Journal: Protein Engineering, Design & Selection

Article DOI: gzw009

Article title: The Parasol Protocol for computational mutagenesis

First Author: P.G.A. Aronica

Corr. Author: R.J. Leatherbarrow

INSTRUCTIONS

1. **Author groups:** Please check that all names have been spelled correctly and appear in the correct order. Please also check that all initials are present. Please check that the author surnames (family name) have been correctly identified by a pink background. If this is incorrect, please identify the full surname of the relevant authors. Occasionally, the distinction between surnames and forenames can be ambiguous, and this is to ensure that the authors' full surnames and forenames are tagged correctly, for accurate indexing online. Please also check all author affiliations.

2. **Figures:** If applicable figures have been placed as close as possible to their first citation. Please check that they are complete and that the correct figure legend is present. Figures in the proof are low resolution versions that will be replaced with high resolution versions when the journal is printed.

3. **Missing elements:** Please check that the text is complete and that all figures, tables and their legends are included.

4. **Special characters:** Please check that special characters, equations, dosages and units, if applicable, have been reproduced accurately.

5. **Funding:** Please provide a Funding statement, detailing any funding received. Remember that any funding used while completing this work should be highlighted in a separate Funding section. Please ensure that you use the full official name of the funding body, and if your paper has received funding from any institution, such as NIH, please inform us of the grant number to go into the funding section. We use the institution names to tag NIH-funded articles so they are deposited at PMC. If we already have this information, we will have tagged it and it will appear as coloured text in the funding paragraph. Please check the information is correct.

Journal: Protein Engineering, Design & Selection

Article DOI: gzw009

Article title: The Parasol Protocol for computational mutagenesis

First Author: P.G.A. Aronica

Corr. Author: R.J. Leatherbarrow

AUTHOR QUERIES - TO BE ANSWERED BY THE CORRESPONDING AUTHOR

The following queries have arisen during the typesetting of your manuscript. Please click on each query number and respond by indicating the change required within the text of the article. If no change is needed please add a note saying "No change."

Query No.	Nature of Query
Q1	Please check that all names have been spelled correctly and appear in the correct order. Please also check that all initials are present. Please check that the author surnames (family name) have been correctly identified by a pink background. If this is incorrect, please identify the full surname of the relevant authors. Occasionally, the distinction between surnames and forenames can be ambiguous, and this is to ensure that the authors' full surnames and forenames are tagged correctly, for accurate indexing online. Please also check all author affiliations.
Q2	Figures have been placed as close as possible to their first mention in the text. Please check that the figures are accurately placed in the text, that the images are correct, and that they have the correct caption and citation.
Q3	Two figures are labelled as Figure 2. So, we have changed the second one to Figure 3 and renumbered the remaining figures. Please confirm that this is ok.
Q4	Initial citation of Figure 7 is out of order but also cited in the correct order thereafter. Is this ok or we shall remove the initial citation and renumber the figures accordingly.
Q5	Table II is cited in text, but the table is missing. Please check.
Q6	Please provide the volume number and page range for Azoitei *et al*. (2014) and Bradshaw *et al*. (2010) references.
Q7	Please provide the year of publication for the following references: Cano *et al*. and Liu and Kuhlman.
Q8	Please provide the publisher location for the following references: Chipot and Pohorille (2007), Jensen (2006), and Sanders and Kandrot (2010).
Q9	Please note that we have relabelled this figure to ensure typographical consistency. Please check that the changes made are accurate.
Q10	In order to validate your funding information prior to publication, please check and confirm whether the name of the funding body given in your manuscript is complete and correct. If any edits are required please mark them on the text. Please also expand any acronyms used in this section. If multiple grants are cited, please ensure the text of your funding statement clearly indicates which grant applies to which funding body.

Query No.	Nature of Query
Q11	Please note that there is a £350/$700 charge for each figure reproduced in colour in print. The print and online versions of the journal must be identical - we therefore do not offer colour online only. If you have supplied colour figures, please confirm that you accept the charges. Alternatively, please let us know if you would prefer to have your figures reproduced in black and white at no cost. If this is the case, please ensure that the legend/text is worded to avoid using reference to colour, or supply amended images. If we do not receive a response from you, we will assume that figures supplied in colour should be produced in colour, and you will be invoiced accordingly.

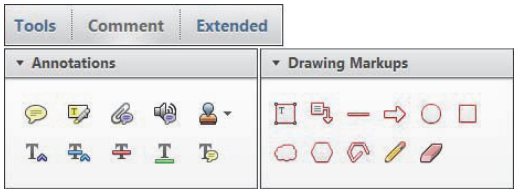
MAKING CORRECTIONS TO YOUR PROOF

These instructions show you how to mark changes or add notes to the document using the Adobe Acrobat Professional version 7 (or onwards) or Adobe Reader XI (PDF enabled for marking corrections).
To check what version you are using go to **Help** then **About**.
If you do not have Adobe Reader XI, please visit the following link to download it for free: http://get.adobe.com/reader.

Displaying the toolbars

Acrobat Professional X, XI and Reader XI

Select **Comment, Annotations and Drawing Markups.**
If this option is not available, please let me know so that I can enable it for you.

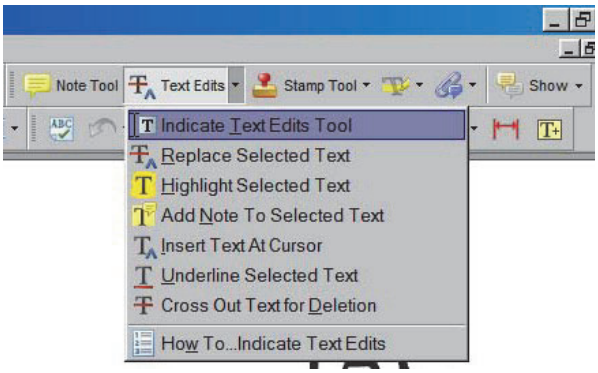


Acrobat Professional 7, 8 and 9

Select **Tools, Commenting, Show Commenting Toolbar**.

Using Text Edits

This is the quickest, simplest and easiest method both to make corrections, and for your corrections to be transferred and checked.

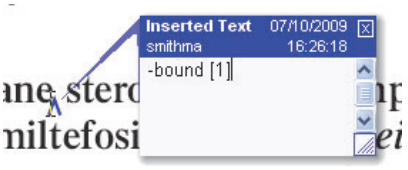


1. Click **Text Edits**
2. Select the text to be annotated or place your cursor at the insertion point.
3. Click the **Text Edits** drop down arrow and select the required action.

   You can also right click on selected text for a range of commenting options.

Pop up Notes

With *Text Edits* and other markup, it is possible to add notes. In some cases (e.g. inserting or replacing text), a pop-up note is displayed automatically.



To **display** the pop-up note for other markup, right click on the annotation on the document and selecting **Open Pop-Up Note**.

To **move** a note, click and drag on the title area.

To **resize** of the note, click and drag on the bottom right corner.

To **close** the note, click on the cross in the top right hand corner.

To **delete** an edit, right click on it and select **Delete**. The edit and associated note will be removed.

Saving Comments

In order to save your comments and notes, you need to save the file (**File, Save**) when you close the document.
A full list of the comments and edits you have made can be viewed by clicking on the Comments tab in the bottom-left-hand corner of the PDF.

Protein Engineering, Design & Selection, 2016, 1–14
doi: 10.1093/protein/gzw009
Methods

Methods

The Parasol Protocol for computational mutagenesis

P.G.A. Aronica[1], C. Verma[2], B. Popovic[3], R.J. Leatherbarrow[1,4,5], and I. R. Gould[1,5] Q1: No change

[1]Department of Chemistry and Institute of Chemical Biology, Imperial College London, South Kensington Campus, Exhibition Road, London SW7 2AZ, UK, [2]Bioinformatics Institute, A*STAR (Agency for Science, Technology and Research), Biopolis, Singapore, Singapore, [3]Department of Antibody Discovery and Protein Engineering, MedImmune, Aaron Klug Building, Granta Park, Cambridge CB21 6GH, UK, and [4]Liverpool John Moores University, 2 Rodney Street, Liverpool L1 2UA, UK

[5]To whom correspondence should be addressed. E-mail: r.leatherbarrow@imperial.ac.uk (R.J.L.); i.gould@imperial.ac.uk (I.R.G.)

Edited by Valerie Daggett

Received 7 March 2016; Revised 7 March 2016; Accepted 15 March 2016

Abstract

To aid in the discovery and development of peptides and proteins as therapeutic agents, a virtual screen can be used to predict trends and direct workflow. We have developed the Parasol Protocol, a dynamic method implemented using the AMBER MD package, for computational site-directed mutagenesis. This tool can mutate between any pair of amino acids in a computationally expedient, automated manner. To demonstrate the potential of this methodology, we have employed the protocol to investigate a test case involving stapled peptides, and have demonstrated good agreement with experiment.

Key words: molecular dynamics, *in silico* mutation

Introduction

Computational screens are a fundamental tool in drug discovery and development (Clark, 2008). It has become routine to use them to aid in the identification of compounds that show potential for use in medicine, and increasingly sophisticated techniques can make for great predictive ability (Cano *et al.*). The main advantage of these methods is obvious: the automation and parallelisation of the process allows to handle amounts of data that would be difficult to replicate with experimental techniques. As a result, they can be highly cost-effective, together with the potential for faster scanning of large virtual libraries. However, the speed and relative low cost of virtual screening methods are subordinate to their predictive power.

There has been considerable interest in developing virtual screens for systems which involve expensive (Masso and Vaisman, 2007) and time-consuming (Park *et al.*, 2012) techniques such as peptide, protein and antibody synthesis. Protein–protein interactions (PPIs) have been increasingly studied due to their central role in cellular regulation (Stites, 1997), but are notoriously problematic to analyse (Fry, 2006), because their binding surfaces are broad and relatively featureless. This makes it hard to determine which residues are important for the interaction (Kozakov *et al.*, 2011), and hotspot analysis is merely one tool in the arsenal. Development of a rapid and predictive computational method would therefore be a very powerful tool in drug discovery and development. There are computational examples where virtual mutagenesis has been used to analyse and direct the development of antibodies as drugs (Clark *et al.*, 2006; Sivasubramanian *et al.*, 2006; Azoitei *et al.*, 2014; Schwans *et al.*, 2014). Molecular dynamics (MD) is a widely applied tool for simulating proteins and can be used to extract information about structure and molecular interactions. Widely used computer packages for MD include AMBER (Salomon-Ferrer *et al.*, 2013a,b; Case *et al.*, 2015), GROMACS (Berendsen *et al.*, 1995), NAMD (Phillips *et al.*, 2005) and CHARMM (Brooks *et al.*, 2009), which can use different force fields and parameter sets. For our work, we have used the AMBER package,

which we have also previously applied successfully in the analysis of PPIs (Bradshaw *et al.*, 2010, 2012).

AMBER is currently limited to the simple mutations, as it only has in-built protocols for mutations to alanine and glycine. More general introduction of mutations presents complications due to the potential for steric clashes by residues of increased size. Software solutions that have been used successfully include the rotamer-based SWISS-MODEL (Guex and Peitsch, 1997) and the more sophisticated RosettaDesign (Liu and Kuhlman). In this paper, we report an alternative approach that generates mutations during the MD simulation itself. The advantage of such an approach is that it should allow more scope for local rearrangements to accompany the mutations, which in turn may make it more likely to generate realistic mutant structures that are also better starting points for further MD simulations.

Our efforts have produced what we describe as the 'Parasol Protocol', a dynamic method that is able to mutate between any pair of amino acids, although the methodology can be extended to non-natural residues as it is exemplified in this paper using stapled residues. The method is remarkably flexible in its execution and capable of a vast range of functional group interconversions, including rings, most common functional groups such alcohols, carboxylic acids, amines, amides and thiols, and the introduction of charged residues. The majority of the processes required for the 'Parasol Protocol' have been automated and optimised for rapid use with the AMBER MD package. In this paper, we report on our first application of the 'Parasol Protocol' in the PPI of the MCL-1/MCL-1 BH3 helix interaction (Stewart *et al.*, 2010).

Challenges in mutagenesis

There already exist several tools that are able to perform computational mutagenesis. In this section, we will outline some of these existing methods and explain how our approach differs from these.

In computational mutagenesis experiments, the single greatest challenge is to grow residues, via mutation from a small amino acid to a larger one. Going from large to small is inherently an easier process and therefore commonplace: alanine scanning mutagenesis, for example, is a well-established technique (Massova and Kollman, 1999). Any change that involves the addition of atoms and groups, however, is potentially problematic, since there may not be any physical space in which to add these new atoms if the interface is especially tight. How the environment adapts to accommodate the different side chains is extremely difficult to predict, and a variety of techniques has been developed in order to get around this problem. Figure 1 shows this issue in pictorial form.

Q2: No change

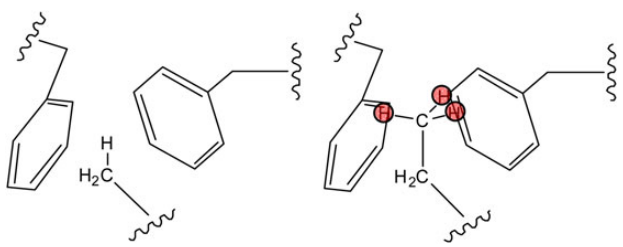


Fig. 1 Schematic depiction of the problem of mutating from a small amino acid to a larger one. On the left, the original residue has a methyl group that needs to be converted to an ethyl group. The starting structure does not allow this to be accommodated and simple replacement of atoms that would lead to structural instability.

Q9: No change

Q11: Black and White only

One of the simplest ways to obviate this problem is by using a library of rotamers for each side chain, and implementing an algorithm to determine which form of the residue can best fit, given the environment it is in. This is the approach used by software such as SwissPDBViewer (Guex and Peitsch, 1997) and PyMOL (Vaz *et al.*, 2010), and it has the advantage of being extremely quick; however, a significant limitation of this technique is that it only considers the structure of the mutated residue and does not allow for any other conformational changes that might occur in the surrounding amino acids. As it is not unreasonable to assume that a mutation will affect more than just the residue that is being mutated, further steps are required to predict the conformational behaviour of the amino acids around the point of mutation. To attempt to remove the restriction on surrounding residues, software such as TRITON (Damborský *et al.*, 2001) or SWISS-MODEL (Arnold *et al.*, 2006) employ homology modelling (Martí-Renom *et al.*, 2000), which models protein backbones and residues based on the structure of other proteins with comparable sequences. The assumption, not always correct (Fiser *et al.*, 2000), is that similar proteins or segments of proteins will fold in similar ways, thus giving us the ability to predict how a change in the sequence will translate in terms of conformation. Finally, RosettaDesign (Liu and Kuhlman) uses a combination of these two approaches, the rotamer library complemented by homology modelling.

While these single-point approaches are fast and simple to implement (Fischer *et al.*, 2011), they do present some limitations (Humphris and Kortemme, 2008), mostly because they focus principally on the residue to mutate rather than the environment around it. Predicting how the structure around the mutation is modified by the change is not straightforward, and the risk is that a false energy minimum might be generated, seemingly stable but which does not reflect the 'true' structure that would be reached when the same mutation is made experimentally. Nevertheless, because of their simplicity, these techniques remain popular and widely used.

Another possible approach to mutations involving growth is that offered by the thermodynamic integration (TI) and free energy perturbation (FEP) models (Chipot and Pearlman, 2002). TI and FEP are considered the gold standard in computational chemistry (Jensen, 2006), and they employ alchemical transformation (Yang *et al.*, 2004) to carry out mutagenesis. This technique uses the gradual manipulation of parameters such as mass and charge to convert from one functional group to another, effectively mutating the amino acid. It is called alchemical because the intermediate structures, with partial charges and half-formed bonds, are not chemical species; this evolving interpolation between the two states allows the system around the mutation to adapt to it and rearrange the environment's structure to respond to the change. This is considered the gold standard because it is exact with respect to statistical mechanics, but this comes at the cost of speed and needs significant computing power. TI and FEP are very expensive in terms of computational resources, and they are significantly limited in application due to the long timescales required to reach convergence; for this reason, they are rarely used in mutagenesis (Gouda *et al.*, 2003).

We have identified two of the many possible approaches to mutagenesis, both of which attempt to solve the problem when growing residues. However, these two strategies both have limitations and are not conducive to being fully automated in an algorithmic fashion. Therefore, we have chosen to develop a protocol that could unite the best qualities of the two techniques. The method is based on MD coupled with a gradual transformation of one amino acid to another. In this respect, it is also alchemical, as the intermediate states are not natural chemical species. It also allows bond rotations to

accommodate the structural effects of mutations, but does so via the inherent motions of MD rather than by the use of rotamer libraries.

Our efforts have produced the Parasol Protocol, a gradual method which takes advantage of the AMBER architecture and suite of programs to manipulate amino acids and proteins, leading to mutagenesis. The protocol is fully automated, fast and inexpensive to implement on modern computers, and it has been used with success on a variety of systems to mutate between all natural amino acids, as well as some exemplar non-natural residues. In the next sections, the workings of the protocol will be detailed, followed by a case study in which Parasol is used to replicate experimental data.

Methods

Work was carried out on a Linux machine with Fedora 12 and CentOS 6, equipped with two Intel® Dual-Core E5700 central processing unit (CPU) cores running at 3.00 GHz and a single GTX 680 graphical processing unit (GPU) running at 1.63 GHz. MD simulations and analysis were carried out with the AMBER 12 and AMBERTools 12 package (Case *et al.*, 2012), using the FF99SB force field (Lindorff-Larsen *et al.*, 2010).

Analysis and supporting scripts were written and run on the CPUs, leaving the bulk of the simulation to the GPU. This was done using PMEMD (Particle Mesh Ewald Molecular Dynamics) in its CUDA (Sanders and Kandrot, 2010; Salomon-Ferrer *et al.*, 2013a,b) (Compute Unified Device Architecture) package, which runs on GPUs. This is to take advantage of the architecture of GPUs, which, being far more parallelised, are much faster than CPUs (Anderson *et al.*, 2008; Xu *et al.*, 2010). This translates, we have found, to a 20-fold increase in speed in simulations.

Simulations were neutralised prior to running, using counterions (Cl^- in the case of positively charged systems, Na^+ for negatively charged ones), and solvated with TIP3P water molecules (Jorgensen *et al.*, 1983). The employed solvation shell was a truncated octahedral box in which no part of the system was closer than 8 Å from the edge. Periodic boundary conditions were used, with an 8 Å radius; inside the radius, non-bonded terms were calculated explicitly by the force field, and outside, the particle mesh Ewald method (Darden *et al.*, 1993) was used to describe interactions. The SHAKE algorithm (Ryckaert *et al.*, 1977) was employed to lengthen simulations by constraining bonds including hydrogen atoms, with the effect of lessening the computational load.

Simulations consist of three stages. The first is the minimisation stage, performed on the crystal structure, and used to remove inherent clashes and artefacts left over from crystallisation. It consists of a first minimisation (500 steps of steepest descent protocol) and a second minimisation (500 steps of conjugate gradient protocol). The second stage is the equilibration, which is itself divided into first and second equilibration. In the first equilibration, the temperature is raised from 0 to 300 K in the NVT ensemble, while the second equilibration is performed in the NPT ensemble at 1.0 bar of pressure. Once the system has been equilibrated and its temperature and pressure are as close as possible to biological values, the production run is carried out in the NPT ensemble. The temperature is controlled throughout with a Langevin thermostat (Loncharich *et al.*, 1992).

There are two sorts of simulations we run: mutations carried out with the Parasol Protocol and simulations. Mutations use for the first equilibration a 0.5 fs time step for 25 ps, while for the second equilibration, they use the same time step over 6.25 ps. As will be explained later, the mutation is divided into 11 short production runs each with a

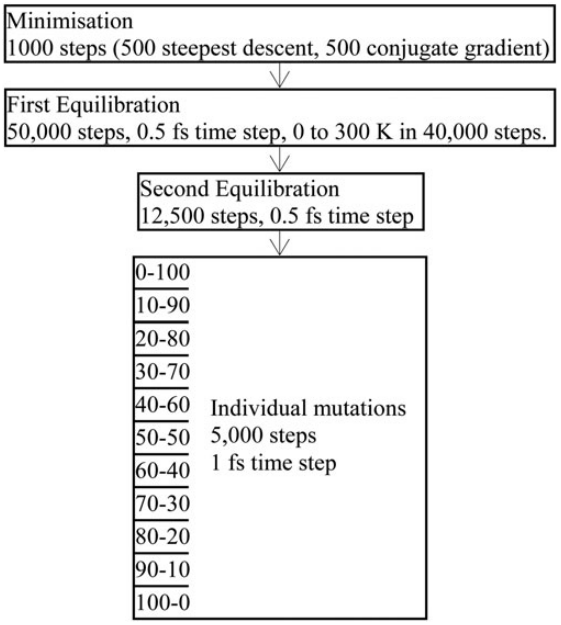


Fig. 2 A schematic representation of the mutation sequence.

1 fs time step lasting 5 ps, although we have experimented with different time step sizes. Figure 2 shows this schematically.

The simulations are run on native or mutated structures to generate average, equilibrated dynamics over a longer period of time and obtain MMGB/PBSA estimates of free energy. The first and second equilibrations both use a 2 fs time step over 250 ps, while the production run proper uses the same time step over 1 ns. Simulations are usually run in 10 copies for greater statistical significance.

Supporting manipulations are run with programs that are part of AMBERTools, and include LEaP, ptraj, ante-MMPBSA and parmed.py. Visual inspection of proteins and systems was carried out with VMD (Humphrey *et al.*, 1996). For GB calculations, the GB5 suggested parameters were used (mbondi2 radii, LCPO calculation of the SASA and 0.005 surface tension offset), while for PB calculations, the PB1 suggested parameters (mbondi radii, Molsurf calculation of the SASA and 0.0072 surface tension offset).

The Parasol Protocol

The underlying concept behind the Parasol Protocol is to allow a slow and gradual growth or removal of functional groups during an MD simulation, which allows the side chain to assume the best structure, as well as allowing the environment around it to adapt to the change. To this end, custom residues and atom types are used to finely control and manipulate all the parameters and properties of the mutated residue, which include the masses and charges of the atoms, the length and width of the bonds, angles and dihedrals, and the interactions between different parts of the amino acid. During the simulation, the parameters are slowly morphed from the starting values to the final ones, resulting in the final mutated state.

The scheme of the simplest mutation, the conversion of a hydrogen into a methyl group, is shown in Fig. 3 as it is implemented using the Parasol Protocol. Q3: No change

The methyl group is placed so that it occupies the same virtual space as the hydrogen from which it is to be grown, with the carbon halfway between carbon and hydrogen, and the hydrogens superimposed on the original hydrogen. At the beginning of the simulation, this methyl group

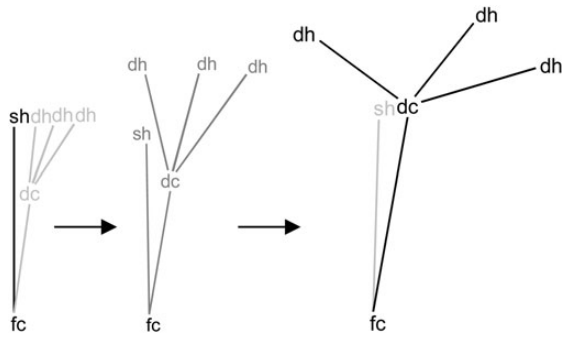


Fig. 3 A schematic representation of the Parasol Protocol, showing here the mutation from a hydrogen to a methyl group. (Left) The initial structure, with the hydrogen to be mutated (sh, special hydrogen) and the carbon it is attached to (fc, fixed carbon). The methyl to be grown is added by placing its carbon (dc, dummy carbon) halfway between fc and sh, and positioning the three hydrogens superimposed on sh. The colouring (black for 'realness', grey for varying degrees of 'non-existence') represents the gradual change of the parameters across the mutation. During the simulation, the methyl hydrogens 'open out' in a manner reminiscent to the way that a parasol opens, giving rise to the name of this protocol.

is 'non-existent', as shown by it being greyed out in Fig. 3. This means that the charges of its atoms are all 0, its interactions with any other atoms are set to be 0 and it has no ability to affect anything else in the simulation. It is present in the system, but nothing can 'see' it, and it can 'see' nothing. Furthermore, its bonds, angles and dihedrals are restrained, so that it occupies a small volume around the hydrogen, with bonds set at shorter distances (half a standard C–H length, 0.545 Å) and the angle set at 180°. In contrast, the parameters of the hydrogen, even if it has a custom atom type, are set to be as normal.

As the simulation progresses, the residue's parameters are modified in a stepwise fashion. We have found that increasing these in 10% increments is the optimal approach. So, after a first simulation where the distribution of parameter characteristics is 0%:100% (the methyl is 0% 'real', and the hydrogen is 100%), a second simulation is performed where the parameters are changed to 10:90: the charge of the methyl begins to increase, that of the hydrogen begins its descent to 0, the bonds and angles begin to lengthen and contract, and so on. After this, the 20:80 state is simulated, followed by the 30:70, the 40:60 and so on until the 100:0 is performed. In each step, the simulation is performed on the last set of coordinates obtained from the previous simulation. This gives a total of 11 short runs (5 ps for each) at the end of which the hydrogen has become 'non-existent', while the methyl is now 'real'. The motion of the methyl as it opens and takes its tetrahedral conformation resembles that of a parasol, which is what the protocol is named after.

This whole process is gradual and dynamic; no minimisation or equilibration is carried out between steps, thus enabling the system to adapt continuously to the mutation. Grown functional groups are allowed to rotate and move as freely as the system allows them to, so they can explore the conformational space in a purely dynamic way.

The success of this protocol in growing methyl groups led us to adapt and modify the strategy to grow and manipulate other functional groups, such that all those present in natural residues (like alcohols, amines, carboxylic acids, rings, etc.) can be handled. The protocol can add and remove any of these groups, and it can perform even more complicated and sophisticated mutations, like growing two groups at the same time, or the simultaneous growth and removal of groups. This is shown in Fig. 4.

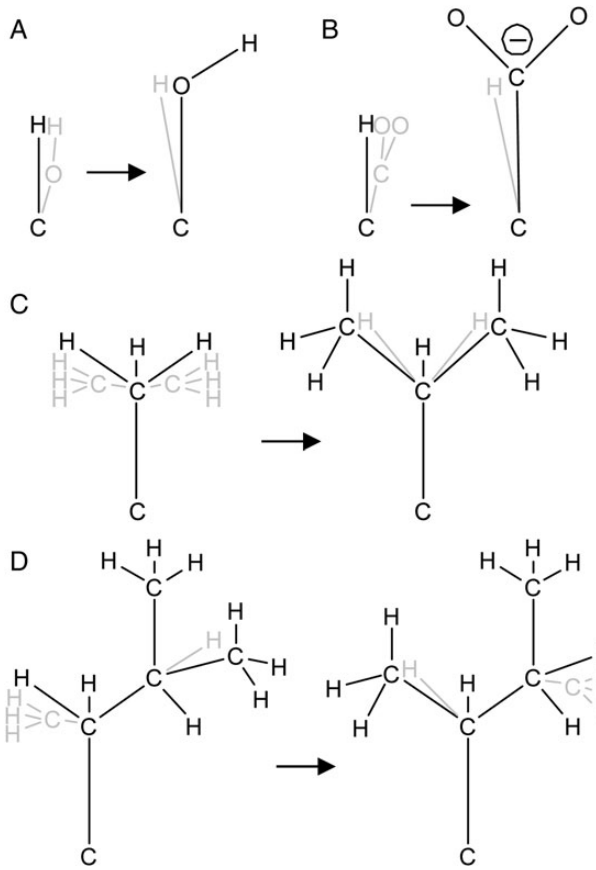


Fig. 4 A few representative mutations possible with the Parasol Protocol. The protocol can (**A**) grow functional groups such as alcohols, (**B**) grow charged groups, (**C**) perform two growths at the same time (such as two methyls in alanine to produce valine) or (**D**) grow and remove at the same time (such as leucine to isoleucine).

The approach to growing rings is slightly different, and shown in Fig. 5. Rather than adding atoms and groups at the end, these are added in between other atoms, so that they may grow sideways and generate ring-like structures. Our approach to generate aromatic residues is first to grow an intermediate hypothetical amino acid that has four carbon atoms and is a cyclobutadiene version of phenylalanine, which we term annuline. This has been parameterised for this purpose and the protocol uses this as a common intermediate for aromatic residues. This strategy of using intercalated carbons has also been used for other mutations, such as cysteine to homocysteine, or norleucine to arginine.

Another feature of the protocol is sequential mutations, which are carried out without removal of the solvation shell for greater continuity between the two systems. Because the protocol cannot create something out of nothing, some mutations need to be done in multiple steps. For example, alanine cannot be mutated to isoleucine in one step, and valine must be used as an intermediate. The protocol is set, so that it keeps the same solvation shell throughout, also avoiding superfluous minimisations and equilibrations.

The protocol has been automated for maximum efficiency and speed. The generation of the input files and the carrying out of the simulations is controlled by a series of scripts that run automatically, which leads to a very streamlined, rapid process. In an average sized system of ~20 000 atoms consisting of protein and solvation shell, a

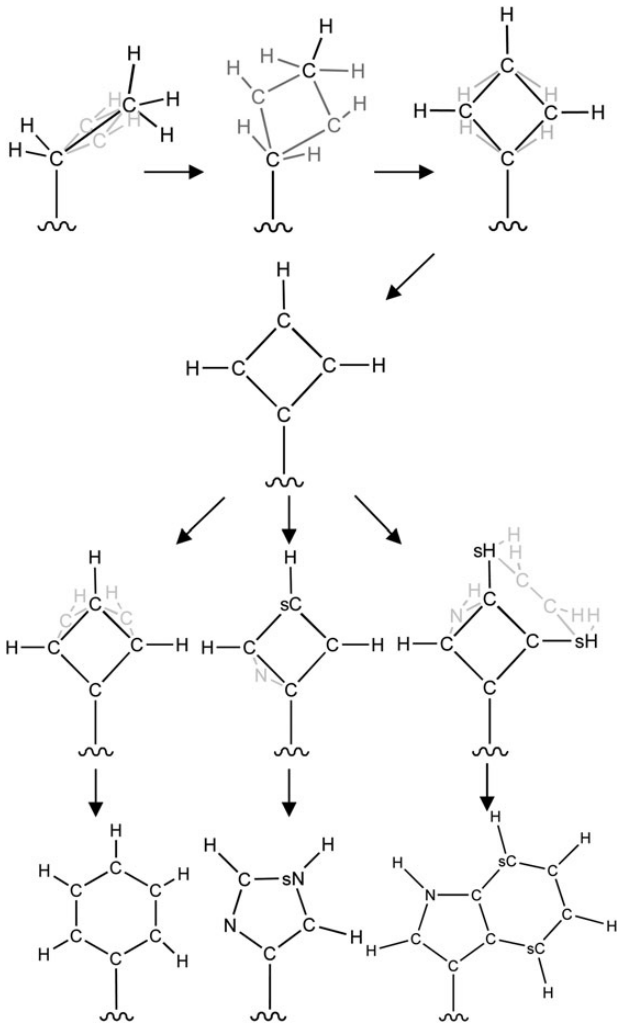


Fig. 5 The intercalated carbon method used to make rings. Rather than adding the carbons at the end of chains, adding them between other carbons leads to ring formation. In this example, the ethyl-like group from norvaline is made into a four-membered ring, generating the artificial annuline, used as a precursor to the other aromatic residues (mutations to phenylalanine, histidine and tryptophan shown).

one-step mutation will take ~10 min, and subsequent mutations on the same system a further 5 min each; larger systems will require more time. This great speed is due to the protocol relying on PMEMD.CUDA (Götz *et al.*, 2012), which allows for a drastic improvement in performance.

Case study: growing stapled peptides

The Parasol Protocol works efficiently and consistently, and it can be used to manipulate proteins and peptides by mutating residues. It differs from existing methods in that it allows for local conformational rearrangements to be accommodated automatically and without the need for manual intervention. The question we need to answer at this point is whether the mutated structures it produces can be used to predict trends and direct workflow. Ideally, it should generate mutated structures whose properties can be computationally compared with those of the native. To test this, we have applied the method to an investigation of the effect of mutations on the strength of PPIs. Specifically, we have computed the interaction energy between a protein receptor and another protein or peptide ligand, using the MMPBSA (Lee *et al.*, 2000) and the related MMGBSA (Gohlke and Case, 2004) protocols. Where experimental data are available, this allows direct evaluation of the predictive power of the method, by calculating the energies for both native and mutant structures, and comparing with the value obtained in the laboratory.

There are a vast number of systems that could be examined in this way, but to exemplify the power of our new protocol, we have chosen to examine the effect of a set of mutations that would be very difficult to derive by alternative methods. Stapled peptides, which involve a covalent tether between amino acids distant in the sequence, have been the subject of many recent studies as they offer the advantage of conformationally fixing small peptides into a bioactive conformation (Verdine and Hilinski, 2012; Chang *et al.*, 2013; Lau *et al.*, 2014). Generating a stapled structure involves not only mutating two residues to half-staples, there is also the need to covalently bridge these two halves while producing energetically feasible structures. The adaptive dynamic approach of the Parasol Protocol can easily be used to grow even such demanding mutations, providing a suitable and unique test. The full protocol as implemented for growing staples is shown in Fig. 7. The first step is to place two norleucine residues (straight-chain version of leucine) in *i*, *i* + 4 positions where the staple is to be added. This is done by successive mutagenesis of the starting amino acids. Once two norleucine residues have been created, a tether is formed between the two ends. This bond has no force constant at the beginning of the simulation, meaning that it does not have the ability to affect the dynamics of the norleucines, which are able to flop around and move as freely as they can. However, as the run progresses, the bond gains strength, meaning that it forces the two carbons to come together, while its equilibrium length, fixed at an arbitrarily large value at the beginning of the simulation to allow the norleucine side chains as much freedom as possible, begins to decrease until it reaches the value of a C–C double bond. The structure is constrained so that it adopts a *cis* geometry, as this is the conformation found in the system described below. At the same time, two methyl groups are grown onto the α-carbon to match the staple used in experimental conditions.

As with the Parasol Protocol, graduality is the key; by slowly increasing the force of the bond from 0 to the actual value for a C–C bond and at the same time reducing the bond distance from large to the correct length, the system is allowed to adapt to the change and incorporate the staple in the new structure. The stapler tool uses the same minimisation, equilibration and production parameters as those described above for the mutation of amino acids. It should be noted that this staple is by no means the only one that could be created and with the correct use of atoms and parameters, any sort of staple can be introduced.

The specific system that we investigated was that of the MCL-1/MCL-1 BH3 helix interaction (Stewart *et al.*, 2010) (PDB code: 3MK8). MCL-1 is part of the BCL-2 protein family, the members of which possess anti-apoptotic properties. Some cancers exploit this ability to propagate uncontrollably, and therefore, there is considerable interest in finding ways of preventing MCL−1's action. This has led the authors of the original work to discover an inhibitory peptide in the MCL-1 BH3 α-helix, which specifically and selectively binds to MCL-1. The interaction is depicted in Fig. 6. As the binding is dependent upon BH3 having the correct helical conformation, the authors have applied a chemical staple to enforce the structure. The staple in this example is a relatively simple aliphatic chain built with residues in the *i*, *i* + 4 positions, with a double bond in the middle and a methyl in place of the α-hydrogen. It is made from non-natural

Q4

Q4: Not sure what the praxi is in thi case: fig 7 is cite first but in the layout it comes later. I would defer to your superior expertise

amino acids incorporating olefin tethers (Schafmeister *et al.*, 2000) linked via ruthenium-catalysed ring-closing metathesis (Blackwell *et al.*, 2001). It is chemically possible to position the staple between any *i*, *i* + 4 pair of residues in the sequence, but the question as to which position to select is not a trivial one, as two amino acids have to be sacrificed to incorporate the ligation. It is therefore fundamental to ensure that activity persists after the staple is added and avoid removing residues that are important for specificity or interaction.

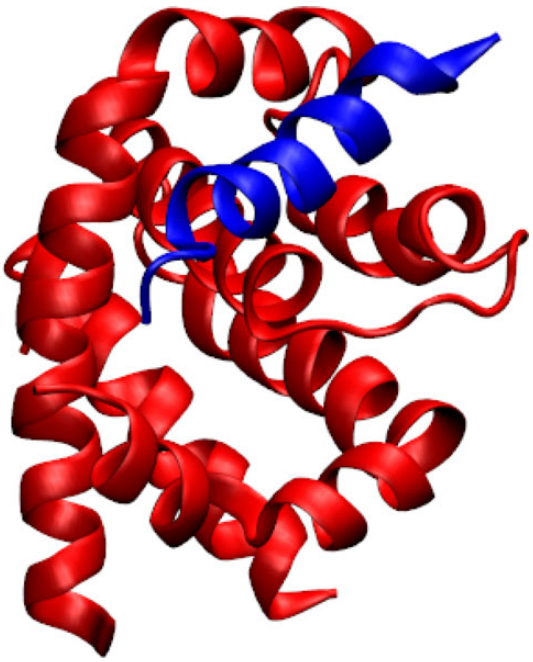


Fig. 6 The analysed system, with MCL-1 in red and the BH3 helical peptide in blue. As the peptide needs that conformation to interact, a staple can be added to it to ensure the proper structure in retained in solution and so increase potency.

The authors describe a series of five staple position variants, which were arrived at after performing alanine-scanning mutagenesis to identify the best residues to replace. These results are shown in Table I and illustrate clearly that not all positions are equally viable. The synthesis of stapled peptides is technically demanding and the authors did not attempt to produce each possible *i*, *i* + 4 staple.

To see how computational methods might have expedited this study, we have taken the crystal structure provided by the authors and used the Parasol Protocol to recreate the stapled peptides that were used in the original study (Fig. 7). After this, the interaction between the peptide and the protein was calculated with the MMPBSA and MMGBSA protocols. For such a procedure to be successful requires two factors to be in place. First, it should be possible to generate suitable mutant structures for the system under study. Secondly, the technique used to calculate the energetics should be robust enough to provide correct values. The energetic calculations used are well established and their advantages and limitations understood (Gohlke and Case, 2004). There is therefore just the question of how well the grown staple structures reproduce the experimental observations.

In each case studied, it was found possible to grow a staple that resulted in a plausible, energetically stable protein–ligand complex. The resulting structures were then used to predict the binding energetics. It was found that the computational approach generated predictions that, in broad terms, give good agreement with those that are experimentally derived. It must be stressed that the nature of the

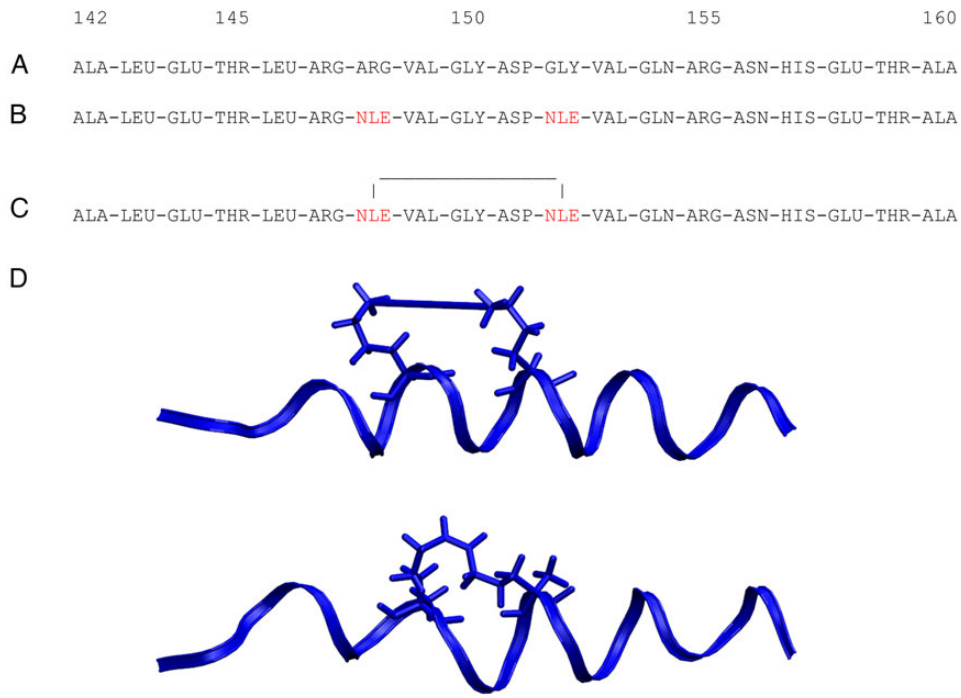


Fig. 7 The stapling procedure implemented within the Parasol Protocol. The native peptide (**A**) is taken, and two residues in *i*, *i* + 4 positions are mutated to norleucines (**B**). These norleucines are then tethered with a loose bond with arbitrarily large length and 0 force constant (**C**). As the simulation progresses, the tether brings the two ends together and the staple is formed (**D**). Note the *cis* structure of the double bond in the final product, as well as the methyl grown on the α-carbon. The numbering and sequence of the peptide are different from the original used by Stewart *et al.* (the terminal lysine and phenylalanine are missing) because they were not resolved in the crystal structure.

Table I. Full peptide scan of the helix, varying the staple across every possible position, and comparison with the experimental data of Stewart *et al.* (Stewart *et al.*, 2010)

Identifier	Sequence	Experiment K_D (nM)	Rank	GB (kcal/mol)	Rank	PB (kcal/mol)	Rank
Native	ALETLRRVGDGVQRNHETA	245 ± 29	N/A	−49.7 ± 6.0		−68.4 ± 6.2	
143–147	AXETLXRVGDGVQRNHETA	No experimental data		−43.5 ± 6.0	10	−59.5 ± 6.1	10
144–148	ALXTLRXVGDGVQRNHETA	18 ± 4	2	−54.1 ± 5.4	5 (3)	−71.3 ± 5.4	5 (3)
145–149	ALEXLRRXGDGVQRNHETA	No experimental data		−55.2 ± 5.5	3	−71.9 ± 5.5	4
146–150	ALETXRRVXDGVQRNHETA	No experimental data		−42.4 ± 5.0	11	−57.7 ± 5.3	11
147–151	ALETLXRVGXGVQRNHETA	No experimental data		−26.9 ± 4.8	14	−43.2 ± 5.0	14
148–152	ALETLRXVGDXVQRNHETA	43 ± 16	4	−49.8 ± 6.1	8 (4)	−66.5 ± 6.3	8 (4)
149–153	ALETLRRXGDGXQRNHETA	No experimental data		−53.8 ± 5.4	6	−68.6 ± 6.1	7
150–154	ALETLRRVXDGVXRNHETA	>1000	5	−33.4 ± 5.1	12 (5)	−49.3 ± 5.6	12 (5)
151–155	ALETLRRVGXGVQXNHETA	No experimental data		−31.6 ± 5.6	13	−47.8 ± 5.8	13
152–156	ALETLRRVGDXVQRXHETA	No experimental data		−51.0 ± 5.3	7	−70.9 ± 5.6	6
153–157	ALETLRRVGDGXQRNXETA	No experimental data		−59.2 ± 5.5	1	−77.1 ± 5.3	1
154–158	ALETLRRVGDGVXRNHXTA	10 ± 3	1	−56.8 ± 5.3	2 (1)	−74.4 ± 5.3	3 (2)
155–159	ALETLRRVGDGVQXNHEXA	No experimental data		−45.5 ± 6.0	9	−63.2 ± 6.1	9
156–160	ALETLRRVGDGVQRXHETX	33 ± 10	3	−54.6 ± 6.6	4 (2)	−76.9 ± 6.6	2 (1)

Staples with experimental data are highlighted in grey. The columns on the right rank the peptides by activity, with red as the worst, orange as the bottom third, yellow the middle, green the top and light green the best. The numbers in the parentheses refer to internal ranking between peptides with experimental data.

computational energy calculations means that we do not expect calculated energies to relate to binding energy directly; rather, we are looking for gross differences in rank order to be reflected. In this particular case, peptide 150–154 binds far weaker than the others and this is fully predicted by the calculations. The four other stapled peptides all bind well, with 154–158 being the most potent. In energy terms, there is not much difference between them, with only a 4-fold difference in K_D (<1 kcal mol^{-1}). While we would not necessarily expect the computational energy calculations to be precise enough to differentiate between these, the predicted rank order does indeed place them in roughly the correct order, with the best peptide determined experimentally being predicted correctly by the calculations (best in GB and second best in PB). Both the GB and PB calculation procedures produce very similar predicted rank orders and either seems appropriate to use.

Figure 8 plots the correlation between the energy derived from the calculation, using the GB method, and that from the experimental K_D values. There is clear demarcation between the poorly binding 150–154 staple and the others, and overall excellent correlation. This result is very pleasing as it suggests that computational simulation can offer an excellent guide to the experimental design. In this particular instance, stapled peptides are very difficult to synthesise and so methods that reduce time and costs have clear benefit.

The original publication described the synthesis and testing of five of the stapled peptides discussed above. However, there are a total of 14 positions within the sequence where a staple can be introduced. Creating these synthetically would be a considerable challenge, but computationally, it is easy to perform a more complete staple scan of the whole sequence. This computational stapling procedure was carried out for all possible positions on the peptide. The results are also shown in Table I together with the available experimental results. It should be noted that an inherent factor in the methods used to derive energetics from the MMPB(GB)SA (Chipot and Pohorille, 2007) MD simulations is that the values obtained are of relatively low precision. The energy is computed as a thermodynamic cycle, shown in Fig. 9; the difference in energy between the complex and the lone receptor and ligand, however, does not only arise from the interactions between the two, but also from random motions in the system, such as loops unfolding and chains fluctuating, which may have large contributions

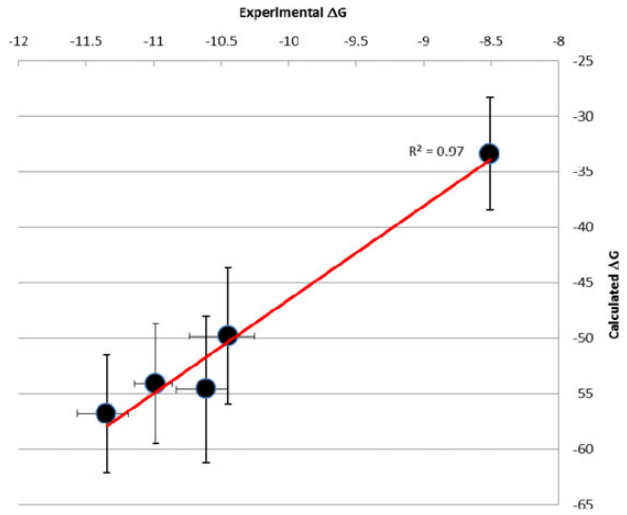


Fig. 8 Correlation of experimental DG values with those calculated using the GB protocol following growth of staples using the Parasol Protocol.

that may not affect the active site at all. If the difference in energy is small, it may be overshadowed by these other effects, and thus it may be hard to compare the interaction energies if they are particularly close.

When interpreting these data, it is therefore important not to place too much emphasis on small differences but where the differences are large then these can be significant. For example, the interaction energy differences between the weakest binding 150–154 peptide and the strongest binding peptide 154–158 demonstrate the predictive power of such calculations.

Our work can be compared with a previous study (Joseph *et al.*, 2012) which had the same aim of replicating the experimental results and observations of that particular system. Our own data compare favourably and improve upon those finds. Both methods correctly predict the worst inhibitor, but our new method can also correctly predict the second-worst inhibitor, which is not as adequately described in the previous study.

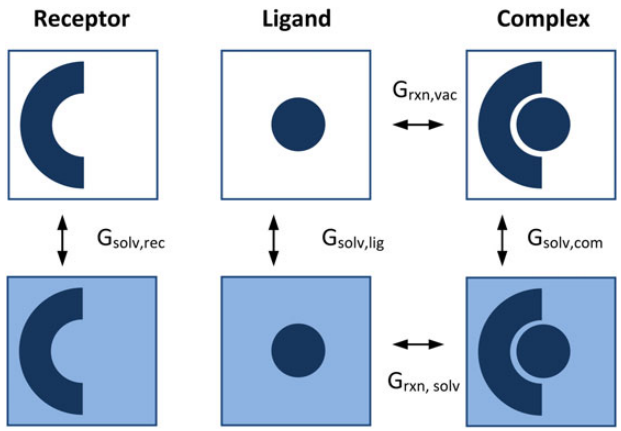


Fig. 9 The thermodynamic cycle used to calculate the interaction energy in solution, $\Delta G_{rxn,solv}$. Because it is hard to compute directly (Stewart *et al.*, 2010), the cycle is calculated instead, taking the energies of solvation of the single components (receptor, rec, ligand, lig, and complex, com) and calculating the *in vacuo* interaction energy.

Q5: Change to table I; there is no table II.

Q5 Table II includes a series of predictions for which there are no experimental data. It is interesting to note that the peptides that were selected for synthesis include neither the predicted worst nor the predicted best peptides. We must stress that the calculations are only suggested to be a *guide* to experimental design, but it is interesting to speculate how we would have used these data if they had been available. On the basis of these results, we would certainly have advocated that the untested peptide 153–157 be tried as this is a possible improvement on the best of the rest. Peptides 143–147, 146–150, 147–151, 150–154 and 151–155 result in calculated energies that are significantly lower than the best examples and so could safely be omitted.

The non-stapled native peptide was also tested for comparison. While the best of the stapled versions do show lower energies that are in part consistent with their improved binding, it is not appropriate to consider that this is the sole factor responsible. The computational calculation does not take into account the entropic contributions to the interaction, as the peptide is modelled already in its pocket and with the correct conformation. Under experimental conditions, the interaction also depends on the equilibrium between the bound helical structure and random coil, which these simulations do not replicate. The reason that stapling is advantageous is that it helps stabilise and pre-form the correct secondary structure required for appropriate docking onto the target. For this reason, any comparison of stapled versus non-stapled forms is unlikely to give a complete description. For comparisons of the stapled versions to be valid, we implicitly assume that each of these has an equivalent propensity to result in an organised α-helical structure.

Conclusion

We have described here a novel protocol for computational site-directed mutagenesis, the Parasol Protocol. The method differs from existing procedures by being fully dynamic, allowing the environment to adapt to the mutation. As it does not rely on reference to existing structures, it is possible to incorporate quite complex mutations, which we have exemplified here by introducing a covalent staple within a polypeptide chain. In computational terms, the method, which we have implemented within the molecular mechanics package AMBER, is fast, relatively cheap and is able to mutate between all natural amino acids. Owing to the flexibility of the protocol, it is relatively easy to extend the methodology and so allow the introduction of new structures such as staples and non-natural amino acids.

In this paper, we show how this procedure, applied to the introduction of stapled peptides, can be used in combination with prediction of binding energy, in order to give broad agreement with experimental results. We have also shown how it is simple and rapid to predict the effect of introducing a far greater range of mutations than would be feasible experimentally, which would be of significant benefit to studies involving the design of more potential inhibitors.

Further work can focus on staples of a different chemical nature, featuring perhaps different functional groups, and of varying length (i, $i+3$, i, $i+5$, i, $i+7$) applied in other systems. Work to validate the Parasol Protocol and make it more automated and accessible is well underway.

Q10: EPSRC: Engineering and Physical Sciences Research Council

Acknowledgements

We thank Imperial College and the Institute of Chemical Biology for support.

Funding

We acknowledge EPSRC and MedImmune for funding. Q10

Q6: Azoitei: The original link is broken, here's a full citation: Proteins. 2014 Oct 82(10): 2770-278

References

Anderson,J.A., Lorenz,C.D. and Travesset,A. (2008) *J. Comput. Phys.*, **227**, 5342–5359.

Arnold,K., Bordoli,L., Kopp,J. and Schwede,T. (2006) *Bioinformatics*, **22**, 195–201.

Azoitei,M.L., Ban,Y.A., Kalyuzhny,O., Guenaga,J., Schroeter,A., Porter,J., Wyatt,R. and Schief,W.R. (2014) *Proteins Struct. Funct. Bioinf.* Q6

Berendsen,H.J.C., van der Spoel,D. and van Drunen,R. (1995) *Comput. Phys. Commun.*, **91**, 43–56.

Blackwell,H.E., Sadowsky,J.D., Howard,R.J., Sampson,J.N., Chao,J.A., Steinmetz,W.E., O'Leary,D.J. and Grubbs,R.H. (2001) *J. Org. Chem.*, **66**, 5291–5302.

Bradshaw,R.T., Patel,B.H., Tate,E.W., Leatherbarrow,R.J. and Gould,I.R. (2010) *Protein Eng. Des. Sel.*

Q6: Bradshaw: Volume 24, Issue 1-2, Pp 197-207.

Bradshaw,R.T., Aronica,P.G.A., Tate,E.W., Leatherbarrow,R.J. and Gould,I.R. (2012) *Chem. Sci.*, **3**, 1503–1511.

Brooks,B.R., Brooks,C.L., III, Mackerell,A.D., Jr, *et al.* (2009) *J. Comput. Chem.*, **30**, 1545–1614.

Cano,G., García-Rodríguez,J. and Pérez-Sánchez,H. *Lett. Drug Des. Discov.*, **11**, 33–39.

Q7: Cano: 2014 Q7

Case,D.A., Darden,T.A., Cheatham,T.E., III, *et al.* (2012) University of California, San Fransico, **1**.

Case,D., Berryman,J.T., Betz,R.M., *et al.* (2015) *AMBER 2015*. University of California, San Francisco.

Chang,Y.S., Graves,B., Guerlavais,V., *et al.* (2013) *Proc. Natl Acad. Sci. USA*, **110**, E3445–E3454.

Chipot,C. and Pearlman,D.A. (2002) *Mol. Simul.*, **28**, 1–12.

Q8: Chipot: New York

Chipot,C. and Pohorille,A. (2007) *Free Energy Calculations: Theory and Applications in Chemistry and Biology*. Springer. Q8

Clark,D.E. (2008) *Expert Opin. Drug Discov.*, **3**, 841–851.

Clark,L.A., Boriack-Sjodin,P.A., Eldredge,J., *et al.* (2006) *Protein Sci.*, **15**, 949–960.

Damborský,J., Prokop,M. and Koča,J. (2001) *Trends Biochem. Sci.*, **26**, 71–73.

Darden,T., York,D. and Pedersen,L. (1993) *J. Chem. Phys.*, **98**, 10089–10092.

Fischer,A., Seitz,T., Lochner,A., Sterner,R., Merkl,R. and Bocola,M. (2011) *ChemBioChem*, **12**, 1544–1550.

Fiser,A., Do,R.K.G. and Šali,A. (2000) *Protein Sci.*, **9**, 1753–1773.

Fry,D.C. (2006) *Pept. Sci.*, **84**, 535–552.

Gohlke,H. and Case,D.A. (2004) *J. Comput. Chem.*, **25**, 238–250.

Gouda,H., Kuntz,I.D., Case,D.A. and Kollman,P.A. (2003) *Biopolymers*, **68**, 16–34.
Götz,A.W., Williamson,M.J., Xu,D., Poole,D., Le Grand,S. and Walker,R.C. (2012) *J. Chem. Theory Comput.*, **8**, 1542–1555.
Guex,N. and Peitsch,M.C. (1997) *Electrophoresis*, **18**, 2714–2723.
Humphrey,W., Dalke,A. and Schulten,K. (1996) *J. Mol. Graph.*, **14**, 33–38.
Humphris,E.L. and Kortemme,T. (2008) *Structure*, **16**, 1777–1788.
Jensen,F. (2006) *Introduction to Computational Chemistry*. John Wiley & Sons, Inc.
Jorgensen,W.L., Chandrasekhar,J., Madura,J.D., Impey,R.W. and Klein,M.L. (1983) *J. Chem. Phys.*, **79**, 926–935.
Joseph,T.L., Lane,D.P. and Verma,C.S. (2012) *PLoS One*, **7**, e43985.
Kozakov,D., Hall,D.R., Chuang,G.-Y., *et al.* (2011) *Proc. Natl Acad. Sci. USA*, **108**, 13528–13533.
Lau,Y.H., de Andrade,P., Skold,N., McKenzie,G.J., Venkitaraman,A.R., Verma,C., Lane,D.P. and Spring,D.R. (2014) *Org. Biomol. Chem.*, **12**, 4074–4077.
Lee,M.R., Duan,Y. and Kollman,P.A. (2000) *Proteins Struct. Funct. Bioinf.*, **39**, 309–316.
Lindorff-Larsen,K., Piana,S., Palmo,K., Maragakis,P., Klepeis,J.L., Dror,R.O. and Shaw,D.E. (2010) *Proteins Struct. Funct. Bioinf.*, **78**, 1950–1958.
Liu,Y. and Kuhlman,B. *Nucleic Acids Res.*, **34**, W235–W238.
Loncharich,R.J., Brooks,B.R. and Pastor,R.W. (1992) *Biopolymers*, **32**, 523–535.
Martí-Renom,M.A., Stuart,A.C., Fiser,A., Sánchez,R., Melo,F. and Šali,A. (2000) *Annu. Rev. Biophys. Biomol. Struct.*, **29**, 291–325.
Masso,M. and Vaisman,I.I. (2007) *Bioinformatics*, **23**, 3155–3161.
Massova,I. and Kollman,P.A. (1999) *J. Am. Chem. Soc.*, **121**, 8133–8143.
Park,D., Hou,X., Sweedler,J.V. and Taghert,P.H. (2012) *Peptides.*, **36**, 251–256.
Phillips,J.C., Braun,R., Wang,W., *et al.* (2005) *J. Comput. Chem.*, **26**, 1781–1802.
Ryckaert,J.-P., Ciccotti,G. and Berendsen,H.J.C. (1977) *J. Comput. Phys.*, **23**, 327–341.
Salomon-Ferrer,R., Case,D.A. and Walker,R.C. (2013a) *WIRES Comput. Mol. Sci.*, **3**, 198–210.
Salomon-Ferrer,R., Götz,A.W., Poole,D., Le Grand,S. and Walker,R.C. (2013b) *J. Chem. Theory Comput.*, **9**, 3878–3888.
Sanders,J. and Kandrot,E. (2010) *CUDA by Example: An Introduction to General-Purpose CPU Programming*. Addison Wesley.
Schafmeister,C.E., Po,J. and Verdine,G.L. (2000) *J. Am. Chem. Soc.*, **122**, 5891–5892.
Schwans,J.P., Hanoian,P., Lengerich,B.J., Sunden,F., Gonzalez,A., Tsai,Y., Hammes-Schiffer,S. and Herschlag,D. (2014) *Biochemistry*, **53**, 2541–2555.
Sivasubramanian,A., Chao,G., Pressler,H.M., Wittrup,K.D. and Gray,J.J. (2006) *Structure*, **14**, 401–414.
Stewart,M.L., Fire,E., Keating,A.E. and Walensky,L.D. (2010) *Nat. Chem. Biol.*, **6**, 595–601.
Vaz,F., Hanenberg,H., Schuster,B., *et al.* (2010) *Nat. Genet.*, **42**, 406–409.
Verdine,G.L. and Hilinski,G.J. (2012) *Methods Enzymol.*, **503**, 3–33.
Xu,D., Williamson,M.J. and Walker,R.C. (2010) *Ann. Rep. Com. Chem.*, **6**, 2–19.
Yang,W., Bitetti-Putzer,R. and Karplus,M. (2004) *J. Chem. Phys.*, **120**, 9450–9453.

Q8: Jensen: Chichester, UK.

Q7: Liu: 2006

Q8: Sanders: Upper Saddle River, NJ

Figure 1.

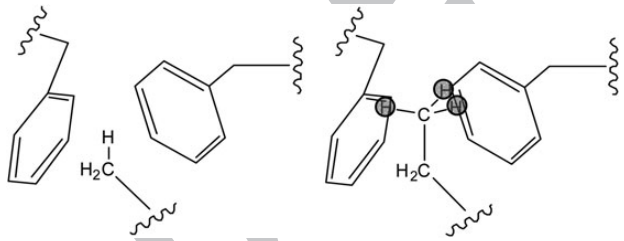


Figure 6.

Figure 7.

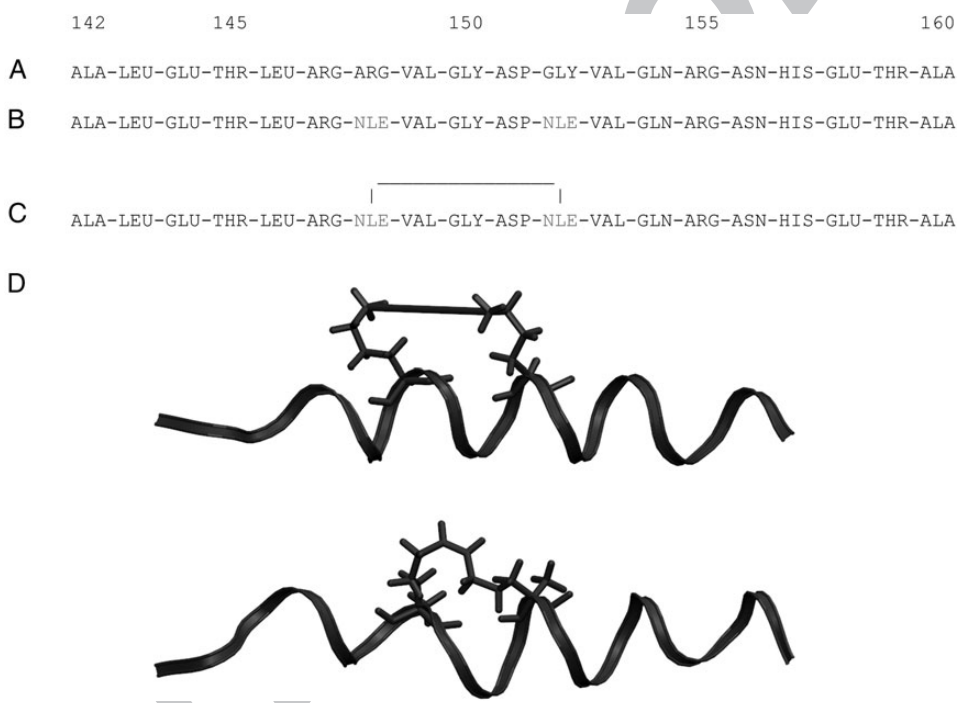


Figure 8.

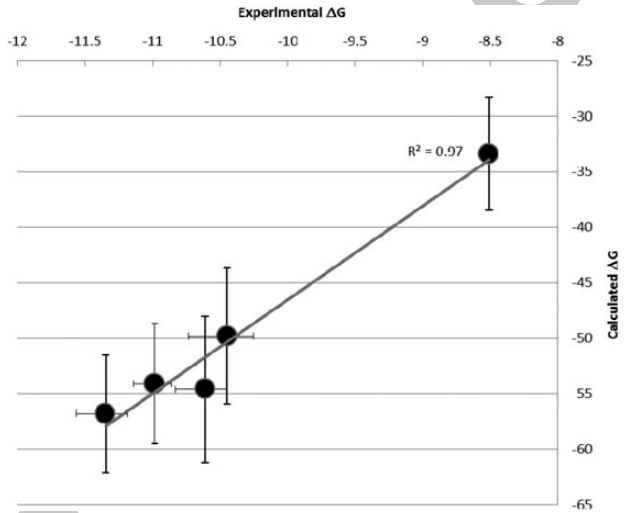


Figure 9.