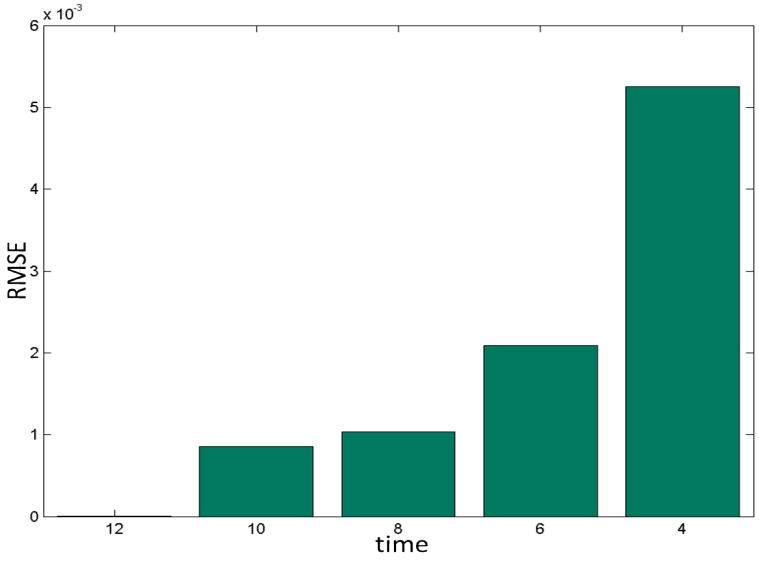


Figure 8-16 RMSE of $\boldsymbol{A}$ for the n-39 network under 6 perturbations with different number of measurements. The RMSE value ranges from $8.2 * 10^{-6}$ to 0.053.

8.5 Discussion

In our study, biological networks are modelled by linear dynamical models. Traditional methods, such as the expectation maximization (EM) algorithm, infer the transition matrix of a linear dynamic model in an iteratively way. In the E-step, a distribution of concentrations is inferred via the forward-backward algorithm based on the current estimation of transition matrix. The M-step then updates transition matrix based on the distribution of concentrations obtained in E-step. The E-step and M-step are carried out iteratively until convergence. Using the forward-backward algorithm, the estimation of time-series concentrations is closely related to the estimation of transition matrix. However, the transition matrix obtained in M-step may not be very accurate, especially when the number of measurements is limited. In our experiment, the ratio between the number of measurements and the total number of species is only 0.5 ($N = 150$ and $M = 300$). In this case, EM algorithm cannot work well as the forward-backward algorithm assumes the transition matrix is accurate. Therefore our method infers the dynamic model of networks in a two-step manner: do concentration inference first and then do transition matrix inference later. In each step, we adopt the sparse Bayesian learning method and inference procedure is referred to as type-II maximum likelihood (Tipping 2000). We can also alternatively use the variational Bayesian method (Bishop & Nasrabadi 2006), where type-II maximum likelihood method is its one special case. Unlike type-II maximum likelihood method which only returns point estimates of hyperparameters, the variational Bayesian method also returns distributions of hyperparameters.

In our experiment, we use LDPC code to randomly generate the measurement matrix. When we apply our method to infer real biological network, we need to generate the measurement matrix based on the feature of network. For example, the similarity of all involved species, such as their sequence and 3D structures should be investigated. Then we can construct a database to hold candidates of combined-measurements. In wet-lab experiments, all measurements will be chosen from this database. This chapter only focuses on introducing the method to infer networks over a predetermined timescale. To construct a complete network model over all time-scales, we need to repeat the measurement and inference procedures over different timescales. How to integrate them into a unified whole is itself an attractive problem.

In the future, we can consider using CCELL to carry out experimental design. More specifically, from the output of the transition matrix inference, we can find which elements in the transition matrix have large uncertainty, where the uncertainty can be reflected by the

covariance. Then we can carry out more measurements related to these uncertain elements in the experiments to reduce their uncertainty. In this way, we can follow an efficient experimental design strategy to improve the inference performance.

In conclusion, CCELL is a promising routine to reveal the mechanism of a complex cellular signal transduction system from a holistic perspective. The current situation, where cell-scale signal transduction models are rarely built due to its difficulty, may be changed. Signalling network databases can be built more efficiently by incorporating much more cell level models to comprehensively understand complex biological processes. Better understanding of complex biological processes is fundamental to understand life and design drugs.

9 Conclusions

9.1 Summary of thesis achievements

This thesis has studied five most frequently asked questions in quantitative data analysis, which are detecting associations, identifying groups, constructing classifiers, deriving connectivity and building dynamic models. For each question, we propose our own methods. Those methods are applied to handle different data types, which are generated from the fields of Bioinformatics and medical imaging. This thesis has shown that applying methods which are traditionally used in some particular fields to a new field can bring new insights into research questions. The detailed contributions of this thesis are:

- A new index, CR-value, which is obtained from a Lasso based model, is proposed to find significant voxels in fMRI analysis. This idea is originated from the gene association study. We have demonstrated that our method can perform better than other commonly used approaches at the presence of high level noises. In contrast to GLM, whose P-values for voxels are obtained independently, our method also takes the smoothness constraints into account. For the detection of brain activations, active voxels derived from task-related activation tend to be gathered locally, increasing the chance of discovering functional brain regions. Explicit consideration of the smoothness of the signals can enhance sensitivity to true effects.
- A matching algorithm, based on hierarchical clustering, has developed to match identities across samples in quantitative Proteomics measurements. Benefit from this approach, all ion information gathered in mass spectrometer can be used, which may have a large impact in the area of biomarker discovery. We have shown that some features which are not found by traditional approach can be detected by our methods. Moreover, our method can increase the available protein space with complete coverage, making statistical analyses and data mining methods more feasible. A GUI has been developed to help users easily process their data. The developed tool is being used in UBIOPRED project.
- A robust feature selection approach, used in biomarker discovery, is adapted to MVPA in fMRI. Rather than investigating sophisticated machine learning methods for classification, our focus is generating an optimal feature set that can be used to build an accurate predictive model. With the application of our method, the MVPA study can be improved in the way that the returned feature set has the accuracy value

higher than the value from the stability method and the stability value higher than the value from the accuracy method

- An elastic PC algorithm, which is derived from the PC method, is proposed to approximate minimum partial correlation for functional connectivity inference in fMRI analysis. The main idea of this method is using minimum partial correlation to estimate the dependence of nodes in the network. The elastic PC algorithm is derived from PC while minimizing the influence of predefined significant level with given computation time. The beauty of the elastic PC algorithm is saving computation effort by reusing results from previous steps.
- An idea of inferring large scale signalling network model under limited measurement using sparse signal recovery techniques has been proposed. This idea is inspired from the following two key observations: 1) sparsity of variations can be obtained by separation of timescales; 2) Combined-measurement can be achieved via cross-reactivity. We suggest sparse Bayesian learning is a promising approach that can be adapted to infer networks without measuring all involved species individually. The proposed method, CCELL, can potentially reveal the mechanism of a complex cellular signal transduction system from a holistic perspective.

9.2 Future work

This thesis proposes a set of analysis methods that potentially benefit quantitative data analysis. Along with the need of technical improvements for each method, in the future a systematic approach of integrating various data types for personalized medicine should be developed. The future work is as follows:

- Human body are being quantified at various levels, where molecular level data is of increasing importance. In this thesis, we have developed a method to process high throughput Proteomics data. This method can also be used for lipidomics data. However, genomics data, which is of great importance, are not studied. Some methods proposed in this thesis can be used to analyse this data type. For example, the CR-value method for active voxel detection in fMRI analysis can be used in gene association; the elastic PC algorithm can be used to generate gene regulatory network. In the future, more work should be on genomic data, which contains a lot of biomarkers of diseases.

- Network analysis takes the interaction of components into consideration. In this thesis, we study the inference of functional connectivity and also signalling pathway model. In the future, we can generate various networks for the same patient using various data. For example, to study the brain plasticity of multiple sclerosis patients, functional connectivity and gene regulatory network can be both generated. By analysing networks from different data types together, we can obtain a comprehensive understanding of diseases.
- Having independently generated and analysed various data, the next step should be data integration. We have only pointed out we need to use various data for personalized medicine. However, we have not investigated a systematic approach to integrate them together. Data integration is a quite challenge topic. This is because various data types have different dimensionalities. Simply putting them together to build a prediction model will result that data types with more features dominate the prediction. Therefore, it is worthwhile to investigate a comprehensive approach to integrate various data together. Our current thought is mapping all molecular features to pathways and use pathways as an integrated feature. However, some data types, such as imaging data, cannot be mapped to pathways. It requires a much more work on this issue.
- Massive amount of datasets are being generated in the medical research. In this thesis, we only focus on investigating computational methods to process and extract information from them. In the future, we need to propose a scheme for optimal experimental design. This is because carrying out wet-lab experiments requires a lot of resources, which are expensive. For example, the usage of high throughput machines for Omics profiling and the consumption of antibodies in immunoprecipitation cost a lot. Ideally, an optimal experimental design scheme can reduce the amount of experimental measurements and guide the experimental procedure based on current observations in real time. In chapter 8, we have proposed a method to infer transition matrix of a dynamic model. Our initial attempt of experimental design can be based on this work. The covariance of inferred matrix can be used to indicate the uncertainty of current inference; and the next optimal experimental measurement is the one that maximally reduce the uncertainty.
- Considering the big data issue in medical research, parallel computation becomes of great importance. Currently, we are implementing our matching algorithm for

Proteomics data analysis in a distributed environment. In the future, we would like to parallelise other methods proposed in this thesis. As the basis of chapter 6 and 8 is sparse Bayesian learning, the first attempt will be finding a new way of updating parameters and hyperparameters, in which computations can be distributed. For example, we can consider changing the traditional EM approach into the framework of optimisation, while the cost function in optimisation is still derived from Bayesian model. In this way, some existing distributed optimisation methods can be used directly. Parallel computing can reduce the computation time and also make our method feasible for big data.

Bibliography

Abdulrahman, H. & Henson, R.N., 2016. Effect of trial-to-trial variability on optimal event-related fMRI design: Implications for Beta-series correlation and multi-voxel pattern analysis. *NeuroImage*, 125, pp.756–766.

Abeel, T. et al., 2009. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3), pp.392–398.

Aebersold, R. & Mann, M., 2003. Mass spectrometry-based proteomics. *Nature*, 422(6928), pp.198–207.

Ahmed, I. et al., 2011. False Discovery Rate Estimation for Stability Selection: Application to Genome-Wide Association Studies. *Statistical Applications in Genetics and Molecular Biology*, 10(1), pp.1–20.

Aho, K., Derryberry, D. & Peterson, T., 2014. Model selection for ecologists: The worldviews of AIC and BIC. *Ecology*, 95(3), pp.631–636.

Alexander, D.H. & Lange, K., 2011. Stability selection for genome-wide association. *Genetic Epidemiology*, 35(7), pp.722–728.

Alivisatos, A.P. et al., 2012. The Brain Activity Map Project and the Challenge of Functional Connectomics. *Neuron*, 74(6), pp.970–974.

Allen, J. et al., 2012. Comparing statistical methods for constructing large scale gene networks. *Plos One*, 7(1).

Alonso-Gutierrez, J. et al., 2015. Principal component analysis of proteomics (PCAP) as a tool to direct metabolic engineering. *Metabolic Engineering*, 28, pp.123–133.

Argyriou, A., Evgeniou, T. & Pontil, M., 2008. Convex multi-task feature learning. *Machine Learning*, 73, pp.243–272.

Arnold, T. & Emerson, J., 2011. Nonparametric Goodness-of-Fit Tests for Discrete Null Distributions. *The R Journal*, pp.34–39.

Ashburner, J. et al., 2013. SPM8 Manual The FIL Methods Group (and honorary members). *Functional Imaging Laboratory*, pp.475–1.

Attias, H., 1999. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems (NIPS)*. pp. 209–215.

August, E. & Papachristodoulou, A., 2009. Efficient, sparse biological network determination. *BMC systems biology*, 3, p.25.

Bailey, D.L. et al., 2005. *Positron Emission Tomography Basic Sciences*,

Baker, S.M., 2010. Adaptive Unscented Kalman Filter for Estimation of Parameters in Kinetic Metabolic Models. , pp.185–186.

Barzel, B. & Barabási, A.-L., 2013. Network link prediction by global silencing of indirect correlations. *Nature biotechnology*, 31(8), pp.720–5.

Beaumont, M.A., Zhang, W. & Baldwin, J.D., 2002. Approximate Bayesian computation in population genetics. *Genetics*, 162(4), pp.2025–35.

Beck, A. & Teboulle, M., 2009. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1), pp.183–202.

Becker, S., Bobin, J. & Candès, E.J., 2011. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 91125, pp.1–37.

Becker, S.R., Candès, E.J. & Grant, M.C., 2011. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3), pp.165–218.

Berger, J.O., 1985. Statistical Decision Theory and Bayesian Analysis. *Journal of the American Statistical Association*, 83, p.266.

Bergmann, S., Ihmels, J. & Barkai, N., 2003. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67(3 Pt 1), p.031902.

Berndt, S.I. et al., 2013. Genome-wide meta-analysis identifies 11 new loci for anthropometric traits and provides insights into genetic architecture. *Nature Genetics*, 45, pp.501–512.

Bertsekas, D. & Scientific, A., 2015. *Convex Optimization Algorithms*, Athena Scientific.

Bhavsar, P. et al., 2010. Effect of p38 MAPK inhibition on corticosteroid suppression of cytokine release in severe asthma. *European Respiratory Journal*, 35(4), pp.750–756.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. *Journal of the American Statistical Association*, 92, p.482.

Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*, Springer.

Bishop, C.M. & Tipping, M.E., 2000. Variational relevance vector machines. *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, (1), pp.46–53.

Bishop, C.M.C. & Nasrabadi, N., 2006. *Pattern recognition and machine learning*,

Boccara, N., 2010. *Modeling Complex Systems*, Springer.

Bond, N.J. et al., 2013. Improving qualitative and quantitative performance for MSE-based label-free proteomics. *Journal of Proteome Research*, 12(6), pp.2340–2353.

Bonifacino, J., Dell'Angelica, E. & Springer, T., 2001. Immunoprecipitation. *Current Protocols in Immunology*, pp.1–28.

Bosnell, R. et al., 2008. Reproducibility of fMRI in the clinical setting: implications for trial designs. *Neuroimage*, 42(2), pp.603–610.

Bosnell, R.A. et al., 2011. Motor practice promotes increased activity in brain regions structurally disconnected after subcortical stroke. *Neurorehabilitation and neural repair*, 25(7), pp.607–616.

Boyd, S., 2010. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1), pp.1–122.

Boyd, S. & Vandenberghe, L., 2004. *Convex optimisation*, Cambridge University Press.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), pp.5–32.

Buckner, R.L., Krienen, F.M. & Yeo, B.T.T., 2013. Opportunities and limitations of intrinsic functional connectivity MRI. *Nature neuroscience*, 16(7), pp.832–7.

Burniston, J.G. et al., 2014. Label-free profiling of skeletal muscle using high-definition mass spectrometry. *Proteomics*, 14, pp.2339–2344.

Bush, W.S. & Moore, J.H., 2012. Chapter 11: Genome-wide association studies. *PLoS computational biology*, 8(12), p.e1002822.

Cader, S. et al., 2006. Reduced brain functional reserve and altered functional connectivity in

patients with multiple sclerosis. *Brain*, 129(2), pp.527–537.

Candès, E.J., Romberg, J. & Tao, T., 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2), pp.489–509.

Candes, E.J. & Tao, T., 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12), pp.5406–5425.

Capinski, M. & Polanski, A., 2016. Parameter Estimation in Systems Biology Models by Using Extended Kalman Filter. In *Man–Machine Interactions 4*. Springer International Publishing, pp. 195–204.

Carlson, T.A., Schrater, P. & He, S., 2003. Patterns of activity in the categorical representations of objects. *Journal of cognitive neuroscience*, 15(5), pp.704–717.

Cha, K. et al., 2015. Discovering transnosological molecular basis of human brain diseases using biclustering analysis of integrated gene expression data. *BMC Medical Informatics and Decision Making*, 15(Suppl 1), p.S7.

Chen, C.-Y. et al., 2016. Altered Functional Connectivity of the Insula and Nucleus Accumbens in Internet Gaming Disorder: A Resting State fMRI Study. *European Addiction Research*, pp.192–200.

Chen, R. et al., 2012. Personal omics profiling reveals dynamic molecular and medical phenotypes. *Cell*, 148(6), pp.1293–1307.

Chen, S.S., Donoho, D.L. & Saunders, M.A., 1998. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1), pp.33–61.

Cheng, J. et al., 2001. Learning Bayesian Networks from Data : An Information-Theory Based Approach. *Russell The Journal Of The Bertrand Russell Archives*, pp.1–74.

Cheng, Y. & Church, G.M., 2000. Biclustering of expression data. *International Conference on Intelligent Systems for Molecular Biology*, 8, pp.93–103.

Chou, I.-C., Martens, H. & Voit, E.O., 2006. Parameter estimation in biochemical systems models with alternating regression. *Theoretical biology & medical modelling*, 3, p.25.

Clarke, G.M. et al., Basic statistical analysis in genetic case-control studies. *Nat.Protoc.*, 6(2), pp.121–133.

Cleveland, W.S. & Loader, C.R., 2001. Smoothing by Local Regression: Principles and Methods. *ANNMEDPSYCHOL*, Annales-Me(10), pp.10–49.

Cleveland, W.W.S., 1979. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74(368), pp.829–836.

Collins, G.S. et al., 2015. Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD Statement. *Annals of internal medicine*, 162(1), pp.55–63.

Colombo, D. & Maathuis, M., 2012. A modification of the PC algorithm yielding order-independent skeletons. *arXiv preprint*, (2010), pp.1–22.

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3), pp.273–297.

Cotter, D. et al., 2006. LMPD: LIPID MAPS proteome database. *Nucleic acids research*, 34(Database issue), pp.D507–D510.

Coveney, P. et al., 2014. Computational Biomedicine. In *Computational Biomedicine*. p. 296.

Cox, D.D. & Savoy, R.L., 2003. Functional magnetic resonance imaging (fMRI) "brain reading": Detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2), pp.261–270.

Cox, D.R., 1958. The regression analysis of binary sequences (with discussion). *J Roy Stat Soc B*, 20, pp.215–242.

Craddock, R.C. et al., 2013. Imaging human connectomes at the macroscale. *Nature methods*, 10(6), pp.524–539.

Curtis, R.E. et al., 2012. Finding genome-transcriptome-phenome association with structured association mapping and visualization in GenAMap. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp.327–38.

Dai, W. et al., 2009. Compressive sensing DNA microarrays. *EURASIP journal on bioinformatics & systems biology*, (i), p.162824.

Dale, A.M., Fischl, B. & Sereno, M.I., 1999. Cortical surface-based analysis: I. Segmentation and surface reconstruction. *Neuroimage*, 9(2), pp.179–194.

Dauwels, J. et al., 2010. A comparative study of synchrony measures for the early diagnosis of Alzheimer's disease based on EEG. *NeuroImage*, 49(1), pp.668–693.

Davatzikos, C. et al., 2005. Classifying spatial patterns of brain activity with machine learning methods: application to lie detection. *NeuroImage*, 28(3):663–(3), pp.663–8.

Davis, J. et al., 2014. Towards a classification of biomarkers of neuropsychiatric disease: from encompass to compass. *Molecular Psychiatry*, 20(2), pp.152–153.

Davis, T. et al., 2014. What do differences between multi-voxel and univariate analysis mean? How subject-, voxel-, and trial-level variance impact fMRI analysis. *NeuroImage*, 97, pp.271–283.

Desdouits, N., Nilges, M. & Blondel, A., 2015. Principal Component Analysis reveals correlation of cavities evolution and functional motions in proteins. *Journal of Molecular Graphics and Modelling*, 55, pp.13–24.

Donoho, D., 2006. Compressed Sensing. *Transactions on Information Theory*, 54(4), pp.1289–1306.

Drucker, E. & Krapfenbauer, K., 2013. Pitfalls and limitations in translation from biomarker discovery to clinical utility in predictive and personalised medicine. *The EPMA journal*, 4(1), p.7.

Dudley, J.T. & Karczewski, K.J., 2013. *Exploring personal genomics*,

Dunn, J.D., Reid, G.E. & Bruening, M.L., 2010. Techniques for phosphopeptide enrichment prior to analysis by mass spectrometry. *Mass Spectrometry Reviews*, 29(1), pp.29–54.

Dutta, R., Bogdan, M. & Ghosh, J.K., 2015. Model Selection and Multiple Testing - A Bayesian and Empirical Bayes Overview and some New Results. *Journal of Indian Statistical ...*, (2000), pp.1–29.

Eckardt, K.-U. et al., 2015. Autosomal dominant tubulointerstitial kidney disease: diagnosis, classification, and management—A KDIGO consensus report. *Kidney International*, 1(4), pp.1–8.

Efron, B. et al., 2004. Least angle regression. *The Annals of statistics*, 32(2), pp.407–499.

von Elm, E. et al., 2008. The Strengthening the Reporting of Observational Studies in

Epidemiology (STROBE)statement: guidelines for reporting observational studies. *J Clin Epidemiol*, 61(4), pp.344–349.

Van Essen, D.C. et al., 2012. The Human Connectome Project: A data acquisition perspective. *NeuroImage*, 62(4), pp.2222–2231.

Van Essen, D.C. et al., 2013. The WU-Minn Human Connectome Project: an overview. *NeuroImage*, 80, pp.62–79.

Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), pp.861–874.

Feizi, S. et al., 2013. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology*, 31(8), pp.726–33.

Fey, D., Findeisen, R. & Bullinger, E., 2008. Parameter estimation in kinetic reaction models using nonlinear observers facilitated by model exten. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 17(1 PART 1), pp.313–318.

Fisher, R., 1936. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2), pp.179–188.

Fisher, R. a, 1924. The distribution of the partial correlation coefficient. *Metron*, 3(3-4), pp.329–332.

Fisher, R.A., 1921. On the "probable error" of coefficient of correlations deduced from a small sample. *Metron*, 1, pp.1–32.

Fisher, R.A. & Fisher, R.A., 1915. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4), pp.507–521.

Floyer-Lea, A. & Matthews, P.M., 2004. Changing brain networks for visuomotor control with increased movement automaticity. *Journal of neurophysiology*, 92(4), pp.2405–2412.

Floyer-Lea, A. & Matthews, P.M., 2005. Distinguishable brain activation networks for short- and long-term motor skill learning. *Journal of neurophysiology*, 94(1), pp.512–518.

Fornito, A., Zalesky, A. & Breakspear, M., 2013. Graph analysis of the human connectome: promise, progress, and pitfalls. *Neuroimage*, 80, pp.426–444.

Forrest, A.R.R. et al., 2003. Phosphoregulators: Protein kinases and protein phosphatases of mouse. *Genome Research*, 13(6 B), pp.1443–1454.

Frank, S.A., 2002. Specificity and Cross-Reactivity. In *Immunology and Evolution of Infectious Disease*. p. 359.

Freedman, D., 2005. *Statistical models: Theory and practice*, Cambridge University Press.

Freedman, D. & Lane, D., 1983. A Nonstochastic Interpretation of Reported Significance Levels. *Journal of Business & Economic Statistics*, 1(4), pp.292–298.

Friedman, J. et al., 2007. Pathwise coordinate optimization. , 1(2), pp.302–332.

Friedman, J., Hastie, T. & Tibshirani, R., 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), pp.432–441.

Friman, O. & Westin, C.-F., 2005. Resampling fMRI time series. *NeuroImage*, 25(3), pp.859–67.

Friston, K., 2011. Functional and effective connectivity: a review. *Brain connectivity*, 1(1), pp.13–36.

Friston, K. et al., 1995. Statistical Parametric Maps in Functional Imaging : A General Linear Approach. *Human Brain Mapping*, 2, pp.189–210.

Friston, K. et al., 2007. Variational free energy and the Laplace approximation. *NeuroImage*, 34(1), pp.220–234.

Friston, K., Harrison, L. & Penny, W., 2003. Dynamic causal modelling. *NeuroImage*, 19(4), pp.1273–1302.

Friston, K., Moran, R. & Seth, A.K., 2013. Analysing connectivity with Granger causality and dynamic causal modelling. *Current Opinion in Neurobiology*, 23(2), pp.172–178.

Galagali, N. & Marzouk, Y.M., 2015. Bayesian inference of chemical kinetic models from proposed reactions. *Chemical Engineering Science*, 123, pp.170–190.

Gallager, R., 1962. Low-density parity-check codes. *Electrical Engineering*.

Garcia-Closas, M. et al., Genome-wide association studies identify four ER negative-specific breast cancer risk loci. *Nat.Genet.*, 45(4), p.392.

Geromanos, S.J. et al., 2009. The detection, correlation, and comparison of peptide precursor and product ions from data independent LC-MS with data dependant LC-MS/MS. *Proteomics*, 9(6), pp.1683–1695.

Getz, G., Levine, E. & Domany, E., 2000. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences of the United States of America*, 97(22), pp.12079–12084.

Glasser, M.F. et al., 2013. The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*, 80, pp.105–124.

Gomez-Cabrero, D. et al., 2014. Data integration in the era of omics: current and future challenges. *BMC systems biology*, 8 Suppl 2(Suppl 2), p.I1.

Gonzalez, O.R. et al., 2007. Parameter estimation using Simulated Annealing for S-system models of biochemical networks. *Bioinformatics*, 23(4), pp.480–486.

Gounaris, C. & Floudas, C., 2009. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1), pp.3–38.

Granger, C.W.J., 1969. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3), pp.424–438.

Granqvist, E., Oldroyd, G.E. & Morris, R.J., 2011. Automated Bayesian model development for frequency detection in biological time series. *BMC Systems Biology*, 5(1), p.97.

Griffanti, L. et al., 2014. ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging. *NeuroImage*, 95, pp.232–247.

Griffiths, J.D., 2005. *Introduction to Quantum Mechanics*, Pearson Prentice Hall.

Guo, Y. & Yang, X., 2012. System Biology Approach to Study Cancer Related Pathways. In *Systems Biology in Cancer Research and Drug Discovery*. Springer Netherlands, pp. 39–67.

Han, R. et al., 2013. Elastic algorithms for guaranteeing quality monotonicity in big data mining. In *2013 IEEE International Conference on Big Data*. pp. 45–50.

Hanson, S.J., Matsuka, T. & Haxby, J. V., 2004. Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: Is there a “face” area? *NeuroImage*, 23(1), pp.156–166.

Harrison, L., Penny, W.D. & Friston, K., 2003. Multivariate autoregressive modeling of

fMRI time series. *NeuroImage*, 19(4), pp.1477–1491.

Hartigan, J. a & Wong, M. a, 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28(1), p.100.

Hastie, T., Tibshirani, R. & Friedman, J.H., 2009. *The elements of statistical learning: data mining, inference, and prediction*,

Hastie, T. & Tibshirani, R.F., 2009. *The Elements of Statistical Learning*, New York: Springer.

Hawellek, D.J. et al., 2011. Increased functional connectivity indicates the severity of cognitive impairment in multiple sclerosis. *Proceedings of the National Academy of Sciences* , 108 (47), pp.19066–19071.

Haynes, J.D. & Rees, G., 2005. Predicting the stream of consciousness from activity in human visual cortex. *Current Biology*, 15(14), pp.1301–1307.

Haynes, J.-D. & Rees, G., 2005. Predicting the orientation of invisible stimuli from activity in human primary visual cortex. *Nature neuroscience*, 8(5), pp.686–691.

Hecker, M. et al., 2009. Gene regulatory network inference: Data integration in dynamic models-A review. *BioSystems*, 96(1), pp.86–103.

Hemelrijk, C.K., 1990. A matrix partial correlation test used in investigations of reciprocity and other social interaction patterns at group level. *Journal of Theoretical Biology*, 143(3), pp.405–420.

Hendriks, B.S., Hua, F. & Chabot, J.R., 2008. Analysis of mechanistic pathway models in drug discovery: P38 pathway. *Biotechnology Progress*, 24(1), pp.96–109.

Herbert, C.G. & Johnstone, R. a. W., 2010. *Mass Spectrometry Basics*,

Herman, G.T., 2009. *Fundamentals of computerized tomography: Image reconstruction from projection*,

Hermundstad, A.M. et al., 2013. Structural foundations of resting-state and task-based functional connectivity in the human brain. *Proceedings of the National Academy of Sciences* , 110 (15), pp.6169–6174.

Hew, M. et al., 2006. Relative corticosteroid insensitivity of peripheral blood mononuclear cells in severe asthma. *American Journal of Respiratory and Critical Care Medicine*, 174(2), pp.134–141.

Hinks, T.S.C. et al., 2015. Innate and adaptive T cells in asthmatic patients: Relationship to severity and disease mechanisms. *Journal of Allergy and Clinical Immunology*, pp.1–11.

Holehouse, A. et al., 2012. Developing a novel integrated model of p38 MAPK and glucocorticoid signalling pathways. *2012 IEEE Symposium on Computational Intelligence and Computational Biology, CIBCB 2012*, pp.69–76.

Hollingworth, W. et al., 2000. The diagnostic and therapeutic impact of MRI: An observational multi-centre study. *Clinical Radiology*, 55(11), pp.825–831.

Hug, S. et al., 2015. Bayesian Model Selection methods and their application to biological ODE systems. In *Uncertainty in Biology, a computational modeling approach*.

Hug, S. et al., 2013. High-dimensional Bayesian parameter estimation: Case study for a model of JAK2/STAT5 signaling. *Mathematical Biosciences*, 246(2), pp.293–304.

Hussain, S.F. & Ramazan, M., 2016. Biclustering of human cancer microarray data using co-similarity based co-clustering. *Expert Systems with Applications*, 55, pp.520–531.

Hutchison, R.M. et al., 2013. Dynamic functional connectivity: promise, issues, and interpretations. *Neuroimage*, 80, pp.360–378.

Hyvarinen, A. & Smith, S.M., 2013. Pairwise likelihood ratios for estimation of non-Gaussian structural equation models. *Journal of Machine Learning Research*, 14, pp.111–152.

Ito, K., Chung, K.F. & Adcock, I.M., 2006. Update on glucocorticoid action and resistance. *Journal of Allergy and Clinical Immunology*, 117(3), pp.522–543.

Iyer, S.P. et al., 2013. Inferring functional connectivity in MRI using Bayesian network structure learning with a modified PC algorithm. *NeuroImage*, 75, pp.165–75.

Jaakkola, T. & Jordan, M., 2000. Bayesian parameter estimation via variatonal methods. *Statistics and computing*, 10, pp.25–37.

Jenkinson, M. et al., 2012. Fsl. *NeuroImage*, 62(2), pp.782–790.

Jenkinson, M. et al., 2002. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2), pp.825–841.

Jensen, D.R., 1997. Mixture models: Theory, geometry and applications. *Journal of Statistical Planning and Inference*, 59(1), pp.179–181.

Ji, S., Xue, Y. & Carin, L., 2008. Bayesian compressive sensing. *IEEE Transactions on Signal Processing*, 56(6), pp.2346–2356.

Jolliffe, I.T. & Cadima, J., 2016. Principal component analysis : a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical , Physical and Engineering Sciences*, 374.

Jordan, M.I. & Wainwright, M.J., 2008. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1–2), pp.1–305.

Joyce, A.R. & Palsson, B.Ø., 2006. The model organism as a system: integrating'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7(3), pp.198–210.

Kaever, A. et al., 2014. Meta-analysis of pathway enrichment: Combining independent and dependent omics data sets. *PLoS ONE*, 9(2).

Kamitani, Y. & Tong, F., 2005. Decoding the visual and subjective contents of the human brain. *Nature neuroscience*, 8(5), pp.679–685.

Karr, J.R. et al., 2012. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2), pp.389–401.

Khan, N.L. et al., 2005. Mutations in the gene LRRK2 encoding dardarin (PARK8) cause familial Parkinson's disease: Clinical, pathological, olfactory and functional imaging and genetic data. *Brain*, 128(12), pp.2786–2796.

Kholodenko, B.N., 2006. Cell-signalling dynamics in time and space. *Nature Cell Biology*, 7(March), pp.165–176.

Killiany, R. et al., 2000. Use of structural magnetic resonance imaging to predict who will get Alzheimer's disease. *Ann Neurol*, 47(4), pp.430–439.

King, R. et al., 2010. Journal of Statistical Software. , 36(August), pp.1–2.

Kirk, P. et al., 2013. Balancing the robustness and predictive performance of biomarkers. *Journal of computational biology*, 20(12), pp.979–89.

Kirk, P.D.W., Toni, T. & Stumpf, M.P.H., 2008. Parameter inference for biochemical systems that undergo a Hopf bifurcation. *Biophysical journal*, 95, pp.540–549.

Koboldt, D.C. et al., 2012. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418), pp.61–70.

Kolch, W., 2000. Meaningful relationships: the regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *The Biochemical journal*, 351 Pt 2, pp.289–305.

Koller, D. & Friedmand, N., 2009. *Probabilistic graphical models - Principles and techniques*,

de la Fuente, A. et al., 2004. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics (Oxford, England)*, 20(18), pp.3565–3574.

Lange, E. et al., 2007. A geometric approach for the alignment of liquid chromatography-mass spectrometry data. *Bioinformatics (Oxford, England)*, 23(13), pp.i273–81.

Lauritzen, S.L., 1996. *Graphical models*, Oxford university press.

LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning. *Nature*, 521(7553), pp.436–444.

Lee, H. et al., 2011. Sparse brain network recovery under compressed sensing. *IEEE Transactions on Medical Imaging*, 30(5), pp.1154–1165.

Leonardi, N. et al., 2013. Principal components of functional connectivity: a new approach to study dynamic brain connectivity during rest. *NeuroImage*, 83, pp.937–950.

Levin, Y. et al., 2007. Label-free LC-MS/MS quantitative proteomics for large-scale biomarker discovery in complex samples. *Journal of Separation Science*, 30(14), pp.2198–2203.

Li, G.Z. et al., 2009. Database searching and accounting of multiplexed precursor and product ion spectra from the data independent analysis of simple and complex peptide mixtures. *Proteomics*, 9(6), pp.1696–1719.

Li, Y. et al., 2015. Supplementary: Modular construction of mammalian gene circuits using TALE transcriptional repressors. *Nature chemical biology*, 11(february), pp.1–28.

Liao, R., McKeown, M.J. & Krolik, J.L., 2006. Isolation and minimization of head motion-induced signal variations in fMRI data using independent component analysis. *Magnetic resonance in medicine*, 55(6), pp.1396–1413.

Lillacci, G. & Khammash, M., 2010. Parameter Estimation and Model Selection in Computational Biology. , 6(3), p.17.

Locke, A. et al., 2015. Genetic studies of body mass index yield new insights for obesity biology. *Nature*, 518(7538), pp.197–206.

Lockhart, R. et al., 2014. A significance test for the lasso. *Annals of Statistics*, 42(2), pp.413–468.

Lu, Y. et al., 2014. Personalized Medicine and Human Genetic Diversity. *Cold Spring Harbor Perspectives in Medicine*, 4(9), p.a008581.

Lucas, J.E. et al., 2012. Metaprotein expression modeling for label-free quantitative proteomics. *BMC Bioinformatics*, 13(1), p.74.

Lum, P.Y. et al., 2013. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3, p.1236.

Luo, C. et al., 2014. Reduced functional connectivity in early-stage drug-naive Parkinson's disease: A resting-state fMRI study. *Neurobiology of Aging*, 35(2), pp.431–441.

Luscombe, N.M., Greenbaum, D. & Gerstein, M., 2001. What is bioinformatics ? An introduction and overview. *Yearbook of Medical Informatics*, pp.83–100.

MacKay, D.J.C., 1992. Bayesian Interpolation. *Neural Computation*, 4(3), pp.415–447.

Madeira, S.C. & Oliveira, A.L., 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 1(1), pp.24–45.

Manning, G. et al., 2002. The protein kinase complement of the human genome. *Science (New York, N.Y.)*, 298(5600), pp.1912–1934.

Manolio, T.A., 2010. Genomewide Association Studies and Assessment of the Risk of Disease. *N. Engl. J. Med.*, 363(2), pp.166–176.

Marcus, D.S. et al., 2013. Human Connectome Project informatics: Quality control, database services, and data visualization. *NeuroImage*, 80, pp.202–219.

Marrelec, G. et al., 2006. Partial correlation for functional brain interactivity investigation in functional MRI. *Neuroimage*, 32(1), pp.228–237.

Matthews, P.M. & Jezzard, P., 2004. Functional magnetic resonance imaging. *Journal of neurology, neurosurgery, and psychiatry*, 75(1), pp.6–12.

McShane, L.M. et al., 2006. REporting recommendations for tumor MARKer prognostic studies (REMARK). *Breast Cancer Research and Treatment*, 100(2), pp.229–235.

Meinshausen, N. & Bühlmann, P., 2006. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3), pp.1436–1462.

Meinshausen, N. & Bühlmann, P., 2010. Stability selection. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 72(4), pp.417–473.

Meinshausen, N., Meier, L. & Bühlmann, P., 2008. P-values for high-dimensional regression. , p.25.

Mendes, P. & Kell, D.B., 1998. Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14(10), pp.869–883.

Minnier, J., Tian, L. & Cai, T., 2011. A Perturbation Method for Inference on Regularized Regression Estimates. *Journal of the American Statistical Association*, 106(496), pp.1371–1382.

Möller, C. et al., 2015. Alzheimer Disease and Behavioral Variant Frontotemporal Dementia: Automatic Classification Based on Cortical Atrophy for Single-Subject Diagnosis. *Radiology*, p.150220.

Molloy, E.K., Meyerand, M.E. & Birn, R.M., 2014. The influence of spatial resolution and smoothing on the detectability of resting-state and task fMRI. *NeuroImage*, 86, pp.221–230.

Moore, W.C. et al., 2010. Identification of asthma phenotypes using cluster analysis in the severe asthma research program. *American Journal of Respiratory and Critical Care Medicine*, 181(4), pp.315–323.

Motoda, H. & Fawcett, T., 2004. Learning to Decode Cognitive States from Brain Images. , pp.145–175.

Murphy, K., 2012. *Machine Learning: a Probabilistic Perspective*, MIT press.

Myers, J.L., Well, A.D. & Jr, R.F.L., 2010. *Research design and statistical analysis*, Routledge.

Necela, B.M. & Cidlowski, J.A., 2004. Mechanisms of glucocorticoid receptor action in

noninflammatory and inflammatory cells. *Proceedings of the American Thoracic Society*, 1(3), pp.239–246.

Neilson, K.A. et al., 2011. Less label, more free: Approaches in label-free quantitative mass spectrometry. *Proteomics*, 11(4), pp.535–553.

Newman, M.E.J., 2010. *Networks: An Introduction*, OUP Oxford.

Nichols, T.E. & Holmes, A.P., 2001. Nonparametric Permutation Tests For Functional Neuroimaging : A Primer with Examples. , 25(August 1999), pp.1–25.

Nicholson, A.A. et al., 2015. The Dissociative Subtype of Posttraumatic Stress Disorder: Unique Resting-State Functional Connectivity of Basolateral and Centromedial Amygdala Complexes. *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology*, 40(10), pp.2317–26.

Nicolau, M., Levine, A.J. & Carlsson, G., 2011. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences of the United States of America*, 108(17), pp.7265–7270.

Nie, L. et al., 2014. Inferring cell-scale signalling networks via compressive sensing. *PLoS ONE*, 9(4).

Nie, L. et al., 2015. Minimum Partial Correlation: An Accurate and Parameter-Free Measure of Functional Connectivity in fMRI. In *2015 international conference on brain informatics and health*.

Nikolsky, Y. et al., 2009. Functional analysis of OMICs data and small molecule compounds in an integrated "knowledge-based" platform. *Methods in molecular biology (Clifton, N.J.)*, 563, pp.177–196.

Norman, K. a. et al., 2006. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, 10(9), pp.424–430.

Le Novère, N., 2015. Quantitative and logic modelling of molecular and gene networks. *Nature reviews. Genetics*, 16(3), pp.146–58.

O'Toole, A.J. et al., 2005. Partially distributed representations of objects and faces in ventral temporal cortex. *Journal of cognitive neuroscience*, 17(4), pp.580–590.

Oghabian, A. et al., 2014. Biclustering Methods : Biological Relevance and Application in Gene Expression Analysis. , 9(3).

Olson, C.F., 1995. Parallel algorithms for hierarchical. , 21, pp.1313–1325.

Ow, S.Y. et al., 2009. iTRAQ Underestimation in Simple and Complex Mixtures : " The Good , the Bad and the Ugly ." *J Proteome Res*, 8(11), pp.5347–5355.

P. Murphy, K., 1991. *Machine Learning: A Probabilistic Perspective*, MIT press.

Papin, J.A. & Palsson, B.O., 2004. The JAK-STAT signaling network in the human B-cell: an extreme signaling pathway analysis. *Biophysical journal*, 87(1), pp.37–46.

Parikshak, N.N., Gandal, M.J. & Geschwind, D.H., 2015. Systems biology and gene networks in neurodevelopmental and neurodegenerative disorders. *Nature Reviews Genetics*, 16(8), pp.441–458.

Park, H.W. et al., 2015. Classification and implementation of asthma phenotypes in elderly patients. *Annals of Allergy, Asthma and Immunology*, 114(1), pp.18–22.

Patel, R.S. et al., 2006. A Bayesian approach to determining connectivity of the human brain.

Hum Brain Mapp, 27, pp.267–276.

Pearl, J., 2000. *Causality: models, reasoning and inference*, Cambridge Univ Press.

Pearson, K., 2006. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London (1854-1905)*, 58(-1), pp.240–242.

Pearson, T.A. & Manolio, T.A., 2008. How to interpret a genome-wide association study. *JAMA*, 299(11), pp.1335–1344.

Peifer, M. & Timmer, J., 2007. Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *Engineering and Technology*, (2), pp.78–88.

Peng, H., Long, F. & Ding, C., 2005. Feature selection based on mutual information: criteria of max-dependency. *IEEE transactions on pattern analysis and*.

Peng, J. et al., 2009. Partial Correlation Estimation by Joint Sparse Regression Models. *Journal of the American Statistical Association*, 104(486).

Penny, W. et al., 2006. *Statistical Parametric Mapping: The Analysis of Functional Brain Images: The Analysis of Functional Brain Images*,

Penny, W.D., Trujillo-Barreto, N.J. & Friston, K., 2005. Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24, pp.350–362.

Poldrack, R. a & Nichols, T.E., 2011. *Handbook of functional MRI data analysis*, Cambridge University Press.

Polyn, S.M. et al., 2005. Category-specific cortical activity precedes retrieval during memory search. *Science (New York, N.Y.)*, 310(5756), pp.1963–1966.

Poppe, A.B. et al., 2015. Task-based functional connectivity as an indicator of genetic liability to schizophrenia. *Schizophrenia Research*, 162(1-3), pp.118–123.

Proutski, V., 2010. Discovering Genomics Biomarker for Clinical Test: Real Life Example. In *Theranostics Conference 2010*.

Purcell, S. et al., 2007. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics*, 81(3), pp.559–575.

Quach, M., Brunel, N. & D'alché-Buc, F., 2007. Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference. *Bioinformatics*, 23(23), pp.3209–3216.

Quinlan, J.R., 1987. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3), pp.221–234.

Rabiner, E.A. et al., 2011. Pharmacological differentiation of opioid receptor antagonists by molecular and functional imaging of target occupancy and food reward-related brain activation in humans. *Molecular psychiatry*, 16(8), pp.826–835.

Reshef, D.N. et al., 2011. Detecting Novel Associations in Large Data Sets. *Science*, 334(6062), pp.1518–1524.

Ribeiro, D. & Pinto, J.M., 2009. An integrated network-based mechanistic model for tumor growth dynamics under drug administration. *Computers in Biology and Medicine*, 39(4), pp.368–384.

Rish, I., 2001. An empirical study of the naive Bayes classifier. In *Proc. of th IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*. pp. 1–6.

Rondina, J.M. et al., 2014. SCoRS-A method based on stability for feature selection and

apping in neuroimaging. *IEEE Transactions on Medical Imaging*, 33(1), pp.85–98.

Rubin, D.B., 1983. Iteratively reweighted least squares. *Encyclopedia of Statistical Sciences*, 4(5), pp.272–275.

Rucco, M. et al., 2014. Using Topological Data Analysis for diagnosis pulmonary embolism. *ArXiv e-prints*.

Ruckstuhl, A.F. et al., 2001. Baseline subtraction using robust local regression estimation. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 68(2), pp.179–193.

Ryali, S. et al., 2012. Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage*, 59(4), pp.3852–3861.

Sadygov, R.G., Cociorva, D. & Yates, J.R., 2004. Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nature methods*, 1(3), pp.195–202.

Sahin, F. & Bay, J.S., 1998. A Radial Basis Function Approach to a Color Image Classification Problem in a Real Time Industrial Application. In *1st International Conference on Multisource-Multisensor Information Fusion*. pp. 873–879.

Salimi-Khorshidi, G. et al., 2014. Automatic denoising of functional MRI data: combining independent component analysis and hierarchical fusion of classifiers. *NeuroImage*, 90, pp.449–468.

Sarvotham, S., Baron, D. & Baraniuk, R., 2006. Compressed sensing reconstruction via belief propagation. *preprint*, pp.1–24.

Schneider, M. V. & Orchard, S., 2011. Omics technologies, data and bioinformatics principles. *Methods in molecular biology (Clifton, N.J.)*, 719, pp.3–30.

Schulze, W.X. & Usadel, B., 2010. Quantitation in mass-spectrometry-based proteomics. *Annual review of plant biology*, 61, pp.491–516.

Schwaab, M. et al., 2008. Nonlinear parameter estimation through particle swarm optimization. *Chemical Engineering Science*, 63, pp.1542–1552.

Schwarz, G., 1978. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2), pp.461–464.

Sensen, C.W. & Hallgrímsson, B., 2009. *Advanced imaging in biology and medicine: Technology, software environments, applications*,

Shan, Z.Y. et al., 2014. Modeling of the hemodynamic responses in block design fMRI studies. *Journal of cerebral blood flow and metabolism : official journal of the International Society of Cerebral Blood Flow and Metabolism*, 34(2), pp.316–24.

Shaw, D.E. et al., 2015. Clinical and inflammatory characteristics of the European U-BIOPRED adult severe asthma cohort. In *European Respiratory Journal*. pp. 1308–1321.

Shimizu, S., 2006. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7, pp.2003–2030.

Shirer, W.R. et al., 2012. Decoding subject-driven cognitive states with whole-brain connectivity patterns. *Cerebral cortex (New York, N.Y. : 1991)*, 22(1), pp.158–65.

Shliaha, P. V. et al., 2013. Effects of traveling wave ion mobility separation on data independent acquisition in proteomics studies. *Journal of Proteome Research*, 12(6), pp.2323–2339.

Shungin, D. et al., 2015. New genetic loci link adipose and insulin biology to body fat distribution. *Nature*, 518(7538), pp.187–196.

Shuto, T. et al., 2002. Glucocorticoids synergistically enhance nontypeable Haemophilus influenzae-induced toll-like receptor 2 expression via a negative cross-talk with p38 MAP kinase. *Journal of Biological Chemistry*, 277(19), pp.17263–17270.

Silva, J.C. et al., 2006. Absolute quantification of proteins by LCMSE: a virtue of parallel MS acquisition. *Molecular & cellular proteomics : MCP*, 5(1), pp.144–156.

Silva, J.C. et al., 2005. Quantitative proteomic analysis by accurate mass retention time pairs. *Anal Chem*, 77(7), pp.2187–2200.

Simon, D. & Simon, D.L., 2003. Aircraft Turbofan Engine Health Estimation Using Constrained Kalman Filtering. , (August).

Sisson, S. a, Fan, Y. & Tanaka, M., 2007. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6), pp.1760–1765.

Skates, S.J. et al., 2013. Statistical design for biospecimen cohort size in proteomics-based biomarker discovery and verification studies. *Journal of Proteome Research*, 12(12), pp.5383–5394.

Sladky, R. et al., 2011. Slice-timing effects and their correction in functional MRI. *NeuroImage*, 58(2), pp.588–594.

Smith, A.T., Singh, K.D. & Balsters, J.H., 2007. A comment on the severity of the effects of non-white noise in fMRI time-series. *NeuroImage*, 36(2), pp.282–288.

Smith, C.A. et al., 2006. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical chemistry*, 78(3), pp.779–87.

Smith, R., 2012. Stratified, personalised, or precision medicine.

Smith, S.M. et al., 2004. Advances in functional and structural MR image analysis and implementation as FSL. *NeuroImage*, 21, pp.208–219.

Smith, S.M. et al., 2013. Functional connectomics from resting-state fMRI. *Trends in Cognitive Sciences*, 17(12), pp.666–682.

Smith, S.M. et al., 2011. Network modelling methods for FMRI. *NeuroImage*, 54(2), pp.875–891.

Smith, S.M., 2004. Overview of fMRI analysis. *British Journal of Radiology*, 77(SPEC. ISS. 2).

Smith, S.M. et al., 2001. Overview of fMRI analysis. In The British Journal of Radiology, pp. 215–227.

Smith, S.M. et al., 2013. Resting-state fMRI in the Human Connectome Project. *NeuroImage*, 80, pp.144–68.

Smith, S.M., 2012. The future of FMRI connectivity. *NeuroImage*, 62(2), pp.1257–66.

Smith, S.M. & Nichols, T.E., 2009. Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, 44, pp.83–98.

Sparacino, G. & De Nicolao, G., 2001. *Modeling Methodology for Physiology and Medicine*,

Spirtes, P. & Glymour, C., 2000. *Causation, Prediction, and Search*, The MIT Press.

Spirtes, P., Glymour, C.N. & Scheines, R., 2000. *Causation, Prediction, and Search*, The MIT Press.

Strongin, R.G. & Sergeyev, Y.D., 2013. *Global optimization with non-convex constraints: Sequential and parallel algorithms*, Springer Science & Business Media.

Strother, S.C., 2006. Evaluating fMRI preprocessing pipelines. *IEEE Engineering in Medicine and Biology Magazine*, 25(2), pp.27–41.

Sun, X., Jin, L. & Xiong, M., 2008. Extended Kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks. *PLoS ONE*, 3(11).

Swameye, I. et al., 2003. Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 100(3), pp.1028–1033.

Tanay, A., 2005. Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 9(May), pp.122–124.

Tanay, A. et al., 2004. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9), pp.2981–2986.

Tao, D. et al., 2006. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 28(7), pp.1088–1099.

Tibshirani, R., 1994. Regression Selection and Shrinkage via the Lasso. *Journal of the Royal Statistical Society B*, 58, pp.267–288.

Tipping, M., 2000. The relevance vector machine. In *NIPS*.

Tipping, M.E., 2006. Bayesian Inference: An Introduction to Principles and Practice in Machine Learning. *Machine Learning*, pp.1–19.

Tipping, M.E., 2001. Sparse Bayesian Learning and the Relevance Vector Machine. *J ournal of Machine Learning Research*, 1, pp.211–244.

Tomassini, V. et al., 2012. Relating Brain Damage to Brain Plasticity in Patients With Multiple Sclerosis. *Neurorehabilitation and neural repair*, 26(6), pp.581–593.

Toni, T. et al., 2009. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society, Interface / the Royal Society*, 6, pp.187–202.

Toni, T. & Stumpf, M.P.H., 2010. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, 26(1), pp.104–110.

Tsamardinos, I., Brown, L.E. & Aliferis, C.F., 2006. The max-min hill-climbing bayesian network structure learning algorithm. In p. 2006.

Turk-Browne, N.B., 2013. Functional interactions as big data in the human brain. *Science*, 342(6158), pp.580–584.

Tzourio-Mazoyer, N. et al., 2002. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1), pp.273–289.

Varoquaux, G. et al., 2010. Brain covariance selection: better individual functional connectivity models using population prior. *Advances in Neural Information Processing Systems*, pp.2334–2342.

Varoquaux, G., Gramfort, A. & Thirion, B., 2012. Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp.1375–1382.

Veiga, D.F.T. et al., 2007. Genome-wide partial correlation analysis of Escherichia coli microarray data. *Genetics and molecular research : GMR*, 6(4), pp.730–742.

Veselkov, K.A. et al., 2011. Optimized preprocessing of ultra-performance liquid chromatography/mass spectrometry urinary metabolic profiles for improved information recovery. *Analytical Chemistry*, 83(15), pp.5864–5872.

Villaverde, A.F. & Banga, J.R., 2014. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of the Royal Society, Interface / the Royal Society*, 11(91), p.20130505.

Völlm, B.A. et al., 2004. Methamphetamine activates reward circuitry in drug na{\i}ve human subjects. *Neuropsychopharmacology*, 29, pp.1715–1722.

Wang, J. et al., 2010. Graph-based network analysis of resting-state functional MRI. , 4(June), pp.1–14.

Wang, Z. et al., 2009. An extended kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3), pp.410–419.

Wang, Z. et al., 2006. Phosphatase-mediated crosstalk control of ERK and p38 MAPK signaling in corneal epithelial cells. *Investigative Ophthalmology and Visual Science*, 47(12), pp.5267–5275.

Wang, Z. & Chan, L., 2012. Learning Bayesian Networks from Markov Random Fields: An Efficient Algorithm for Linear Models. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(3).

Wasserman, L. & Roeder, K., 2009. High-dimensional variable selection. *Annals of Statistics*, 37(5 A), pp.2178–2201.

Wee, C.-Y. et al., 2012. Constrained sparse functional connectivity networks for MCI classification. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 15(Pt 2), pp.212–9.

Wee, C.-Y. et al., 2014. Group-constrained sparse fMRI connectivity modeling for mild cognitive impairment identification. *Brain structure & function*, 219(2), pp.641–56.

Wegdam, W. et al., 2014. Label-free LC-MSe in tissue and serum reveals protein networks underlying differences between benign and malignant serous ovarian tumors. *PloS one*, e108046.

Weigt, M. et al., 2009. Identification of direct residue contacts in protein-protein interaction by message passing. *Proceedings of the National Academy of Sciences of the United States of America*, 106(1), pp.67–72.

Welch, G. & Bishop, G., 2006. An Introduction to the Kalman Filter. *In Practice*, 7(1), pp.1–16.

Wikipedia, Sensitivity and specificity. Available at: http://en.wikipedia.org/wiki/Sensitivity_and_specificity [Accessed July 3, 2015].

Williams, A. & Halappanavar, S., 2015. Application of bi-clustering of gene expression data and gene set enrichment analysis methods to identify potentially disease causing nanomaterials. *Beilstein journal of nanotechnology*, 6(1), p.under review.

Wimmer, G.E. & Shohamy, D., 2012. Preference by Association: How Memory Mechanisms in the Hippocampus Bias Decisions. *Science*, 338(6104), pp.270–273.

Winkler, A.M. et al., 2014. Permutation inference for the general linear model. *NeuroImage*, 92, pp.381–397.

Winslow, R.L. et al., 2012. Computational Medicine: Translating Models to Clinical Care. *Science Translational Medicine*, 4(158), pp.158rv11–158rv11.

Wisittipanit, N. et al., 2015. Classification methods for the analysis of LH-PCR data associated with inflammatory bowel disease patients. *International journal of bioinformatics research and applications*, 11(2), pp.111–129.

Wolkenhauer, O., 2014. Why model? *Frontiers in Physiology*, 5 JAN(January), pp.1–5.

Wood, A. et al., 2014. Defining the role of common variation in the genomic and biological architecture of adult human height. *Nat Genet.*, 46(11), pp.1173–86.

Wright, S.J., Nowak, R.D. & Figueiredo, M.A.T., 2009. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7), pp.2479–2493.

Wu, W. et al., 2014. Unsupervised phenotyping of Severe Asthma Research Program participants using expanded lung data. *Journal of Allergy and Clinical Immunology*, 133(5), pp.1280–1288.

Xia, M., Wang, J. & He, Y., 2013. BrainNet Viewer: A Network Visualization Tool for Human Brain Connectomics. *PLoS ONE*, 8(7).

Xiong, H. & Choe, Y., 2008. Structural systems identification of genetic regulatory networks. *Bioinformatics (Oxford, England)*, 24(4), pp.553–560.

Yan, S., Yang, X., Wu, C., Zheng, Z., et al., 2014. Balancing the Stability and Predictive Performance for Multivariate Voxel Selection in fMRI Study. In *The 2014 International Conference on Brain Informatics and Health*.

Yan, S., Yang, X., Wu, C., Guo, Y., et al., 2014. Integration of sparse Bayesian learning and random subspace for fMRI Multivariate Pattern Analysis. In *Engineering in Medicine and Biology Society (EMBC), 36th Annual International Conference of the IEEE*.

Yang, X., Guo, Y., et al., 2012. Automating Mass Spectrometry Proteomics Analysis. In *Fourth International Conference on Bioinformatics and Computational Biology*.

Yang, X., Han, R., et al., 2012. Modelling and performance analysis of clinical pathways using the stochastic process algebra PEPA. *BMC bioinformatics*, 13 Suppl 1(Suppl 14), p.S4.

Yang, X. et al., 2015. The Critical Regularization Value: Incorporating Spatial Smoothness to Enhance Signal Detection in Highly Noisy fMRI Data. In *7th international IEEE/EMBS conference on neural engineering*.

Yang, X., Guo, Y. & Guo, L., 2014. An iterative parameter estimation method for biological systems and its parallel implementation. *Concurrency and Computation: Practice and Experience*, 26(6), pp.1249–1267.

Yeung, M.K.S., Tegnér, J. & Collins, J.J., 2002. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9), pp.6163–6168.

Zhan, C. & Yeung, L.F., 2011. Parameter estimation in systems biology models using spline approximation. *BMC systems biology*, 5(1), p.14.

Zhang, G. et al., 2010. Overview of peptide and protein analysis by mass spectrometry.

Current protocols in protein science, Chapter 16(November), p.Unit16.1.

Zhang, J.D. et al., 2015. Pathway reporter genes define molecular phenotypes of human cells. *BMC genomics*, 16(1), p.342.

Zomorodian, A., 2011. Topological data analysis. *Inverse Problems*, 27(12), p.120201.

Zou, H. & Hastie, T., 2005. Regularization and variable selection via the elastic-net. *Journal of the Royal Statistical Society*, 67, pp.301–320.

Zucknick, M., Richardson, S. & Stronach, E. a, 2008. Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods. *Statistical applications in genetics and molecular biology*, 7(1), p.Article7.

Zuo, Y. et al., 2014. Biological network inference using low order partial correlation. *Methods (San Diego, Calif.)*, 69(3), pp.266–73.