User Guide

Provar (v4.81) calculations using MATLAB

Introduction

“Provar” (Probability of variation) is a method for probabalistic scoring of pocket predictions across large sets of related protein structures. These scores are the overall probabilities of particular atoms or residues being found to be lining a pocket in the set of structures. Scores output in PDB format files can be readily visualized through simple colour-coding atoms or residues.

The approach can help compare pockets across multiple conformations of a protein such as those generated by tCONCOORD[1] or any other simulation method. By supplying a suitable sequence alignment it is also possible to visualize pocket conservation across a set of homologous structures. We enable the input of results from several pocket programs, which allows for comparison between those programs.

The Provar algorithm is implemented using MATLAB. The program takes a set of structures and paired pocket predictions and assesses pocket-lining atoms and residues for each structure to generate a set of probability values, which are then written to a reference structure. The Provar algorithm fits within an overall workflow indicated in Figure 1.

This document describes how to use the supplied MATLAB modules and test data sets to generate PDB files that have been updated with Provar pocket-lining probability values. It also indicates how these can be usefully visualized.

The ZIP archive contains the MATLAB source code required to run the Provar algorithm and two example data sets:

Example 1 consists of 250 tCONCOORD simulated conformers of *apo* Bcl-2, alongside the matched PASS[2] pocket predictions.

Example 2 consists of 17 *apo* members of the IL-2 superfamily as determined by CATH (1.20.1250.10) and an alignment generated by Mustang[3].

Pre-requisites

All of the modules are written in MATLAB and require a licensed MATLAB installation including the Statistics Toolbox. This software has been tested on ‘Student Version 7.4.0 (R2007a)’ running under Mac OS X Leopard (10.5). Some familiarity with use of the MATLAB interface is required in order to load modules, amend parameters as required and run the software.

Reference

The following submitted paper covers details of the Provar method, background and rationale, examples and the results of various tests. The two supplied examples recapitulate data used to generate two figures used in the paper: Figure 8B (example 1) and Figure 9 (example 2). Please cite this reference where possible

Paul Ashford, David S Moss, Alexander Alex, Siew K Yeap, Alice Povia, Irene Nobeli and Mark A Williams: Visualisation of variable binding pockets on protein surfaces by probabilistic analysis of related structure sets (submitted).

Note – platform independence

Although the MATLAB code uses platform independent constructs - and should therefore run under Windows operating systems - we have not yet been able to test this thoroughly. Please let us know if you experience any issues!

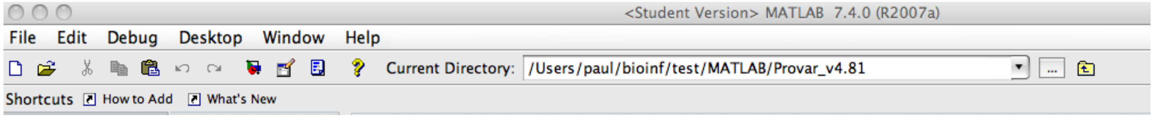
Extracting files, MATLAB working directory and Provar parameters

Extract the zip files

Choose a suitable directory in which to extract the supplied archive “Provar_v4.81.zip”. This will create a subdirectory “Provar_v.481” containing all of the MATLAB modules “*.m” and two subdirectories (“example1_conformers” and “example2_homologues”) containing all of the test data and the required parameter files.

Set the MATLAB current directory

Provar will take this to be the root directory and expect data to be in sub-folders under this. This can easily be set using the MATLAB GUI, as illustrated.



<Student Version> MATLAB 7.4.0 (R2007a)

Current Directory: /Users/paul/bioinf/test/MATLAB/Provar_v4.81

Choose/create a suitable Provar parameter file

The parameter file contains essential settings indicating where the PDB structures are found, which pocket program data to use and where the pocket prediction files are, amongst others.

Each of the examples comes with a suitable parameter file with which to run it. This file is called “fn_Provar_params.m” and a suitable version for each of the examples is found in the respective folder. These files are (helpfully) commented for you to amend to run your own datasets. For the purposes of this test you can simply use one of the two supplied files (see Examples 1 and 2 below).

Run modes – conformations or homologues?

Provar can run in two main modes:

Conformers

The structures are generated using tCONCOORD or from Molecular Dynamics snapshots based on a single starting PDB chain. All PDB structures and the “reference” structure must have identically numbered atoms and residues. No structure or sequence alignment is necessary. Conformer mode expects numbered structure and pocket prediction pairs and produces probabilities at both the atom and amino-acid level.

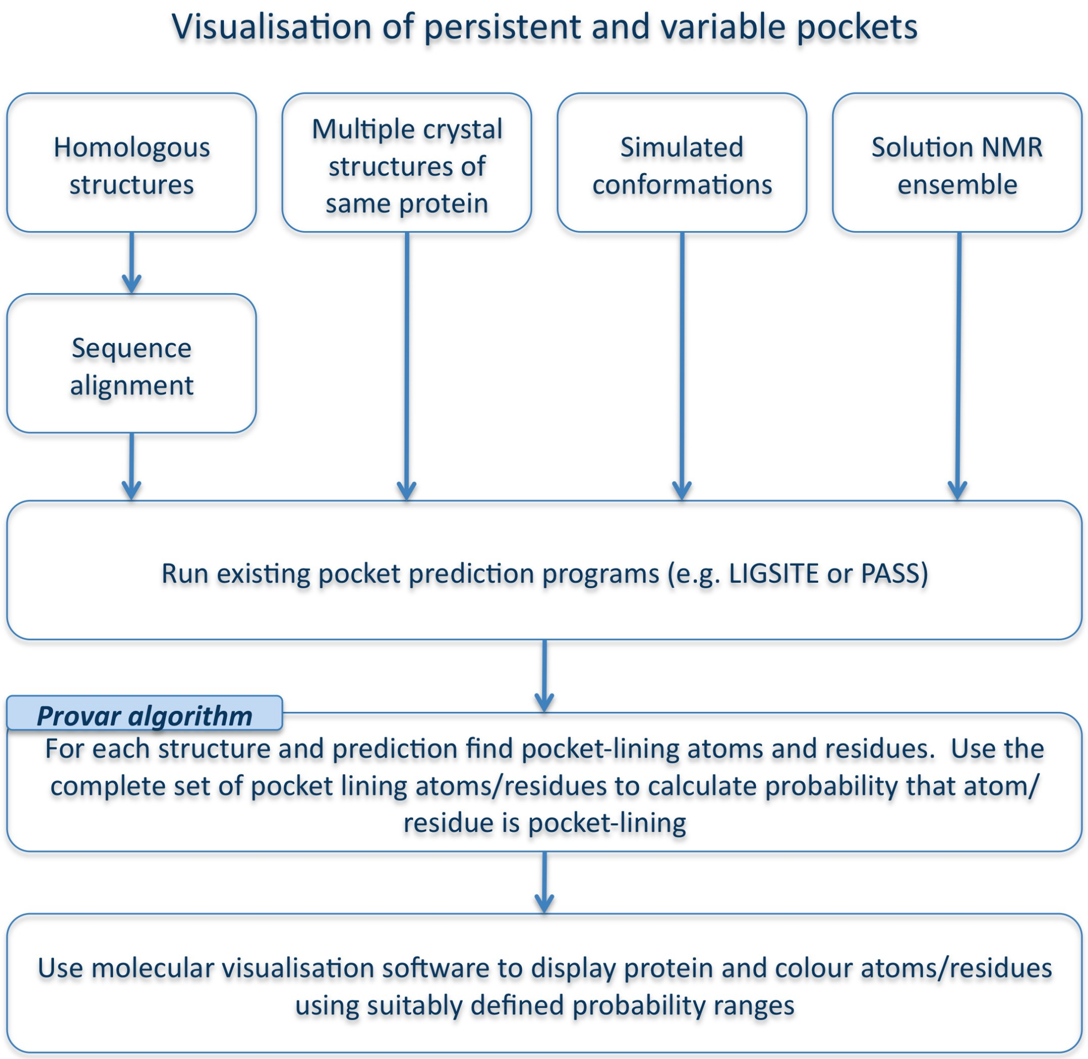
Homologues

A set of structures is provided, along with similarly named pocket predictions and a sequence alignment file in FASTA format.

Homologue mode can only produce amino-acid level outputs. Note that although the alignment file is purely sequence based, it usually best generated through structure-based sequence alignment algorithms such as Mustang.

The other difference between the two modes is in how the probability quartiles are calculated for the final statistics. These are useful guides for visualization and analysis (see the visualization section in the examples and the paper for further discussion).

Figure 1 Overview of the Provar process


Visualisation of persistent and variable pockets
Homologous structures
Multiple crystal structures of same protein
Simulated conformations
Solution NMR ensemble
Sequence alignment
Run existing pocket prediction programs (e.g. LIGSITE or PASS)
Provar algorithm
For each structure and prediction find pocket-lining atoms and residues. Use the complete set of pocket lining atoms/residues to calculate probability that atom/residue is pocket-lining
Use molecular visualisation software to display protein and colour atoms/residues using suitably defined probability ranges

Example 1 – Running Provar for a set of simulated conformations

This example shows how to use the Provar algorithm to process a set of 250 conformers generated from apo Bcl-2, alongside the PASS pocket predictions for each conformer to visualize the protein-protein interface (These data were used to generate Figure 8B in the paper.)

What is provided in the "example1_conformers" sub-directory:

- The set of 250 tCONCOORD simulated conformations of PDB ID: 1GJH under directory "structures". These are numbered sequentially 'conformer_1.pdb, conformer_2.pdb…'

- The matching PASS pocket predictions for each of the above conformers under directory "PASS_pocket_files".

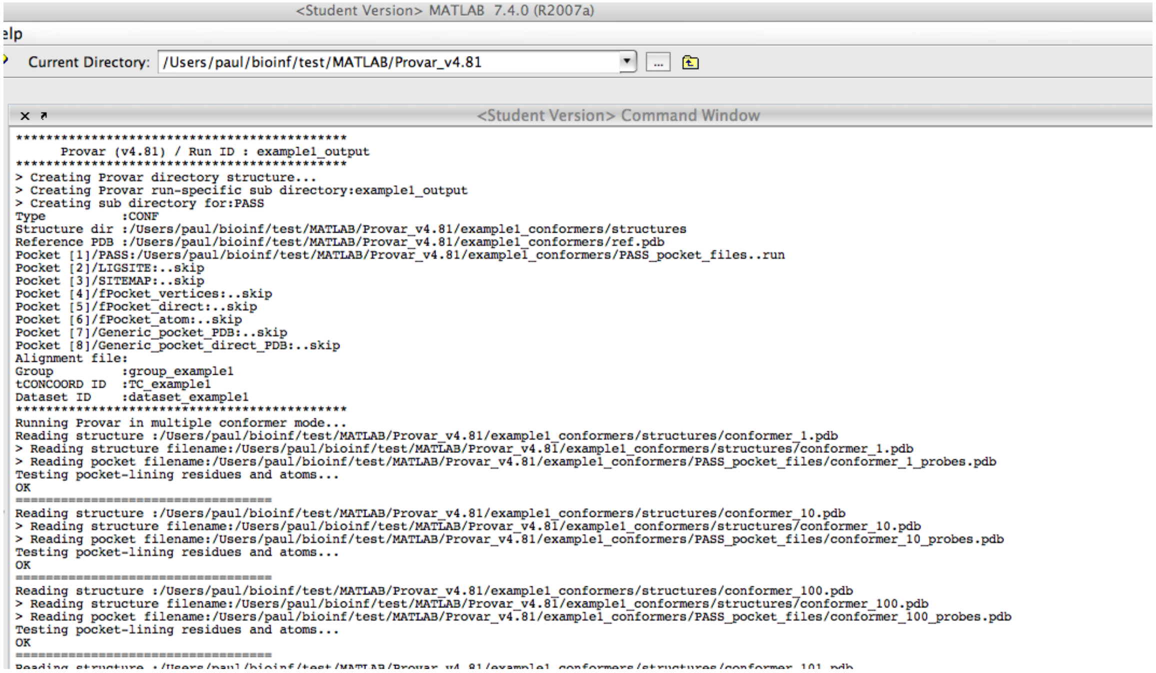
- The reference structure 'ref.pdb'. When Provar writes the probabilities back to a protein it uses this structure.

- The required MATLAB parameter file "fn_Provar_params.m"

To run example 1…

- Copy the parameter file "fn_Provar_params.m" into the "Provar_v4.81" directory
- In MATLAB open the module "Provar.m" in the Editor.
- Press "F5" (or select "Debug", "Run" from the Editor menu) to run this example

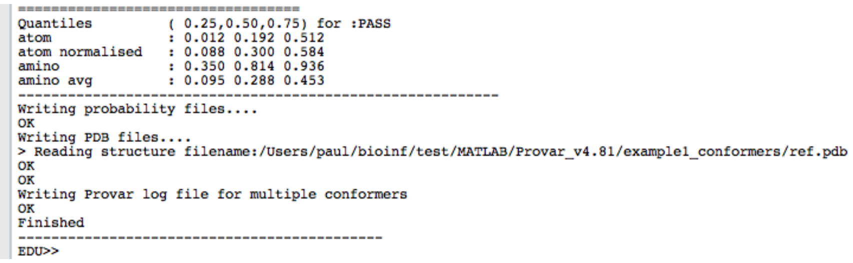
The command window will display progress of the job as illustrated…


<Student Version> MATLAB 7.4.0 (R2007a)
elp
<Student Version> Command Window

Current Directory: /Users/paul/bioinf/test/MATLAB/Provar_v4.81

```
********************************************
      Provar (v4.81) / Run ID : example1_output
********************************************
> Creating Provar directory structure...
> Creating Provar run-specific sub directory:example1_output
> Creating sub directory for:PASS
Type          :CONF
Structure dir :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures
Reference PDB :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/ref.pdb
Pocket [1]/PASS:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/PASS_pocket_files..run
Pocket [2]/LIGSITE:..skip
Pocket [3]/SITEMAP:..skip
Pocket [4]/fPocket_vertices:..skip
Pocket [5]/fPocket_direct:..skip
Pocket [6]/fPocket_atom:..skip
Pocket [7]/Generic_pocket_PDB:..skip
Pocket [8]/Generic_pocket_direct_PDB:..skip
Alignment file:
Group         :group_example1
tCONCOORD ID  :TC_example1
Dataset ID    :dataset_example1
********************************************
Running Provar in multiple conformer mode...
Reading structure :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_1.pdb
> Reading structure filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_1.pdb
> Reading pocket filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/PASS_pocket_files/conformer_1_probes.pdb
Testing pocket-lining residues and atoms...
OK
==================================
Reading structure :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_10.pdb
> Reading structure filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_10.pdb
> Reading pocket filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/PASS_pocket_files/conformer_10_probes.pdb
Testing pocket-lining residues and atoms...
OK
==================================
Reading structure :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_100.pdb
> Reading structure filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_100.pdb
> Reading pocket filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/PASS_pocket_files/conformer_100_probes.pdb
Testing pocket-lining residues and atoms...
OK
==================================
Reading structure :/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/structures/conformer_101.pdb
```

Once complete you should see the following Command Window output:



```
==================================
Quantiles        ( 0.25,0.50,0.75) for :PASS
atom           : 0.012 0.192 0.512
atom normalised  : 0.088 0.300 0.584
amino          : 0.350 0.814 0.936
amino avg       : 0.095 0.288 0.453
----------------------------------------------------------
Writing probability files....
OK
Writing PDB files....
> Reading structure filename:/Users/paul/bioinf/test/MATLAB/Provar_v4.81/example1_conformers/ref.pdb
OK
OK
Writing Provar log file for multiple conformers
OK
Finished
--------------------------------------------
EDU>>
```

Outputs

Under the “example1_conformers” directory will be a sub-directory “example1_output” containing:

“Provar_summary.txt”: A text summary of the files, folders and statistics.

A log file giving a report on the Matlab run.

A sub-folder for each pocket prediction program – in this case just PASS (abbreviation PA - see Appendix for supported programs) as that is the only one we supplied data for and asked for in the “fn_Provar_params.m” parameter file.

Each pocket program sub-folder will contain the probability output files:

example1_output_PASS_p_ATOM_out_3.75A-radial.txt

List of atoms in the reference structure, with Provar pocket-lining probabilites across the ensemble.

example1_output_PASS_p_amino_out_3.75A-radial.txt

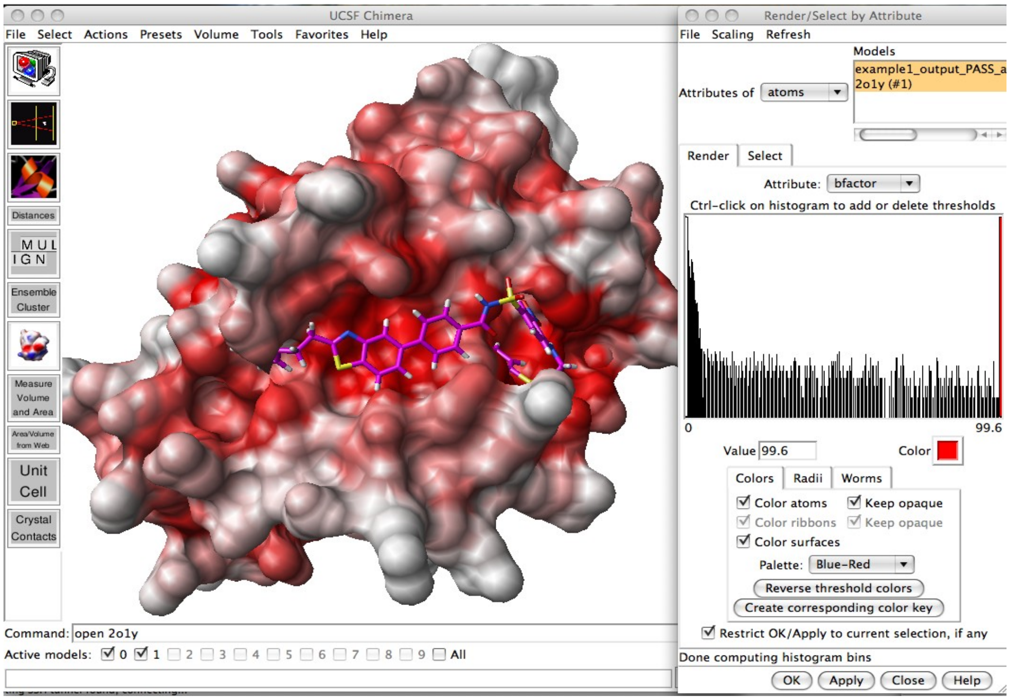
List of residues in the reference structure, with the amino-acid level probabilities calculated over the ensemble.

example1_output_PASS_p_amino_avg_by_atom_out_3.75A-radial.txt

The residue-based scores, normalized by the number of atoms in each residue.

Each of these files has an associated PDB structure, where the probability values have been written into the B-factor column (after multiplication by 100). We can now use a suitable molecular visualization program to render the output – we illustrate the examples using UCSF Chimera[4].

After loading the file "example1_output_PASS_atom_out_3.75A.pdb", we have chosen to display surface, then used Chimera's menu option: "Tools", "Structure Analysis", "Render by Attribute" (dialogue to the right below) to colour by probabilites from 0 (white) to 1 (red). (Note that b-factors are given in the scaled range 0-> 100).



Command: open 2o1y

Active models: ☑ 0 ☑ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □ 9 □ All

Attributes of: atoms

Models: example1_output_PASS_a 2o1y (#1)

Attribute: bfactor

Value 99.6 Color 🟥

☑ Color atoms ☑ Keep opaque

☑ Color ribbons ☑ Keep opaque

☑ Color surfaces

Palette: Blue-Red

☑ Restrict OK/Apply to current selection, if any

Done computing histogram bins

By adjusting the thresholds for the colouring, it is possible to choose which fraction of the conformers flag an atom as pocket lining. For example, if we choose to colour red only for values above 75, the red atoms line pockets in 75% of structures. In the paper we discuss parameterisation in more detail and in example 2 below we discuss the use of quartiles to define cut-offs.

Example 2 – Running Provar on a set of homologues

This example uses a set of 17 *apo* homologous structures of Interleukin-2 (IL-2) taken from representative domains of the CATH superfamily 1.20.1250.10. (These data were used to generate Figures 9 and 10 A&B in the paper).

What is provided in the "example2_homologues" sub-directory:

- The set of 17 single chain PDB files (with chain code) under "structures"

- The matching fPocket[5], PASS and Ligsite-cs[6] pocket predictions for each of the above under specified sub-directories

- The reference structure '2B5IA.pdb'. When Provar writes the probabilities back to a protein it uses this structure. (Note: this is a receptor structure for rendering an analysis purposes – it is not used for Provar calculations. See the comments in the parameter file for more info.)

- A FASTA sequence alignment, generated with Mustang (Mustang_alignment_apo_holo_receptor.fasta)

- The required MATLAB parameter file "fn_Provar_params.m"

To run example 2…

- Copy the parameter file "fn_Provar_params.m" into the "Provar_v4.81" directory (over-writing the version from example 1).
- In MATLAB open the module "Provar.m" in the Editor.
- Press "F5" (or select "Debug", "Run" from the Editor menu) to run this example

The command window will display progress of the job as illustrated – this output is automatically saved to a MATLAB .log file under the example 2 folder. This example generates outputs for three pocket programs simultaneously and the outputs are written to the "FD" (fPocket direct), "PA" (PASS) and "LC" (Ligsite-cs) sub folders of "example2_output.

A Provar run can include all supported pocket programs with a single structure set simultaneously. However, it is imperative that the number of structures, number of pocket predictions (per program) and the number of sequences in the alignment file match.

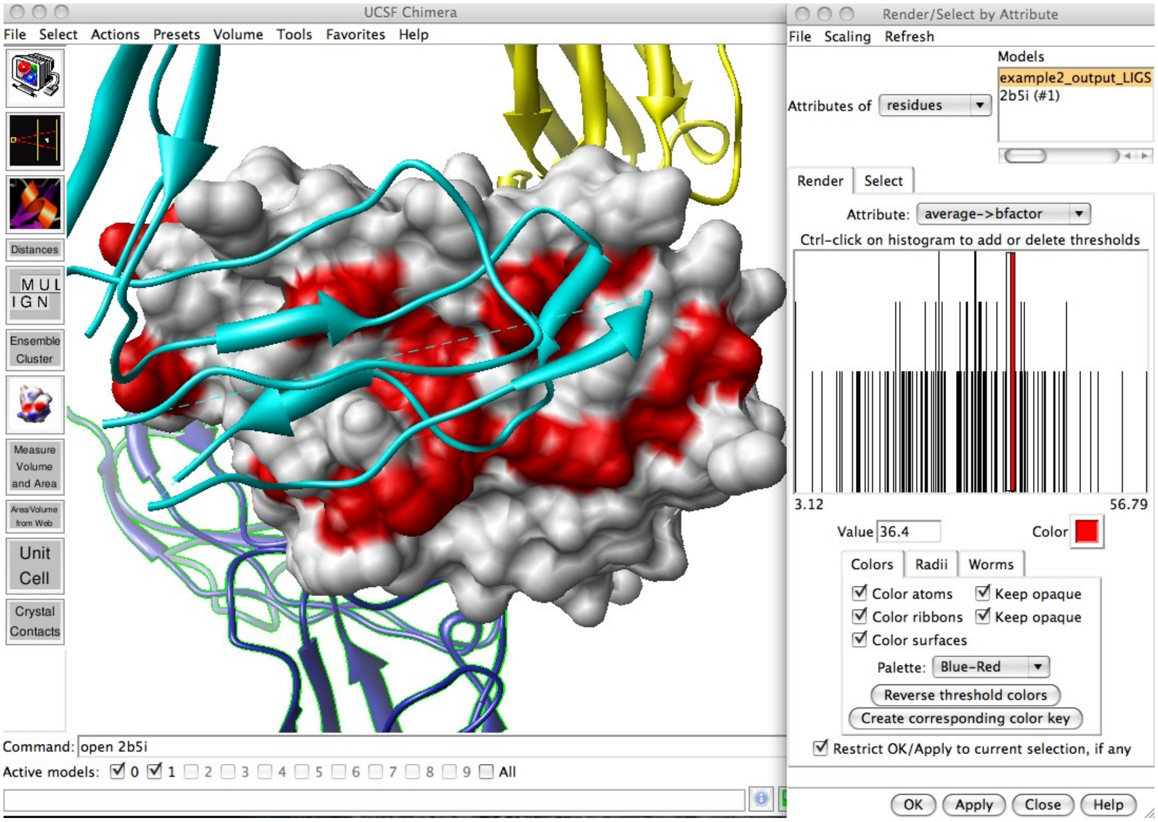
Outputs

These are of a similar format to example 1, with two differences:

- There are no atom based probability or PDB files
- There are a set of global alignment files.

(e.g. example2_output_LIGSITE_p_global_alignment_avg_out_3.75A-radial.txt)

These global files represent the probabilities at each of the global alignment positions in the FASTA file. Naturally, these can't be mapped to a protein structure for display, but they are used to calculate the probability distribution quartiles which are recommended for visualization. These quartiles can be found in "Provar_summary.txt"



Command: open 2b5i

Active models: ☑ 0 ☑ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □ 9 □ All

Models: example2_output_LIGS, 2b5i (#1)

Attributes of: residues

Attribute: average->bfactor

Value: 36.4

☑ Color atoms ☑ Keep opaque

☑ Color ribbons ☑ Keep opaque

☑ Color surfaces

Palette: Blue-Red

☑ Restrict OK/Apply to current selection, if any

We used Chimera’s “Render by Attribute” function as before to only show residues having probabilities > Q3 (quartile 3 = 0.364, or 36.4% in b-factor terms) in red. We have also superimposed the other chains from PDB: 2B5I to show the location of the IL2-IL2Rα interface. By rotating the beta and gamma interfaces can be similarly explored. This shows that by considering just the 17 *apo* structures of this structurally diverse superfamily, we can see pocket conservation overlapping with the known IL-2/IL-2R interfaces – this suggests that pockets are conserved at these sites for some functional reason. It is also possible to use the quartiles to render the most variably pocket-lining residues – this idea is discussed more fully in the paper.

Appendices

Supported pocket prediction programs

Idx	Pocket program	Abbreviation	Expected pocket file suffix	Direct?	MATLAB parameter
1	PASS	PA	_probes.pdb	N	`params.pass_dir`
2	LIGSITE-cs	LC	.pdb_pocket_r.pdb	N	`params.ligsite_dir`
3	Sitemap	SM	_sitemap.pdb	N	`params.sitemap_dir`
5	FPocket	FD	_out/pockets/	Y	`params.fpocket_dir`
6	“Generic”	GP	_pocket.pdb	N	`params.generic_pocket_dir`
7	“Generic direct”	GD	_pocket.pdb	Y	`params.generic_pocket_direct_dir`

Note: “Direct” is where pocket programs provide a list of the pocket-lining atoms in a PDB file, thus Provar does not then need to calculate these based on geometric means. For example, Fpocket’s “pocket_atm.pdb” and CASTp’s “.poc” outputs are suitable, when renamed according to the expected suffix.

Notes and issues

- **Conformational ensembles:** Addition of hydrogens… tCONCOORD instructions recommend the addition of H atoms prior to simulation. However, we’ve experienced unpredictable results from PASS and LIGISTE if the structures containing H-atoms are used as the bases for pocket prediction, therefore they should be removed prior to pocket prediction. The structures in Example 1 have had H atoms removed using a script.
- **Compatibility:** v4.81 uses an invalid Windows syntax for determining fPocket direct file locations.

Acknowledgements

Provar uses a module available from MATLAB Central File Exchange called “PDBRead” by Sachin Nikumbh.
http://www.mathworks.co.uk/matlabcentral/fileexchange/1288-pdbread

References

[1] de Groot B, van Aalten D, Scheek R, Amadei A, Vriend G, Berendsen H: Prediction of protein conformational freedom from distance constraints. Proteins 1997, 29:240–51.

[2] Brady G, Stouten P: Fast prediction and visualization of protein binding pockets with PASS. J Comput Aided Mol Des 2000, 14:383–401.

[3] Konagurthu AS, Whisstock JC, Stuckey PJ, Lesk AM: MUSTANG: a multiple structural alignment algorithm. Proteins 2006, 64:559–74.

[4] Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE: UCSF Chimera–a visualization system for exploratory research and analysis. J. Comput. Chem. 2004, 25:1605–12.

[5] Schmidtke P, Guilloux VL, Maupetit J, Tuffery P: fpocket: online tools for protein ensemble pocket detection and tracking. Nucleic Acids Research 2010, 38:W582–W589.

[6] Hendlich M, Rippmann F, Barnickel G: LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J Mol Graph Model 1997, 15:359–63, 389.