3D-Dock

incorporating FTDock (version 2.0), RPScore, and Multidock

Gidon Moont, Graham R. Smith‡and Michael J. E. Sternberg†

March 2001

Contents

1	Introduction1.1Key to font usage1.2Requirements	
2	Algorithms	4
3	Installation	5
4	Tutorial	10
5	Manuals	13
	5.1 preprocess-pdb.perl	
	5.2 change-pdb-chain-id.perl	
	5.3 ftdock	
	5.4 rpscore	
	5.6 filter	
	5.7 build	
	5.8 randomspin	
	5.9 centres	
		Biomolecular Modelling Laboratory Imperial Cancer Research Fund Lincoln's Inn Fields, P.O. Box 123 London, WC2A 3PX England http://www.bmm.icnet.uk/
•	whom technical questions should be addressed. 1: $+44 \ (0)20 \ 7269 \ 3348$	[†] To whom other correspondence should be addressed. Tel: $+44 (0)20 7269 3565$

Fax: +44 (0)20 7269 3534email: graham.smith@icrf.icnet.uk % f(x) = f(x) + f(x) +

Fax: +44 (0)20 7269 3534 email: m.sternberg@icrf.icnet.uk

1 Introduction

3D-Dock is a suite of programs designed to enable computational prediction of proteinprotein docking. It does this in several steps, as described in Algorithms (2) below. This document is designed to enable the various programs to be run successfully, as well as provide a basic understanding of the underlying algorithms.

Although the suite includes the program multidock, it is not covered by this document (please see http://www.bmm.icnet.uk/docking/).

Although this document explains the basics of how the programs work it does not discuss how various parameters or strategies were decided upon. For this information please refer to the published papers in the References at the end of the document.

1.1 Key to font usage

To try and make things slightly clearer, different fonts are used in this section to signify different things.

- Normal font is explanation and hence most text.
- typewriter font is used for program names, things that would be typed on a command line, and things that would be seen when looking in a file.
- *italics are used for file and directory names*

1.2 Requirements

There are several different requirements that have to be met in order to run this suite of programs. These fall into 3 categories; operating system, hardware, and software.

- **operating system** The main programs were written on, and with an aim to running on, a UNIX style operating system. They were actually written on an SGI/IRIX platform, but have also been tested on the easily available Linux, running on both Intel x86 and DEC Alpha processors. Anything else is not supported, though since the programs are in Perl and C, it is possible that you could compile and run them on something else.
- hardware The main limitation to hardware is RAM. ftdock uses large amounts of memory, and although you could set the parameters to lower this, standard run of the program will want up to 100 Megabytes of memory. If you do not have this as RAM, the program will be paging constantly and may well take weeks to run.
- software You will need a C compiler (no binaries will be made available, due to the optimisation of fourier transform routines depending on the exact architecture of the machine on which they are compiled), and PERL, version 5.003 or later. The only non-standard C libraries required are those of the fast fourier transform, which you will need to download and compile (see Installation (3)).

2 Algorithms

This suite of programs is intended to be able to dock two proteins. This means starting from the known structures of two protein subunits of a biological complex known to exist, in unbound conformations, and ending up with a limited set of possible models for the complex. This overall algorithm is here achieved in up to 4 steps.

- 1. a global scan of translational and rotational space of possible positions of the two molecules, limited by surface complementarity and an electrostatic filter (ftdock).
- 2. an empirical scoring of the possible complexes using residue level pair potentials (**rpscore** (and **rpdock**)).
- 3. using biological information to screen the possible complexes (filter).
- 4. an energy minimisation and removal of steric clashes on the side-chains of the interface (multidock [3]).

The middle two steps are interchangeable in the order in which they are run, and the filter can be run more than once if so desired (see Tutorial). A schematic of the overall approach is shown in Figure 1.

The ftdock algorithm is based on that of Katchalski-Katzir [4]. It discretises the two molecules onto orthogonal grids and performs a global scan of translational and rotational space. In order to scan rotational space it is necessary to rediscretise one of the molecules (for speed the smaller) for each rotation. The scoring method is primarily a surface complementarity score between the two grids, and this is shown in Figure 2. To speed up the surface complementarity calculations, which are convolutions of two grids, Fourier Transforms are used. This means that the convolutions are replaced with multiplications in Fourier space, and despite having to perform the forward and reverse Fourier Transforms, this decreases the overall computation required. The surface complementarity was the only score used in the original method. The original work on ftdock by Gabb [2] found it a useful addition to include an electrostatic filter, and this is again implemented in the current version (though it can be turned off).

The **rpscore** program uses an empirical pair potential matrix to score each possible complex. The pair potentials are at a amino acid residue level. Each potential corresponds to the empirically derived likelihood of a trans-interface pair of two residue types, limited only by a distance cut off [5]. The present most useful matrix is generated from 90 non-homologous interfaces found in the PDB with the aid of SCOP 1.53 (http://scop.mrclmb.cam.ac.uk/scop/), and is shown graphically in Figure 3. If two interfaces are described as pairings of domains A - B and C - D, then a non-homologous interface is defined as being when either A and C, or B and D, are homologous, but not both. Homology is in this case defined as being in the same 'Superfamily' in the SCOP classification tree.

The biological filter is a simple program to screen the complexes by requiring them to have a given chain or residue on one side of the interface within a certain distance of another chain or residue on the other side. The manual (5.6) explains this in full.

For the manual and program multidock please see http://www.bmm.icnet.uk/docking/. This calculates side-chain energy minimisations and removes steric clashes along the interface. It is presently only available as an SGI IRIX5.3 or Linux i386 executable.

3 Installation

You should have obtained the files $gnu_licensed_3D_Dock.tar.gz$ (containing the FTDock component of 3D-Dock) and fftw-2.1.3.tar.gz by ftp. The files for RPScore and Multidock should have been emailed to you; extract them from your mailer into files called, for example, rpscore.tar.gz and multidock.tar.gz. The installation of multidock is covered in its own manual and will not be repeated here.

The first thing is to compile FFTW. You do not need to install it. To do this you should not have to do more than

gunzip fftw-2.1.3.tar.gz — gives you file fftw-2.1.3.tartar xvf fftw-2.1.2.tar — makes a directory fftw-2.1.3 with all the bits inside it.

Change into that directory then

```
./configure --enable-float
make
```

This is all you need to do. There is of course no harm in installing it properly (make install). The reason for using --enable-float is to reduce (typically halve) the memory requirements. If you are going to use FFTW for other programs you may need to reconsider if you want this.

Once you have done this, you can compile the actual programs. FTDock first:

```
gunzip gnu_licensed_3D_Dock.tar.gz — gives you file gnu_licensed_3D_Dock.tar
tar xvf gnu_licensed_3D_Dock.tar — makes a directory 3D_Dock with all the bits inside
it.
```

Change into that directory, and then into the *progs* directory. You will have to edit the *Makefile* to give the correct complete path to the *fftw-2.1.3* directory. This is done by setting the variable FFTW_DIR on line 15. (If you have fully installed FFTW, you will need to set this variable to the directory where it has installed itself, not the directory where the source is.) Then

make

You should now have the executables *ftdock*, *build* and *randomspin*, which are enough to run the first (FTDock) part of the docking suite. The most likely cause for failure is incorrect C flags for your compiler. You will have to change the CC_FLAGS line in the *Makefile* to correct this problem. Now for RPScore:

gunzip rpscore.tar.gz — gives you file *rpscore.tar* tar xvf rpscore.tar — makes a directory *additional_programs* with the code inside it.

Copy all the .c files and the *Makefile* from *additional_programs* into the $3D_Dock/progs$ directory. The new *Makefile* should overwrite the one in $3D_Dock/progs$ previously. Now change into $3D_Dock/progs$. You should edit FFTW_DIR in the *Makefile* again, because you have lost the value you just put in the old one a minute ago. Then once again do

make

You should now have the additional executables *rpscore*, *rpdock*, *filter* and *centres*.

This may seem, and to some extent is, an unnecessarily complicated compilation procedure. However, we are forced into it by the need to issue FTDock and RPScore as two separate stand-alone programs under different licenses.

Finally, change into the 3D_Dock/scripts directory. The first line of the two perl scripts specifies the location of the perl executable; the value given is /usr/bin/perl. If perl is installed elsewhere on your machine (type which perl) you will need to change this line to the correct path.

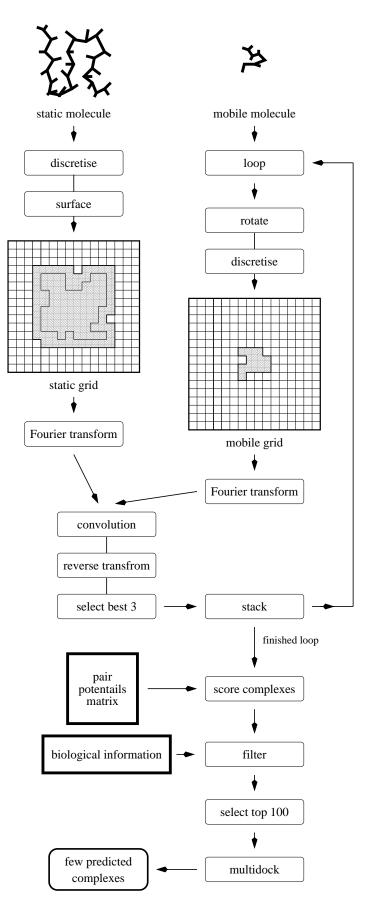


Figure 1: Flow diagram of overall docking method

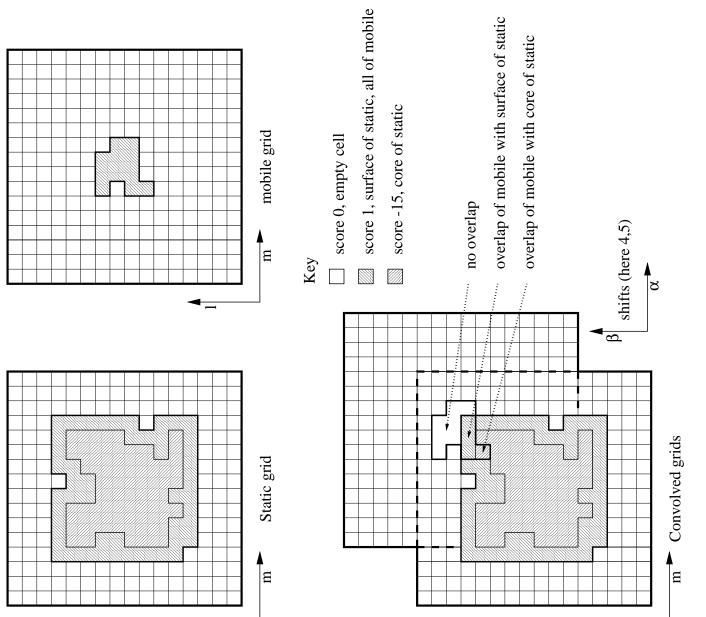


Figure 2: Grid discretisation of molecules and calculation of surface complementarity

< __

Key to Scale

+0.7 • -0.7 ○

	D	E	K	D	٨	v	Б	Р	М	т	L	XX /	v	N	С	Ω	C	ы	c	Т	
D			<u>r</u>		\mathbf{A}	v		r				••		N 。		Q		H	<u>5</u>		Aspartic Acid
E	0	\cap	•	•	0	0	0	0		0	0						0	•		0	Glutamic Acid
K	•		\cap	•	\bigcirc	0		0		0	•		•	0		0	0		0	0	Lysine
R		•		0			•			•					•	•			•	-	Arginine
A	0	•	\cap	0	0		•		•	•		•	•	0	0	0	0		0	0	Alanine
V	0	•	0		<u> </u>		•			•	•		•				0		0	•	Valine
• F		•		•	•	•	•							0	•	•	•				Phenylalanine
P	0	0	0		\cap	•		\cap	•	0	•		•		0	•	0	•		0	Proline
г М	<u> </u>	•		•	•	°				•		-				•	•			•	Methionine
IVI	0	•	•	•	•	•			•	•	•		•			•	•	•	0	•	Isoleucine
I T		0	-			-				•	-		-			•	-	-			
	0		•	•		•	•	•	-	•	•	•	•	0	•		0	•	0	•	Leucine
W			•	•	•	•	•	•		•	•			•		•	•		·	•	Tryptophan
Y	•	·	•	•	•	•	•	•		•	•		•	•	•	•	•	•	•	•	Tyrosine
N	0	•	0	•	0	0	•	•	•	•	0	•	•	0	0	·	0	0	0	0	Asparagine
С	\bigcirc	0	•	•	°	0	•	0			•		•	0		•	•		•	0	Cystine
Q	•	•	0	•	0	·	•	0	•	•	-	•	•	·	•	0	•		•	0	Glutamine
G	0	0	0		0	0	0	0	•	0	0	•	•	0	•	•	0	•	0	0	Glycine
Η	·	•		•	•	۰	•	0		٠	•			0			·	•	•	•	Histidine
S		•	0	•	0	0	•	0		0	0		•	0	•	•	0	•	\bigcirc	•	Serine
Т	0	0	0		0	o		0	•	•	•		•	0	0	0	0	•	•	0	Threonine

4 Tutorial

This tutorial will take the example of bovine pancreatic trypsin inhibitor bound to kallikrein A complex. The necessary PDB files are included in the distribution. You will have to give the full paths to the various executables as appropriate. This tutorial uses the minimum number of options for each program. For complete options and further details please see the manuals section (5) below.

All this tutorial presumes you are executing all the programs in the same directory, and not changing the names of any files produced.

Preprocessing

Manually edit the PDB files so that you have the components you want to dock. Then

```
preprocess-pdb.perl -pdb file.pdb
```

This will give any number of messages, normally complaining of non-standard residue designations. I do not recommend you use this program indiscriminately for other work as it removes everything but the ATOM records of the 20 standard residues it recognises, and it also removes Hydrogens and OXT records as well. The output will have the name *file.parsed*. It will also produce a FASTA format file called *file.fasta* which you may find useful.

Global scan

To run the main program type

```
ftdock -static 2pka.parsed -mobile 5pti.parsed > output &
```

I recommend you redirect the standard out for safety reasons. The program is going to take a long while to run, and it will want to write out stuff throughout. If you want to be able to close the shell without crashing the program, you need to do this. In order to see what is going on, the following UNIX command is ideal

tail -f output

The output you will now have is the file $ftdock_global.dat,$ which will contain 10000 records. (best rank on my run = 1619)

Pair Potential scoring

First copy the file *additional_programs/i90_p05_d4.5_2dp.matrix* to *best.matrix* in the current directory. Then, in order to assign a pair potential score to each record, you should type

rpscore

This very simple command will only work if it can read from a data matrix *best.matrix* in the current directory.

The output is $ftdock_rpscored.dat$, which contains the same 10000 records, but reordered by the new score. (best rank on my run = 65)

Filtering

As is often the case, we have biological information which can reduce the number of possibilities. We want to filter such that the remaining complexes have the inhibitor (chain I) in proximity (distance default is 4.5 Angstroms) to the catalytic triad of the enzyme (chains A and B). This is expressed as

filter -constraints A57:I B102:I B195:I

Each constraint is treated as an **OR** statement. The designators each side of the colon are of the form chainID then residue number (+ insertion code if defined), and for the whole chain, the residue number is simply missed out. The order is irrelevant, so

filter -constraints I:B195 I:A57 B102:I

would give the same output. For more explanations see the manuals section (5) below.

The output is *ftdock_filtered.dat*, which contains a reduced set of records. (in my run 900) (best rank on my run = 12)

In order to have the effect of an ${\bf AND}$ statement, you will have to run the filter program several times.

filter -constraints A57:I -out ftdock_filter_one.dat

then

```
filter -constraints B102:I -in ftdock_filter_one.dat -out ftdock_filter_two.dat
```

then

```
filter -constraints B195:I -in ftdock_filter_two.dat -out ftdock_filtered.dat
```

(best rank on my run = NA . This can often happen that a too strict series of constraints will loose good results)

Side-chain refinement

For the manual and program multidock please see http://www.bmm.icnet.uk/docking/. This calculates side-chain energy minimisations and removes steric clashes along the interface. It is presently only available as an SGI IRIX5.3 or Linux i386 executable.

5 Manuals

5.1 preprocess-pdb.perl

Due to the nature of PDB files, a preprocessor is used to both clean up and add limited information to the PDB files. The cleaning method is described below. The added information is simply the one letter amino acid codes, and a numerical assignment for each residue type, assigned in alphabetical order (1–20).

what the cleaner does

- 1. removes all residues that are not one of the twenty standard amino acids or one of the five standard nucleic acids.
- 2. only keeps atoms it recognises as 'useful' so removes all Hydrogen atoms. It also removes 'OXT' terminal Oxygens, simply because their assignment is not always sensible.
- 3. removes all but the first of an alternative atom indicator entry.
- 4. checks for the correct number of atoms for that residue, then
 - if too many, checks for doubles of any atom type labels and removes all but first (ie copes with missing Alternate Indicator).
 - if still too many atoms for residue, then checks for atom type validity for that residue type.
 - if still too many, will chuck (remove) that residue.
 - if too few, will do nothing, unless -multidock is set, in which case it will attempt to replace with a modelled Alanine.

Command line options

-pdb	PDB style file name no default
-nowarn	turns off all but the most severe warnings
-multidock	this makes the output fit for input into the program multidock this is not for use prior to running ftdock it will change to model Alanine any residue which does not contain its full complement of (non-Hydrogen) atoms

5.2 change-pdb-chain-id.perl

A script to change PDB ChainIDs.

Command line options

-pdb	PDB style file name no default
-old	chain ID that you want to change for a non labelled chain use '' no default
-new	replacement chain ID that you want for a non labelled chain use '' no default

examples

change-pdb-chain-id.perl -pdb 2pka.pdb -old A -new E change-pdb-chain-id.perl -pdb 1hpt.pdb -old ' ' -new I

5.3 ftdock

The main global docking program. Due to the rescue abilities, please do not run this in a given directory more than once at any one time.

-out	output file name default is <i>ftdock_global.dat</i>
-static	larger of the two molecules being docked this PDB style file must be output from preprocess no default
-mobile	smaller of the two molecules being docked this PDB style file must be output from preprocess no default
-grid	number of grid units in one dimension this means a grid of 64 has 64^3 grid units in total this also means that memory requirements go roughly as n^3 of grid size a grid that results in a grid spacing of more than 1 angstrom is unlikely to be useful the grid size must be integer and even (to ease Fourier calcu- lations) no default
-calculate_grid	the desired size of a single grid unit in angstroms due to the limitations on the grid size, the actual grid unit will vary slightly (less than ± 0.01) from the given value default is on with a value of 0.875 to turn off, use -grid option

-angle_step	the maximum planar angle (in degrees) separating any two ro- tations of the mobile molecule when subtended to the point around which the rotation takes place (geometric centre of the mobile molecule)default is 12 degreeswill only accept integer values that are integer factors of 180
-surface	surface thickness in angstroms default is 1.5
-internal	internal deterrent value default is -15.0
-noelec	electrostatics calculations switch default is to do the electrostatics, this switch will turn them off
-keep	number of (best surface complementarity) translations to keep from each rotation default is 3
-rescue	<pre>if your machine falls over, then just run ftdock -rescue in the same directory do not alter anything between the crash and using this to make this option available in this very simple form, two files exist in the directory from which you run ftdock; namely scratch_parameters.dat and scratch_scores.dat. This means that you should not run ftdock more than once at any given time in the same directory. There is no system at present to prevent this from being done, so be careful.</pre>

Understanding the output

Output 1 shows a typical output file to a run of ftdock (default *ftdock_global.dat*) or defined by the -out option). All values that can be controlled by the command line (apart from the output file name) are shown at the top of the file. Along with each value is information showing whether it has been chosen or is the default value (apart from for the molecules which are required and have no default values). These lines must not be changed since the values are used by other programs which use this file for input. In general, it is suggested that any data files produced by any of the programs should not be edited directly, and there should be no need to do so.

Below this are shown a few calculated values. These are purely for the users information and are never used by any of the other programs.

The G_DATA lines contain all the information corresponding to each putative complex. The ID is that used for the build program (see 5.7 below). The previous ID (prvID) is zero in this case as this is the first program. The Surface Complementarity Score (SCscore) is

the value which determines the order of the file, the highest score having the lowest ID. The electrostatic score ratio (ESratio) is there to possibly show varying electrostatic favourability when a group of complexes have the same surface complementarity. It is a ratio as opposed to an absolute value, ranging from 0 (least favourable) to 100 (most favourable). After this come the translational coordinates (x, y, z) expressed as integer grid cell displacements of the mobile molecule's centre from the centre of the static molecule. At the end come the rotational angles $(z_{twist}, \theta, \phi)$ expressed in degrees.

```
FTDOCK data file
Global Scan
Command line controllable values
Static molecule
                                    :: static.parsed
Mobile molecule
                                    :: mobile.parsed
Global grid size
                                    ::
                                          110
                                                    (default calculated)
Global search angle step
                                    ::
                                           12
                                                    (default)
Global surface thickness
                                                    (default)
                                    ::
                                            1.40
Global internal deterrent value
                                    ::
                                          -15.00
                                                    (default)
Electrostatics
                                                    (default)
                                    ::
                                           on
Global keep per rotation
                                    ::
                                            3
                                                    (default)
Calculated values
Global rotations
                                    ::
                                         9240
Global total span (angstroms)
                                           96.079
                                    ::
Global grid cell span (angstroms)
                                            0.873
                                    ::
Data
Type
           ID
                 prvID
                           SCscore
                                          ESratio
                                                           Coordinates
                                                                                   Angles
G_DATA
                     0
                                           13.992
                                                               19
                                                                    -7
            1
                               173
                                                          22
                                                                              120
                                                                                   96 228
G_DATA
            2
                               173
                                            1.941
                                                                4
                                                                    -4
                                                                              144
                                                                                   96 336
                     0
                                                         -27
G_DATA
            3
                     0
                               163
                                           17.331
                                                         -23
                                                               10 -11
                                                                              180
                                                                                   84 156
```

Output 1: Example output from ftdock

5.4 rpscore

The residue level pair potentials scoring program.

-in	input file name default is <i>ftdock_global.dat</i>
-out	output file name default is <i>ftdock_rpscored.dat</i>

-matrix

matrix file name default is *best.matrix* this can be found in the additional_programs directory

Understanding the output

Output 2 shows a typical output file to a run of **rpscore** (default *ftdock_rpscored.dat*) or defined by the **-out** option). All the information from the run of **ftdock** and any previous runs of **rpscore** or **filter** are still at the top of the file, followed by the command line controllable matrix.

The G_DATA lines contain all the information corresponding to each putative complex. The fields are identical to those in the output from ftdock with the exception of RPscore which replaces ESratio. The complexes are now ordered by their residue level pair potential scores (RPscore), and the prvID field has values corresponding to the ID field in the input data file. The prvID field can be used to track the ranking of a complex as the successive programs are run.

```
FTDOCK data file
Global Scan
Command line controllable values
Static molecule
                                   :: static.parsed
.
Global grid cell span (angstroms) ::
                                           0.873
*****
Residue level Pair Potential Scoring
Command line controllable values
Matrix
                                    :: /home/ftdock/data/best.matrix
                                                                        (user defined)
Data
Type
           ID
                 prvID
                          SCscore
                                         RPscore
                                                          Coordinates
                                                                                  Angles
G_DATA
            1
                   502
                              139
                                          14.778
                                                         12
                                                              36
                                                                   14
                                                                            276 24 120
G_DATA
                  5839
                                                              -5
                                                                   7
                                                                             96 60 276
            2
                              114
                                          14.011
                                                         37
G_DATA
            3
                    21
                              151
                                          13.196
                                                         12
                                                              36
                                                                   14
                                                                            276 156 120
```

Output 2: Example output from rpscore

5.5 rpdock

The residue level pair potentials scoring program for use with complexes generated by another docking program apart from ftdock.

Command line options

-p1	PDB style file of one side of the complex. Must have been parsed with pre-process.perl . no default
-p2	<pre>PDB style file of the other side of the complex. Must have been parsed with pre-process.perl. no default</pre>
-matrix	matrix file name default is <i>best.matrix</i> this can be found in the additional_programs directory

Understanding the output

The program returns a line of the form

G_DATA -3.646

to standard out. To screen a list of complexes it is advised to write a perl script wrapper.

5.6 filter

The biological filter program.

-in	input file name default is <i>ftdock_rpscored.dat</i>
-out	output file name default is <i>ftdock_filtered.dat</i>
-distance	the inter-atomic distance cut-off (in angstroms) for determin- ing whether the residues, of which a given two atoms are members of, are in contact or not. default is 4.5

a space separated list of the form

chainID[residuenumber][Icode]:chainID[residuenumber][Icode]
easiest explained by example

- 1. residue 45 of chain A to be in contact with chain B $\tt A45:B$
- residue 45 of chain A to be in contact with residue 3 of chain B A45:B3
- 3. residue 45 , insertion code A, of chain E to be in contact with chain I ${\tt E45A:I}$

the list is treated as a set of logical OR statements, so if any are satisfied, the statement is satisfied. there is a limit of 50 constraints if no constraints are given, the program will simply not run

Understanding the output

Output 3 shows a typical output file to a run of rpscore (default *ftdock_filtered.dat*) or defined by the -out option). All the information from the run of ftdock and any previous runs of rpscore or filter are still at the top of the file, followed by the command line controllable matrix.

The G_DATA lines contain all the information corresponding to each putative complex. The fields are identical to those in the output from rpscore. The complexes are ordered by their residue level pair potential scores (RPscore), and the prvID field has values corresponding to the ID field in the input data file.

5.7 build

The program to build a complex or a range of complexes.

-in	input file name default is <i>ftdock_rpscored.dat</i>
-b0	single complex number to build no default
-b1	beginning of range of complex numbers to build default is 1
-b2	end of range of complex numbers to build default is 10000

FTDOCK data file Global Scan Command line controllable values Static molecule :: static.parsed :: /home/ftdock/data/best.matrix (user defined) Matrix ***** Filter Command line controllable values Constraints :: A57:I B102:I B195:I (3) Distance 4.50 (default) :: Data Type ID prvID SCscore RPscore Coordinates Angles G_DATA 1 14.778 36 276 24 120 1 139 12 14 G_DATA 2 3 151 13.196 36 14 276 156 120 12

Output 3: Example output from filter

-c_alpha

build only the $C\alpha$ atoms

Understanding the output

The outputs from this program are the modelled complexes in PDB format. (There is extra information beyond column 80, but this should not cause problems to other programs such as visualisation tools.) The complexes are called $Complex_xg.pdb$, where x corresponds to the record ID number in the input file. If the $-c_alpha$ option is used, this changes to $CA_Complex_xg.pdb$.

5.8 randomspin

A program that randomly spins a PDB file. This is useful for testing the stability of a docking algorithm with respect to the initial orientations of the molecules.

-in	input file name
	default is unspun.pdb
-out	output file name default is <i>spun.pdb</i>

5.9 centres

A program to visualise the spread of positions of the mobile molecule with respect to the static molecule. The output PDB file *centres.pdb* contains the static molecule and water molecules, each of which represents the central position of the mobile molecule for each complex. Can show clustering of results when it occurs.

Command line options

-in

input file name default is *ftdock_global.dat*

References

- P Aloy, G Moont, H. A. Gabb, E. Querol, F. X. Aviles, and M. J. E. Sternberg. Modelling repressor proteins docking to dna. *Proteins*, 33:535–549, 1998.
- [2] H. A. Gabb, R. M. Jackson, and M. J. E. Sternberg. Modelling protein docking using shape complimentarity, electrostatics, and biochemical information. J. Mol. Biol., 272:106–120, 1997.
- [3] R. M. Jackson, H. A. Gabb, and M. J. E. Sternberg. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. J. Mol. Biol., 276(1):265– 285, 1998.
- [4] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and S. J. Wodak. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Nat. Acad. Sci.*, 89:2195–2199, 1992.
- [5] G. Moont, H. A. Gabb, and M. J. E. Sternberg. Use of pair potentials across protein interfaces in screening predicted docked complexes. *Proteins*, 35(3):364–373, 1999.