ProFit Version 3.1

Dr. Andrew C.R. Martin, Dr. Craig T. Porter
University College London

Document First Written: 25th July, 1996 (University College London)
Updated while at University of Reading
Last updated: 17th February, 2009

1 Introduction and Methodology

ProFit (pronounced Pro-Fit, not profit!) is designed to be the ultimate program for performing least squares fits of two or more protein structures. It performs a very simple and basic function, but allows as much flexibility as possible in performing this procedure. Thus one can specify subsets of atoms to be considered, specify zones to be fitted by number, sequence, or by sequence alignment.

Early versions of **ProFit** did not try to address the question of sorting out equivalent atoms for you beyond doing a sequence alignment. There are other programs such as SSAP and GAFIT which address that problem. You must specify which residues and atoms you consider to be equivalent although the program supports internal sequence alignment to set the zones automatically.

As of **ProFit** V2.0, iterative updating of fitting zones is now supported. Thus you may give a sequence alignment or just a small fragment to initiate the fitting process (a minimum of 3 amino acids). Fitting is performed on this region and then all residue pairs within 3Å are included in the fitting zones and the fitting is repeated. This iterates until the Cα RMSd converges to within 0.01Å. This is particularly useful in conjunction with the initial zone specification based on sequence alignment. Convergence typically takes 3–4 cycles.

ProFit V2.0 also introduced multiple structure fitting. The first structure file is used as a reference set for the first fitting stage but the coordinates are averaged after each stage to derive a template used for subsequent fitting. i.e. Given N files to fit, file 2 is fitted to file 1 and an averaged structure, A, is calculated, file 3 is then fitted to A and a new average, A' is calculated. This continues until all N structures have been fitted. The whole procedure iterates until convergence (typically 3 or 4 cycles).

The program will output an RMS deviation and optionally the fitted coordinates. RMS deviations over alternate zones and atoms may also be calculated without performing a new fit. Thus the zones for calculating the RMS deviation can be different from those used for fitting.

While optimised for proteins, non-protein structures may also be fitted if they are stored in the standard Protein Databank (PDB) format.

ProFit is written to be as easily portable between systems as possible and uses a command-driven interface.

ProFit uses the McLachlan fitting algorithm, essentially a steepest descents minimisation, as described in McLachlan, A.D. (1982) *Rapid Comparison of Protein Structures*, *Acta Cryst.* **A38**, 871–873. This part of the code is based on an implementation by Dr. Mike Sutcliffe.

In summary, **ProFit** has the following features:

1. Portability between different operating systems
2. Ability to specify atom subsets
3. Ability to specify zones:
    - Numerically
    - By sequence
    - By auto sequence alignment
    - By iterative updating and optimization
4. Output RMS deviation over:
    - Fitted region
    - Any other region
    - Any other atom set
5. Optionally output fitted coordinates in PDB format
6. Integrated help facility
7. Fitting zones derived from sequence alignment
8. Iterative updating of fitting zones
9. Multiple structure fitting

2 Version numbering

From V2.6, version numbering in **ProFit** adopted the following scheme. A version of **ProFit** has a version number of the form V$a.b.c.d$ where c and d are optionally present.

a is the major version number. A change in this represents a significant rewrite of **ProFit** and/or the addition of a major new feature set.

b is the minor version number. A change in b represents a new feature added to **ProFit**. It may also indicate a fix to a major show-stopper bug.

c is a bug-fix number. It indicates that a bug in the **ProFit** code has been fixed compared with the previous release.

d is also a bug-fix number, but indicates that bugs have been fixed in the Bioplib libraries used by **ProFit** and not in **ProFit** itself. Alternatively, it may indicate a distribution bug (i.e. missing files for a distribution or a simple documentation change).

In V2.5.x, b was used to represent a new feature *or* a bug fix while c was used to indicate a bug-fix in Bioplib (d is now used for that purpose).

In earlier releases, the scheme was used much more loosely with b being used both for new features and bug fixes in **ProFit** or in Bioplib. Lettered versions (e.g. V1.7g) were sometimes used for bug-fixes and sometimes for internal non-released versions. Lettered versions are now used exclusively for internal non-released versions.

3 Installation

To install ProFit, unpack the tar file. This will create a ProFit directory with a src subdirectory.

3.1 Compiling under UNIX-like operating systems

After unpacking the tar file, go into the src subdirectory of the ProFit directory. It is possible to edit the Makefile to allow the (optional) support for the XMAS library, the GNU Readline library or decompression of gzipped PDB files.

Type:

```
make
```

to create an executable file called 'profit'.

Under recent Linux installations, GNU Readline support should just work. Uncomment the two lines in the Makefile:

```
READLINE    = -DREADLINE_SUPPORT
READLINELIB = -lreadline -lcurses
```

and compile as normal. You may need to install the readline development libraries first, which is done with a command like:

```
yum install readline-devel         (RPM-based systems)
apt-get install libreadline5-dev   (Debian-based systems)
```

On other systems, you will need to obtain and install readline. See: `http://directory.fsf.org/project/readline/`

If you need to install GNU readline manually, some notes are supplied in Section 17.

3.2 Installing under UNIX-like operating systems

Move the profit executable to somewhere in your path (e.g. `~/bin/` or `/usr/local/bin`)

You should now create the environment variables `HELPDIR` and `DATADIR`. These should both point to the top ProFit directory where the files `ProFit.help` and `mdm78.mat` are stored. e.g.

```
(csh)   setenv HELPDIR /home/andrew/ProFitV3.1
        setenv DATADIR /home/andrew/ProFitV3.1
(sh)    export HELPDIR=/home/andrew/ProFitV3.1
        export DATADIR=/home/andrew/ProFitV3.1
```

`mdm78.mat` is the Dayhoff amino acid similarity scoring matrix while `ProFit.help` contains the help text displayed by the `help` command.

Alternatively, you may wish to store these files elsewhere, or have all help files and data files in a single directory.

Under VAX/VMS-like operating systems, these should be ASSIGNs. e.g.

```
ASSIGN $A:[ANDREW.PROFIT] DATADIR
ASSIGN $A:[ANDREW.PROFIT] HELPDIR
```

3.3 Compiling under DOS/Windows operating systems

ProFit V3.1 compiles under Windows using the open source `mingw` compiler. See the `mingw` web site for details: `http://www.mingw.org/wiki/HOWTO_Install_the_MinGW_GCC_Compiler_Suite` It should also compile cleanly using commercial compilers such as the Microsoft, Intel or Borland compilers (though this has not been tested).

To compile with `mingw`, first open a DOS shell and ensure that the `mingw` binary directory from your installation of `mingw` is in your path. For example:

```
PATH=%PATH%;C:\Qt\2009.01\mingw\bin
```

Now change to the ProFit source directory:

```
cd ProFitVx.y\src
```

Now run make by doing:

```
mingw32-make -f Makefile_dos
```

That will create the executable profit.exe

3.4 Installing under DOS/Windows operating systems

If you only wish to run ProFit from the DOS prompt command line, or you are using Windows 95/98/ME, you can edit `C:\autoexec.bat` and add the lines

```
PATH=%PATH%;C:\My Documents\ProFitV3.1\src
SET HELPDIR=C:\My Documents\ProFitV3.1
SET DATADIR=C:\My Documents\ProFitV3.1
```

(Note no double-inverted commas or escaping is required for spaces in directory names.)

This will put the profit executable in your path and set the two environment variables. Of course you can move the files anywhere you want and modify the above commands as required.

If you are using Windows NT/2000/XP or later, you must set environment variables as follows:

1. Open Control Panel
2. Click the System icon
3. Go to the Advanced pane
4. Click the **Environment Variables** button
5. Select the **new** button to create a new environment variable
6. Enter the variable name and value in the appropriate boxes, creating `HELPDIR` and `DATADIR` as above.
7. Edit the `PATH` variable, such that the directory in which you have saved `profit.exe` is added to your path (or move `profit.exe` to a directory already in your path).

Alternatively, if you only plan to run ProFit by double-clicking its icon, simply ensure that

```
profit.exe
mdm78.mat
ProFit.help
```

are all in the same directory. Double clicking the ProFit icon will then find the required files automatically.

3.5 The Rotation Bug

There is a known (but rarely seen) bug with ProFit where a fitted structure may be fitted 180 degrees away from its optimum fit. This only seems to affect fitting of identical structures and appears to result from a saddle point in the RMS surface resulting in apparent convergence. While the effort to correct the bug is ongoing we have taken steps to fix the effect of the bug.

Compiling with GCC with optimization on (-O3) seems to hide the bug. Alternatively, editing the Makefile and uncommenting the line

```
ROTATEREFIT = -DROTATE_REFIT
```

will result in ProFit rotating a fitted structure (42 degrees, Z axis), refitting the structure then selecting the better fit.

4 Starting the program

The program is started from the command line by typing the command:

```
profit
```

Once the program is started, you may read in structures to be fitted. Alternatively, the PDB files may be specified on the command line:

```
profit reference.pdb mobile.pdb
```

By default, **ProFit** does not read HETATM records from the PDB file. This may be changed from the command line by using the -h flag:

```
profit -h
profit -h reference.pdb mobile.pdb
```

Alternatively, once in the program you may give the `HETATOMS` command before reading in the structures (see Section 5).

ProFit can run a script file, a text file of **ProFit** commands, from the command line using the -f flag:

```
profit -f myscriptfile.txt
profit -f myscriptfile.txt -h reference.pdb mobile.pdb
```

Once in the program you may use the command `SCRIPT myscriptfile.txt` to run a script file (see Section 14).

If compiled with XMAS[1] file support, the `-x` flag may be used to specify that the files named on the command line are XMAS files instead of PDB files. Note that the program currently will only write PDB format files.

```
profit -x reference.xmas mobile.xmas
```

If compiled with GUNZIP support then the program can read gzipped PDB files. This will only work on unix-like platforms and assumes that the `gunzip` program is in your path. Note that the uncompressed files will remain in `/tmp` with a name like `readpdb_12345` where 12345 is a process number. You will need to delete these regularly!

Once in the program, you issue commands by typing at the keyboard. These commands may always be abbreviated to the minimum non-ambiguous string. The program is mostly case insensitive; you may mix upper and lower case at will, though uppercase will be used throughout this documentation. The only times that **ProFit** will be case sensitive are when dealing with file names or lowercase chain identifiers (see Section 9).

Support for the GNU Readline library[2] was introduced in **ProFit** V3.0 allowing you to edit the command line and to recall previous commands. Again, this is a compile time option.

You exit from the program by typing `QUIT`.

5 Reading Structures

ProFit reads files in PDB format. If compiled with XMAS support, then XMAS files may also be read, but only PDB files may be written. It uses the concept of a *reference* structure and a *mobile* structure. The reference structure remains static

[1] XMAS is an XML-like file format developed at Inpharmatica, Ltd. which is designed for leaf-heavy data such as protein structure data

[2] The GNU Readline library is available from http://directory.fsf.org/project/readline/

in space and the mobile structure is fitted onto it. When the files are specified on the command line, the reference structure is specified first, the mobile structure second.

Once in the program, you may read the reference structure using the `REFERENCE` command and the mobile structure using the `MOBILE` command. Using these commands causes the equivalent current structure to be deleted from the program's memory first. However, any zone and atom specifications (see Sections 8 and 9) are not deleted. For example, you can read p3hfl.pdb as a new reference structure using the command:

```
REFERENCE p3hfl.pdb
```

and read p3hfm.pdb as a new mobile structure with:

```
MOBILE p3hfm.pdb
```

If compiled with XMAS support, then the XMAS format is specified by placing the keyword `XMAS` after the commands `REFERENCE` or `MOBILE`:

```
REFERENCE XMAS p3hfl.pdb
MOBILE XMAS p3hfm.pdb
```

When you read a structure containing insertions, you will receive a warning message to this effect. This dates from when the program was unable to handle residue specifications containing insertion codes, but is still useful to draw your attention to the fact that they are present.

Note that atoms with coordinates of 9999.00, 9999.00, 9999.00 will be ignored during all calculations allowing atoms with undefined coordinates to be handled.

When fitting multiple structures (new in **ProFit** V2.0), you use the `MULTI` command to read in the structures. See Section 10.

6 Getting Help

Note: *For the help facility to work, you must have the* `ProFit.help` *file either in the current directory or in a directory pointed to by the environment variable* `HELPDIR`*. See Section 3 or the* `INSTALL` *file for details.*

To get help within the programs simply type `HELP` and you will be presented with a list of commands which the help facility knows about. The `ProFit>` prompt will also change to `Help>`. You may then type the name of a command to get help on that command. Typing `HELP` once in the help facility will repeat the list of available help topics. Like the main command interface, the help facility will accept upper or lower case and you may abbreviate commands. If your abbreviation is ambiguous (i.e. more than one command starts with the letters you have specified), help will be supplied on all the commands which match[3].

If the help text is longer than 21 lines, you will see a prompt saying

```
More...
```

in which case you should hit the Return (or 'Enter') key to get the next page of help.

Once at the `Help>` prompt, you should simply hit the Return (or 'Enter') key to get back to the main `ProFit>` prompt.

If you know the topic on which you need help, you may type the name of the command after the `HELP` keyword at the main `ProFit>` prompt. After the help message is printed, you will be returned directly to the `ProFit>` prompt. For example, if you want help on the `ZONE` command, you may type:

[3]The one exception to this is if the letters you supply are an abbreviation of `HELP`, when the list of help topics will be shown again.

```
HELP ZONE
```

Allied to help, is the `STATUS` command. This tells you the current status of the program: what structures are loaded, fitting zones, atoms and the like.

7 Fitting Structures

Having read in a reference and a mobile structure, you actually fit them by giving the `FIT` command. When you do this, you will get a message like:

```
Fitting structures...
RMS: 0.366
```

However, this will only work if the two structures are of identical composition i.e. if the sequences are the same and the same atoms are present in both. If there are any mismatches, the first such mismatch will be reported and the RMS deviation will not be calculated.

Since you will frequently need to fit non-identical structures, you may use the `ZONE` and `ATOMS` commands to specify which residues should be considered equivalent and which atoms should be considered in the calculation.

If you are using zone or atom specifications, the RMS deviations will be displayed over the atoms and zones specified in those commands.

Normally the fitting procedure will not be completed if there are any mismatched atoms or atoms missing from one of the two structures. The program issues an error message about atoms missing in the mobile structure which are found in the reference structure. The `IGNOREMISSING` command causes the program to issue a warning instead of an error and the fitting proceeds ignoring the mismatched atoms. The default behaviour is restored by using the `NOIGNOREMISSING` command.

8 Specifying Atom Subsets

The `ATOMS` command is used to specify a subset of atoms to be used in the calculations. It has the syntax:

```
ATOMS atm[,atm]...
```

i.e. you specify the `ATOMS` keyword followed by one or more atom names separated by commas. A `*` may be used to specify all atoms and a `~` or `^` may be placed at the beginning of the specification to inverse the selection. For example, to fit only Cα atoms:

```
ATOMS CA
```

to fit N, Cα, C and O atoms:

```
ATOMS N,CA,C,O
```

to fit sidechains only (i.e. everything except N, Cα, C and O atoms):

```
ATOMS ^N,CA,C,O
```

to return to fitting all atoms:

```
ATOMS *
```

The PDB atom name field is 4 characters wide followed by a space. The first two characters are the right-justified element type, so for normal protein and DNA atoms consist of a space followed by a N, C, O, S or P. Thus the atom name field for a Cα contains ' CA '. HETATMs such as calcium will contain the two characters CA in the first two fields. i.e. 'CA '. When you specify an atom type it is matched against the atom name field from the *second character onwards*, unless you preceed it with a `<`. Thus to match a Cα you use `CA`, but to match Calcium, you use `<CA`. For example, as stated above, to match Cα atoms:

```
ATOMS CA
```

while to match calcium atoms

```
ATOMS <CA
```

and to match both Cα and calcium:

```
ATOMS <CA,CA
```

Wildcards are also allowed. A `%` or a `?` may be used to match a single letter at any point in the specification while a `*` may be used to match all remaining characters (thus `C*` is allowed, but `*G` is not). These special characters may be escaped by preceding them with a `\`. For example to fit all carbons:

```
ATOMS C*
```

or to match all atoms at the γ position:

```
ATOMS ?G*
```

and to match the `C4*` atoms in DNA:

```
ATOMS C4\*
```

If atom names contain spaces (e.g. in heme groups) the whole atom specification must be enclosed in double inverted commas:

```
ATOMS "N A,N B,N C"
```

9 Specifying Zones

The `ZONE` command is used to specify zones in the two structures which are considered equivalent. The complete syntax for the command is:

```
ZONE CLEAR|((*|(X...[,n][/m])|(j-k))[:(*|(X...[,n][/m])|(j-k))])
```

where `X...` is an amino acid sequence, `n` is a number of residues, `m` is the occurrence number, `j` and `k` are residue specifications of the form *[chain][.]resnum[insert]*. Items in square brackets are optional and alternatives are marked by a `|` and grouped in parentheses.

`ZONE` commands are cumulative. Thus each zone you specify is added to those currently active. To clear all zones (i.e. fit all residues), the `ZONE CLEAR` or `ZONE *` command may be given. To clear a single zone, the DELZONE command can be used (see the end of this section).

When a new zone is added, a warning message is displayed if the new zone overlaps an existing zone. Overlapping zones will be flagged with `*` when using the `STATUS` command.

Although it appears complex, the syntax is actually very simple and consists of two identical sections separated by a colon (:). The left half is applied to the reference structure and the right half to the mobile structure. In its simplest form, the right hand half of the expression is absent and the specification is applied to both reference and mobile structures. For example:

```
ZONE 24-34
```

will set the zone to include residues 24–34 in both structures. If you wanted to fit 24–34 in the reference structure with 25–35 in the mobile structure, this simply becomes:

```
ZONE 24-34:25-35
```

Single residues can be specified using the same syntax:

```
ZONE 44-44:55-55
```

You may also specify chain names and insertion codes. The chain name is placed before the residue number and the insertion code afterwards. For example:

```
ZONE L25A-L30
```

fits residues 25A–30 in the L chain of both structures. Optionally, the chain name may be separated from the residue number using a full stop. For example:

```
ZONE L.25A-L.30
```

Using the full stop also makes the statement case-sensitive. In practice, the full stop separator is used with numeric chain names to separate the chain name from the residue number and with lowercase chain names.

```
ZONE 1.25-1.30
ZONE b.1-b.60:A.1-A.60
```

Simple wildcards may also be used. For example

```
ZONE H*:B*
```

fits the reference H chain with the mobile B chain,

```
ZONE -10:50-59
```

fits from the first residue to residue 10 in the reference structure with 50–59 in the mobile structure.

```
ZONE *:1-100
```

fits all residues in the reference structure with 1-100 in the mobile structure.

If the structure file contains negatively numbered residues and you are using residue numbering, you can escape the minus sign in the residue number using a backslash:

```
ZONE \-4-10:\-1-13
```

will fit residues -4 to 10 in one structure with -1 to 13 in the other.

Alternatively, you may specify the zones to be fitted by giving a sequence fragment. Together with that fragment, you may specify the number of residues to consider starting at that point. If the fragment occurs more than once in the sequence you may specify which occurrence you wish to consider. For example:

```
ZONE CAR:VNS
```

fits the first occurrence of CAR in the reference set with first occurrence of VNS in the mobile set;

```
ZONE CAR,10:VNS,10
```

fits 10 residues starting at the first occurrence of CAR in the reference set with 10 residues from the first occurrence of VNS in the mobile set;

```
ZONE CAR,5/2
```

fits 5 residues from second occurrence of CAR in both structures;

```
ZONE 24-34:EIR,11
```

fits 24-34 in the reference set with 11 residues starting at the first occurrence of EIR in the mobile set.

By default, **ProFit** works in 'Residue Number' mode, i.e. the numbers used in zone commands are the numbers seen in the PDB file. The alternative mode is 'Sequential' mode where residues are numbered sequentially throughout the structure (including throughout multiple chains). Any chain names appearing in zone specifications will be ignored in Sequential mode. To switch mode, you use the `NUMBER SEQUENTIAL` or `NUMBER RESIDUE` commands.

The `DELZONE` command specifies zones to be deleted from the user-defined list of fit zones. `DELZONE` uses the same syntax as the `ZONE` command. The command matches the specified zone with a zone in the user-defined list of fitting zones and deletes the matching zone from the list. Entering either `DELZONE ALL` or `DELZONE *` will delete all user-defined zones.

9.1 Sequence Alignment

Note: *For sequence alignment to work, you must have the* `mdm78.mat` *file either in the current directory or in a directory pointed to by the environment variable* `DATADIR`*. This is the Dayhoff amino acid similarity scoring matrix. See Section 3 or the* `INSTALL` *file for details.*

Another way of specifying zones is to let the program do it. **ProFit** allows you to perform a simple Needleman and Wunsch sequence alignment and to apply zones automatically derived from that sequence alignment. This is done by issuing the `ALIGN` command. The sequence alignment is displayed, any currently active fitting zones are cleared and replaced by zones derived from the alignment. Additional zones may also be specified in the usual way.

As of Version 3.0, **ProFit** offers a choice of three alignment options:

1. The default alignment option is a chain-by-chain alignment where the first chain in the mobile is aligned with first chain in the reference, the second chain in the mobile is aligned with the second chain in the reference, and so on. If the number of chains does not match then a warning is issued.
2. The `ALIGN WHOLE` command gives a whole sequence alignment. The whole sequence (regardless of chain ID) is aligned. If the fitting zones assigned in this manner extend over more than one chain the zones are split into smaller zones at the breaks between chains. This may be useful if a sequence has been split into fragments.
3. If a zone definition is supplied to the `ALIGN` command then **ProFit** will perform an alignment over the defined region to assign fitting zones. (See Section 9 for the syntax for defining zones.)
   It is also possible to append new zones onto the end of the zone list (rather than overwriting the current zone list) by adding `APPEND` after the zone definition. For example one could use following commands:

   ```
   ALIGN A*:B*
   ALIGN B*:A* APPEND
   ```

to align chain A with chain B and then B with A. This is useful when chains appear in different orders in the PDB files.
When doing multiple fitting, it is not possible use the colon notation to define regions on both the reference and mobile structures. This is the same restriction as the `ZONE` command (see Section 10.1).

Clearly, it will normally be necessary to use the `ATOMS` command to specify that only backbone or Cα atoms are included in the fitting. The `TRIMZONES` command can also be used when doing multiple structure fitting to ensure that the fitting zones are identical for all mobile structures. (See Section 10.1)

The `GAPPEN` command allows you to specify an integer gap penalty and gap extension penalty for the sequence alignment performed by the `ALIGN` command. The default values for the gap penalty and gap extension penalty are 10 and 2 respectively.

9.2 Reading an Alignment

If you have an alignment performed outside **ProFit** you may use this to specify the equivalent zones. Any previously defined fitting zones are automatically cleared first. As of **ProFit** V3.0, the `READALIGN` command can be used with structures having more than one chain.

The alignment should be a file in PIR format using – characters to align the sequences. The two sequences are represented by separate entries, i.e. each must have a header of the form:

```
>P1;xxxxxx
title text .......
```

When reading an alignment file for aligning a reference structure with a single mobile structure, the first sequence will be assumed to be that of the reference structure and the second is that of the mobile structure. Any other sequences in the file are ignored. Chain Breaks in a sequence are indicated with a `*`.

```
>P1;REFSEQ
Reference Sequence - first.pdb
WILLIAM*H-ARTNELL-*

>P1;M_0001
Mobile Sequence - second.pdb
--PATRI-K*TR--GHTN*
```

The `READALIGNMENT` command is also used to read in the PIR files containing a multiple sequence alignment. When performing a multiple structure fit, the first sequence *must appear twice* in the sequence alignment file. This is because it is used as both the initial reference and first mobile set:

```
>P1;REFSEQ
Reference Sequence - first.pdb
----WILLIAM*H-ARTNELL-*

>P1;M_0001
Mobile Sequence - first.pdb
----WILLIAM*H-ARTNELL-*

>P1;M_0002
Mobile Sequence - second.pdb
```

```
------PATRI-K*TR--GHTN*

>P1;M_0003
Mobile Sequence - third.pdb
PERTWEE--------------*
```

Note that a bug in using the READALIGNMENT with multiple structure fitting was fixed in V2.3. (The bug caused the program to crash if a deletion appeared in the same place in two or more of the sequences.)

9.3 Limiting Zones Read From an Alignment

When obtaining fit zones from a sequence alignment, either from ALIGN or from READALIGNMENT, it can be useful to limit the zones of residues used. Normally all aligned residue pairs will be used.

For example, if the alignment were:

```
          1         2         3
12345678901234567890123456789012345
ASAHSTGEHNM--PLELLGHISLAM---NPRTY
---HSTADHNLRTPLEVLG--SLAMEDRQPRTY
```

the zones would normally be taken from the following positions in the alignment: 4-11, 14-19, 22-25, 29-33

By using the command:

```
LIMIT 20 28
```

only the zone from 22-25 would be included.

This is particularly useful in conjunction with the ITERATE command (Section 9.4) and when fitting multiple structures (Section 10).

The LIMIT OFF command restores the default behaviour of deriving the zones from the whole alignment.

9.4 Iterative Updating of the Fitting Zones

The ITERATE command switches on the iterative updating of fitted zones during subsequent FIT commands. The ITERATE command may be followed by an optional parameter to specify the cutoff used to include or exclude pairs from the zones. (ITERATE OFF is used to switch it off again.)

Note that this immediately does an ATOMS CA since iteration of zones is only performed on Cα atoms. The program gives an informational message to this effect. See notes below if you want to calculate an RMSd over other atoms.

After the initial fit on the specified zones, the zones are updated such that residue pairs with Cα atoms within a specified cutoff (default 3.0Å) are included and those more distant are excluded. The optimum set of equivalences is obtained using a dynamic programming method.

After updating the zones, the structures are refitted and the procedure iterates to convergence of < 0.01Å, (typically 3 or 4 cycles). The RMSd on Cα atoms is shown after each cycle unless the QUIET command is given before running ITERATE.

You may specify a minimal initial zone of say 3 amino acids on which to fit first. The zone iteration will expand the zones until as many residues as possible can be equivalenced. Alternatively, this option is particularly useful in conjunction with the ALIGN command. Using ALIGN followed by ITERATE gives a particularly convenient method of fitting two arbitrary structures.

As stated above, the `ITERATE` command implies `ATOMS CA`. Having fitted on Cα atoms, you can of course display the RMSd over other atom sets in the usual way using the `RATOMS` command (e.g. `RATOMS N,CA,C,O` will display the backbone RMSd).

Should you wish to refit on another atom set using the iterated zones, simply use `ITERATE OFF` to switch off iteration, select the atom set required using the `ATOMS` command and use `FIT` to refit the structures in the usual way. For example, to fit on backbone atoms:

```
ITERATE OFF
ATOMS N,CA,C,O
FIT
```

9.5 Fitting Zones based on the Temperature Factor Column.

Note that this use of the B-value column is not compatible with the commands described in Section 13.

It is possible to define zones by flagging residues in the temperature factor column of the PDB file using the `BZONE` command. Zones are marked using a positive whole numbers while zeros are ignored. Multiple zones can be marked using additional numbers. So, residues with the B-factor set to 1 will be fitted with one another, residues with the B-factor set to 2 will be fitted with one another, etc.

Assignment of zones is carried out in two ways:

If only the reference structure is marked then the same set of residue numbers will be added as a fitting zone in both the reference and mobile structure.

If both the reference and the mobile structure are marked then fitting zones are assigned by scanning through and setting zones for corresponding continuous stretches of flagged residues in either the reference or mobile structures.

9.6 Centre of Fitting

The default method for fitting is to centre the fit around the centre of geometry of the fit atoms. Alternatively, fitting can be centred around the centre of geometry of a residue specified by the `SETCENTRE` (or `SETCENTER`) command.

```
SETCENTRE CLEAR|(*|i[:j])
```

where i and j are residue specifications of the form *[chain][.]resnum[insert]*. Items in square brackets are optional and alternatives are marked by a `|` and grouped in parentheses.

The command:

```
SETCENTRE 24:35
```

will centre the fit around residue 24 of the reference structure and residue 25 of the mobile structure. The mobile residue number can be omitted. For example:

```
SETCENTRE 33
```

will centre the fit around residue 33 of the reference structure and residue 33 of the mobile structure.

Entering `SETCENTRE CLEAR` or `SETCENTRE *` will clear the centre residue.

9.7 Distance Cutoff for RMSd Calculations

The DISTCUTOFF command specifies a distance cutoff for ignoring atom pairs outside a specified distance when calculating RMSd.

```
DISTCUTOFF [cutoff|ON|OFF]
```

The DISTCUTOFF command specifies a distance cutoff for ignoring atom pairs outside a specified distance when calculating RMSd. Entering DISTCUTOFF ON or DISTCUTOFF OFF will turn the distance cutoff on or off. Entering DISTCUTOFF 2.5 will set the value of the distance cutoff to 2.5 Angstroms and turn the distance cutoff on. A warning is displayed if the distance cutoff is set to zero and turned on. Note that the cutoff is only applied to the final calculation of RMSD and not to the fitting.

10 Multiple Structure Fitting

The MULTI command allows a multiple set of structures to be read in for fitting. The filename specified for MULTI is a 'file of files' i.e. it contains a list of filenames which will be read.

MULTI is used in place of REFERENCE and MOBILE to read in a set of structure files. The first structure file is used as a reference set for the first fitting stage, but the coordinates are averaged after each fitting stage to derive an averaged template used for subsequent fitting.

i.e. Given N files to fit, file 2 is fitted to file 1 and an averaged structure, A, is calculated, file 3 is then fitted to A and a new average, A' is calculated. This continues until all N structures have been fitted. The whole procedure iterates until convergence (typically 3 or 4 cycles).

ProFit V3.0 changes the default method of calculating the average template. As each new mobile structure is added, the degree of change in the averaged structure is inversely proportional to the total number of mobile structures. Consequently, outlying structures should have less effect on the averaged reference structure.

Normally, the coordinates of the first structure in the MULTI list are taken as the starting point for the averaged reference structure. It is possible however, to select another mobile structure as the initial reference structure using the SETREF command. For example, SETREF 3 will use the third mobile structure as the reference strcture. If no structure number is specified, then the SETREF command carries out an all vs. all comparison and the coordinates of the mobile structure with the least overall RMSD to all the other mobile structures are selected as the initial reference structure.

Multiple structures can be fitted with either the FIT or ORDERFIT command. The ORDERFIT command (new in V3.0) will perform the multiple structure fit in a similar manner to the FIT command but fitting the most similar structures first. As the averaged template is updated with each new structure fitted, the order of fitting has a (small) influence on the template. The ORDERFIT command (possibly along with the SETREF command) can provide a standardized fitting scheme.

Progress and RMSds are reported at each iteration unless the QUIET command is used.

By default, RMSDs, pairwise distances and transformation matrices are given in relation to the first mobile structure. The MULTREF command will set **ProFit** to give results in relation to the averaged reference structure rather than the first mobile structure (MULTREF OFF restores the default behaviour).

The resulting fitted files are written with the MWRITE command. Note that there is no "reference" set in the sense used for normal 2-structure fitting; fitted versions

of all N files will be written since the reference set is actually an averaged template used purely as a guide for fitting.

The averaged template can be written to a file using the `WRITE REF` command. As it is a simple numerical average of the cartesian coordinates however, taking the reference structure generated by ProFit as a representation of an actual geometry/conformation accessible by the structure should be done with caution.

When the `MWRITE` command is used, the output filenames are the same as the input files, but with the extension replaced by that specified in the `MWRITE` command. If no extension is specified, then '.fit' will be used. If the input structure files contained no extension, then the extension specified will be appended to the filenames.

Note that since only the extension is changed when writing back the fitted files, you must have permission to write to the directory from which the original files were read.

Multiple-structure fitting is particularly effective in combination with the `ITERATE` command (see Section 9.4) which refines the fitting zones iteratively. This can lead to extremely good multiple structures fits.

Note that multiple structure fitting and zone iteration can be very slow as these have been added to the earlier pair-wise fitting engine. An increase in speed needs a complete re-design of the code.

10.1 Specifying Zones With Multiple Structure Fitting

Currently, the ZONE command may only be used with multiple structure fitting when the same zone specification may be applied to every structure. i.e. You cannot specify a zone for each structure separating the zones with a colon (:)

Thus, the following are legal zones:

```
ZONE 20-30
ZONE C,3
```

while the following are not:

```
ZONE 24-34:25-35
ZONE CAR:VNS
ZONE 24-34:EIR,11
```

For normal use, it is recommended that the `ALIGN`, `TRIMZONES` and `READALIGNMENT` commands (possibly in conjunction with the `LIMIT` command) are used for specifying zones when fitting multiple structures.

As of **ProFit** V3.0, the `TRIMZONES` command can be used in conjuction with the `ALIGN` command. The `ALIGN` command performs a pairwise alignment for each of the mobile structures with the reference. Although fitting each mobile using an individualized set of zones offers the best fitting for each mobile to the reference, there may be times when a like vs. like comparison is required. If the number of residues used for fitting varies, RMS deviation cannot be directly compared between structures.

To allow for a like vs. like comparison, the `TRIMZONES` command resets the fitting zones for each mobile structure to include only fitting residues that are common to all the mobile structures. Thus, by ensuring that the fitting zones are the same for each mobile, the `TRIMZONES` command allows for a like vs. like comparison.

When using `READALIGNMENT` with multiple structures, the first sequence *must appear twice* in the alignment file. This is because it is used as both the first reference and mobile set.

Note that a bug in using the READALIGNMENT with multiple structure fitting was fixed in V2.3. (The bug caused the program to crash if a deletion appeared in the same place in two or more of the sequences.)

10.2 All Versus All Comparisons

As of **ProFit** V3.0, it is possible to perform an all versus all comparison of the mobile structures when fitting multiple structures. The ALLVSALL command requires that the fitting zones set are identical for all mobile stuctures and automatically resets the fitting zones using the TRIMZONES command.

Results are presented as tab-delimited text suitable for loading into a spreadsheet. If the optional filename parameter is given, output is directed to the specified file. If a filename is not specified, or the file cannot be opened, output appears on the screen. If the filename begins with a pipe character (|), the results are piped into the specified program. This is particularly useful with the more (or less) Unix command.

11 Calculating the RMSd Over Other Zones and Atoms

Having fitted the structures using the ZONE and ATOMS commands to specify which residues and atoms should be included in the fitting, the RMS deviation may then be calculated over a different region of the structure and/or a different atom set.

This is achieved using the RZONE and RATOMS commands. The syntax of these commands is identical to that of the ZONE and ATOMS commands described in Sections 8 and 9.

As each RZONE or RATOMS command is given, the RMS deviation is reported over the new set of zones or over the new atom set. Don't forget the RZONE commands are cummulative, like the ZONE commands. Note that the RZONE * or RZONE CLEAR behaves slightly differently from ZONE * or ZONE CLEAR since it resets the zones to be the same as those specified for the fitting using ZONE, ALIGN or READALIGNMENT commands.

The DELRZONE command specifies zones to be deleted from the list of user-defined zones for calculating the RMSd. The DELRZONE command uses the same syntax as the DELZONE command. The command matches the specified zone with a zone in the user-defined list of RMSd calculation zones and deletes the matching zone from the list. Unlike the RZONE command, entering either DELRZONE ALL or DELRZONE * will delete all user-defined RMSd calculation zones rather than returning to the default condition where the RMSd calculation zones are set to the user-defined fitting zones. Thus avoiding the somewhat counterintuitive situation where deleting the last RMSd zone restores all RMSd zones. If no RMSd calculation zones are defined then **ProFit** will calculate the RMSd over all residues.

12 Obtaining Output

12.1 The Fitted Structure

The fitted mobile structure may be written to a file in PDB format using the WRITE command:

```
WRITE fitted.pdb
```

If the first character of the filename is a pipe character (|), then the results will be piped into the specified program. For example:

```
WRITE |less
```

will cause the coordinates to be displayed on the screen using the `less` pager program.

The reference set may also be written:

```
WRITE REFERENCE ref_fitted.pdb
```

(only the three letters 'REF' of the REFERENCE parameter are required). This is only useful if the CENTRE command has been used (see below).

12.2 Centering the Coordinates

By default, the mobile structure is moved to the coordinate frame of the reference set. If the `CENTRE` (or `CENTER`) command is given then the centre of geometry of the fitted coordinates will be located at the origin.

If a residue has been set as the centre of fitting using `SETCENTRE` (see Section 9.6) then that residue will be moved to the origin when the `CENTRE` command is used.

If only two structures are fitted then the `WRITE REFERENCE` command must be used to write the reference set in the origin-centred coordinate frame. If multiple structures are fitted and written using `MWRITE` then the reference set will be written automatically.

12.3 Details of the Fitting

More details about the fitting may be obtained by using the `MATRIX` command. This displays the centres of geometry, the rotation matrix and the translation vector which is the vector between the centres of geometry. Thus to superimpose the mobile structure onto the reference structure using these data, you should translate the mobile set to the origin, apply the rotation matrix, translate back to the original centre of geometry and finally apply the translation vector.

Note that the rotation matrix is not orthogonal and cannot therefore be used to extract Euler angles. This is a result of the fitting method used.

The `NFITTED` command displays the number of atom pairs which were fitted in the last fitting operation. Note that this will not be the number of residues fitted unless you are only fitting one atom type per residue (typically Cα atoms).

12.4 By-residue RMS Deviation

The `RESIDUE` command is used to obtain a by-residue RMS deviation on the currently specified RMS atoms in the currently specified RMS zone. If no `RATOMS` and `RZONE` commands have been used, the atoms and zones used for the fitting will be used.

The `RESIDUE` command may be followed by an optional filename parameter in which case output is directed to the specified file. If the file cannot be opened or a filename is not specified, output appears on the screen. If the first character of the filename is a pipe character (|), then the results will be piped into the specified program. For example:

```
RESIDUE |less
```

will cause the results to be displayed on the screen using the `less` pager program.

If the distance cutoff is set then residues fully outside the distance cutoff are flagged with `**` and residues partially outside the distance cutoff are flagged with `*` (see Section 9.7)

The related command, PAIRDIST prints the pairwise distances between equivalent atom pairs in the reference and mobile structures. PAIRDIST has the same syntax as RESIDUE. If the distance cutoff is set then residues outside the distance cutoff are flagged with *.

12.5 Outputting Fit Zones As An Alignment

As of **ProFit** V3.0, it is possible to output the equivalenced regions found by iterative fitting as an alignment using the PRINTALIGN command. The default output is a (user-friendly) pairwise alignment with the reference and mobile sequences printed as pairs of 60-character wide lines. The optional FASTA and PIR parameters set the printout to (machine-friendly) FASTA or PIR formatting for the chain names and sequences.

Alignments can be exported to a text file using the PRINTALIGN command. **ProFit** V3.0 can read PIR formatted files for assigning zones.

Note: For a set of fit zones to be converted into an alignment, the fitting zones must occur sequentially along the protein chain. Additionally, the fit zones cannot overlap. In other words, to obtain a sequence alignment the fitting zones must be in sequence.

13 Modifying the Fit

The commands described in this section make use of the temperature factor column as a 'flag' and are therefore not compatible with the BZONE command (Section 9.5).

Normally, no weighting is applied during the fitting i.e. all atoms are weighted equally. The WEIGHT command causes the fitting to be weighted by the mean of the B-values in the equivalent atoms. Normally, you wouldn't use this with real B-values, but with some other weight parameter (e.g. SSAP scores).

The BWEIGHT command weights the fitting by the inverse of the mean of the B-values in the equivalent atoms. This is useful for genuine weighting by B-values (i.e. the mobile set atoms will be less heavily weighted).

The NOWEIGHT command switches off weighting.

Atoms can also be removed from consideration in the fitting and RMS deviation calculations using temperature factors as a cutoff. The BVALUE command allows you to specify a B-value cutoff and any atoms with B-values greater than this value will be *ignored completely* in both the fitting and RMS deviation calculations. The B-value may not be higher than this value in either the reference set or the mobile set. For example, if you specify 10, then atoms with B-values greater than 10 will be ignored.

By specifying a negative value for BVALUE, you require that any atoms with B-values less than the absolute value you specify will be ignored. For example, if you specify -10, then atoms with B-values less than 10 will be ignored.

The value may be followed by an optional REF or MOB parameter which restricts checking of B-values to the specified structure.

14 Script Files

While it is possible to run a script from the unix command line using a redirection operator (<) or pipe (|), there are occasions when this is problematic such as when running **ProFit** from within another application. It is possible to use a command line flag to run a script file.

For example, a script file can be run using either a command line flag:

```
profit -f myscriptfile.txt -h reference.pdb mobile.pdb
```

By using the redirection operator:

```
profit -h reference.pdb mobile.pdb < myscriptfile.txt
```

Or by piping input from another program:

```
cat myscriptfile.txt | profit -h reference.pdb mobile.pdb
```

All three options produce identical outputs.

It is also possible to run a script from within **ProFit** using the SCRIPT command:

```
SCRIPT myscriptfile.txt
```

When a script file is run, messages indicating the start and end of the script are sent to stdout, if quiet mode is off. A comment marker (`#`) at the beginning of a line will echo the line to stdout, a useful method for annotating an output file when running non-interactively.

Finally, it is possible to run a script from within a script using the `SCRIPT` command. **ProFit** tracks the number of open/nested scripts and will allow up to 1000 nested scripts to be open. The assumption is that if over a thousand scripts are open then **ProFit** has been sent into an infinite loop (for instance by having a script call itself).

15 Miscellaneous Commands

The `RMS` command may be used to reprint the RMS deviation over the currently defined set of RMS zones and RMS atoms.

If you simply wish to calculate the RMSd between two or more structures without actually fitting them, defining fitting regions in the normal way then typing the `NOFIT` command (instead of the `FIT` command) will set up **ProFit** to perform RMSd calculations but will not fit the structures. The `RMS` command can then be used to print the RMS deviation.

As of **ProFit** V3.0 it is possible to match symmetrical atoms automatically in amino acid sidechains (e.g.CD1 - CD2 and CE1 - CE2 of tyrosine) using the `SYMMATOMS` command. `SYMMATOMS` matches the charged oxygens and nitrogens on arginine, aspartate and glutamate residues and the delta and epsilon carbons of phenylalanine and tyrosine residues. It is also possible to match the nitrogen and oxygen atoms of the amide sidechains of asparagine and glutamine residues and the prochiral methyl groups of valine and leucine. Typing `SYMMATOMS` will display the pairs of atoms currently matched by **ProFit**. Typing `SYMMATOMS ON` or `SYMMATOMS OFF` will turn symmetrical atom matching on or off. Individual residue types, for example ASP, can be turned-on or off by typing SYMMATOMS ASP ON or SYMMATOMS ASP OFF, respectively. Alternatively, SYMMATOMS ALL ON will turn all atom pairs on. By default, the matching of symmetrical atoms is turned-off.

Any operating system command may be run from within **ProFit** by preceding it with a `$`. The string following the `$` is passed to the operating system exactly as given and is useful for obtaining directory listings, typing, editing or copying files.

16 Command Summary

$ ***command*** Passes command to the operating system.

 # *comment* Echoes comment to stdout.

ALIGN [[**WHOLE**|***]|[***zonespec*** [**APPEND**]]] Performs Needleman and Wunsch sequence alignment on the sequences of the two structures and derives zones from the equivalent regions in the alignment. For multiple structure fitting, `ALIGN` performs a pairwise alignments for the reference sequence and each mobile sequence.

It will normally be necessary to use the ATOMS command to specify that only backbone or C-alpha atoms are included in the fitting calculations.

ALLVSALL [***filename***] Performs an all versus all comparison of the mobile structures when fitting multiple structures. Results are presented as tab-delimited text suitable for loading into a spreadsheet.

If the optional filename parameter is given, output is directed to the specified file. If the file cannot be opened or a filename is not specified, output appears on the screen. If the filename begins with a pipe character (|), the results are piped into the specified program.

ATOMS ***atm[,atm]...*** Specifies the atom subset to fit.

BVALUE ***cutoff*** [**REF**|**MOB**] Specify a B-value cutoff. Any atoms with B-values greater than this value will be ignored completely. A negative cutoff specifies that atoms with B-values less than the absolute cutoff should be ignored. The optional `REF` or `MOB` parameter restricts B-value checking to the specified structure.

BWEIGHT Weight the fitting by the inverse of the mean of the B-values in the equivalent atoms.

BZONE Sets fitting zones based on markers in the temperature factor (B-value) column

CENTER [**OFF**] See `CENTRE`.

CENTRE [**OFF**] Cause the coordinates to be written (using the `WRITE` or `MWRITE` commands), with the centre of geometry located at the origin instead of in the same coordinate frame as the reference set.

DELRZONE ***zonespec*** Removes a zone specification to the list of zones considered in RMS deviation calculation. `DELRZONE *` or `DELRZONE ALL` deletes all RMS deviation calculation zones.

DELZONE ***zonespec*** Removes a zone specification to the list of zones considered in fitting. `DELZONE *` or `DELZONE ALL` removes all zone specifications.

DISTCUTOFF [***cutoff*** | **ON** | **OFF**] Specifies a distance cutoff for RMSd calculations.

FIT Performs the actual fitting. Returns the RMS deviation over the atoms included in the fit.

GAPPEN ***val*** [***val***] Specifies an integer gap penalty and a gap extension penalty for the sequence alignment performed by the `ALIGN` command. The default values for the gap penalty and gap extension penalty are 10 and 2 respectively.

HEADER [**ON** | **OFF**] Include PDB header and trailer records when writing structures. By default, only the coordinate section of a file is output when a structure is written.

HETATOMS Read HETATM records with subsequent `MOBILE` and `REFERENCE` commands.

IGNOREMISSING Ignore any atom mismatches and proceed with the fitting. Such atoms are listed as warnings.

ITERATE [**(*limit* | OFF)**] Switches on (or off) iterative updating of the zones for fitting. The `ITERATE` command may be followed by an optional distance cutoff (default: 3.0Å) or by the keyword 'OFF' to switch off iterative zone calculation.

LIMIT (*pos1 pos2* | OFF) Limits the range in an alignment (from `READALIGNMENT`) used to derive zones. `LIMIT OFF` restores the default behaviour.

MATRIX Displays the centres of geometry, rotation matrix and translation vector.

MOBILE [**XMAS**] ***filename*** Reads a mobile PDB structure. If compiled with XMAS support, then the XMAS keyword specifies that the input is in XMAS format.

MULTI *filename* Reads a file of files containing a list of structures for multiple fitting.

MULTREF [OFF] Sets RMSd calculations to give values to the averaged reference rather than the first mobile structure.

MWRITE [***ext***]

Write the results of multiple structure fitting. The structures are written back using the same filenames with which they were read, but with the extension changed to that specified. If no extension is given, then '.fit' is used. Note therefore, that you must have write permission to the directory from which the input files were read.

NFITTED Reports the number of atom pairs fitted.

NOFIT Sets the fitted flag in profit allowing the user to calculate the RMSD on a structure without fitting.

NOHETATOMS Do not read HETATM records with subsequent `MOBILE` and `REFERENCE` commands.

NOIGNOREMISSING Restore the default behaviour of issuing an error message for any atom mismatches and halting the fitting proceedure.

NOWEIGHT Normal, non-weighted fitting.

NUMBER (RESIDUE|SEQUENTIAL) Specifies whether zones are based on residue numbers in the PDB file or on sequential numbering (running through all chains).

OCCRANK *n* Sets ProFit to read the nth ranked highest occupancy atom position for alternative atom positions.

For structure files containing partial occupancies, lower occupancy atoms can be read using by setting the occupancy rank parameter to read alternative atom positions.

By default, OCCRANK is set to 1 and reads the highest ranked atom position, a setting of 2 will read the second most occupied position and a setting of 3 will read the third most occupied position, etc.

ORDERFIT Performs a fit of all mobile structures to the reference structure. The most similar structures are fitted first.

PAIRDIST [***filename***] Prints the pairwise distances between equivalent atom pairs. If the first character of the (optional) filename is a pipe character (|), then the results will be piped into the specified program. For example:

```
PAIRDIST |less
```

will cause the results to be displayed on the screen using the `less` pager program.

PRINTALIGN [**FASTA**|**PIR**] [***filename***] Prints current fitting zones as a sequence alignment. The default output is a (user-friendly) pairwise alignment with the reference and mobile sequences printed as pairs of 60-character wide lines. The optional `FASTA` and `PIR` parameters set the printout to FASTA or PIR formatting. **ProFit** can read PIR-formatted files using the `READALIGN` command.

QUIET [**OFF**] Switches on (or off) quiet mode. In quiet mode, warning messages are suppressed and progress of iterative zone updating and multiple structure fitting is not reported.

QUIT Exits from the program.

RATOMS ***atm[,atm]...*** Specifies atoms over which to calculate the RMS deviation. Fitting must already have been performed.

READALIGNMENT ***filename*** Reads an alignment in PIR sequence file format and sets zones based on that alignment. Note that when used with multiple structures, the first sequence *must appear twice* in the alignment file. This is because it is used as both the first reference and mobile set.

REFERENCE [**XMAS**] ***filename*** Reads a reference PDB structure. If compiled with XMAS support, then the XMAS keyword specifies that the input is in XMAS format.

RESIDUE [***filename***]

Gives a by-residue RMS deviation. If the first character of the (optional) filename is a pipe character (|), then the results will be piped into the specified program. For example:

```
RESIDUE |less
```

will cause the results to be displayed on the screen using the `less` pager program.

RMS Recalculate the RMS deviation over the zones and atoms currently defined with `RZONE` and `RATOMS`.

RZONE ***zonespec*** Adds a zone specification to the list of zones considered in RMS deviation calculation. `RZONE *` or `RZONE CLEAR` resets the zones for RMSD calculation to be the same as that specified with the `ZONE` command.

SCRIPT ***filename*** Executes a script file.

SETCENTER ***residue*** See `SETCENTRE`.

SETCENTRE ***residue*** Specifies a single residue as the centre of fitting. Entering SETCENTRE CLEAR or SETCENTRE * will clear the centre residue.

SETREF [***n***] Sets the reference structure to the nth mobile structure when fitting multiple structures.

If no structure number is given then the reference is automatically set by performing an all versus all comparison of the mobile structures then selecting the structure with the lowest overall RMSD to the other mobile structures.

STATUS[***filename***]

Reports current program status. If the optional filename parameter is given, output is directed to the specified file. If the file cannot be opened or a filename is not specified, output appears on the screen. If the filename begins with a pipe character (|), the results are piped into the specified program.

SYMMATOMS [[**OFF**|**ON**|**ALL**]|***xxx*** [**OFF**|**ON**] where *xxx* is a three-letter amino acid code.

Enabes the auto-matching of symmetrical atoms (eg CD1 - CD2 and CE1 - CE2 of tyrosine) in **ProFit**

`SYMMATOMS` matches charged oxygens and nitrogens on arginine, aspartate and glutamate residues and the delta and epsilon carbons of phenylalanine and tyrosine residues.

It is also possible to match the nitrogen and oxygen atoms of the amide sidechains of asparagine and glutamine residues and the prochiral methyl groups of valine and leucine.

Typing `SYMMATOMS` will display the pairs of atoms currently matched by **ProFit**. Typing `SYMMATOMS ON` or `SYMMATOMS OFF` will turn symmetrical atom matching on or off.

Individual residue types, for example ASP, can be turned-on or off by typing `SYMMATOMS ASP ON` or `SYMATM ASP OFF`, respectively. Alternatively, `SYMMATOMS ALL ON` will turn all atom pairs on.

By default, the the matching of symmetrical atoms is turned-off

TRIMZONES This command is used primarily with fitting zones derived using `ALIGN`. With pairwise alignments, the lengths of the aligned regions may vary and there may be gaps in the alignments from one structure to another. The `TRIMZONES` command trims the ends of the aligned zones and adds gaps allowing for a like versus like comparison by using fitting zones that are common to all the structures.

`TRIMZONES` is automatically called by the `ALLVSALL` and `SETREF` commands. This command is only used with multiple structures.

WEIGHT Weight the fitting by the mean of the B-values in the equivalent atoms.

WRITE [**REFerence**] ***filename*** Writes the fitted structure to a PDB file. If the first character of the filename is a pipe character (|), then the results will be piped into the specified program. For example:

```
WRITE |less
```

will cause the coordinates to be displayed on the screen using the `less` pager program.

If the `REFERENCE` keyword is given (only the letters 'REF' are required), then the reference set will be written. This is used in conjunction with the `CENTRE` command.

WTAVERAGE [ON|OFF] Sets the weighting system for the averaged reference structure to the default weighting system where the change in the coordinates of the reference structure is inversely proportional to the number of mobile strucures. The weighted averaging scheme was introduced to lower the effect that outlying structures have on the averaged reference. (Default: ON)

The alternative weighting scheme sets the coordinates of the reference stucture to the average of the reference and the mobile structures. This was the scheme used by ProFit prior to version 3.0. (`WTAVERAGE OFF`)

ZONE *zonespec* Adds a zone specification to the list of zones considered in fitting. `ZONE *` or `ZONE CLEAR` removes all zone specifications.

17 Installing the GNU Readline library

If you wish to compile with Readline support and your unix-like system does not have the Readline library installed, follow the tips here to help with Readline installation. Most Linux installations will have the readline library installed already and all you need to do is uncomment the two lines in the Makefile:

```
READLINE    = -DREADLINE_SUPPORT
READLINELIB = -lreadline -lcurses
```

On recent versions of Linux, if this doesn't work, then you may have to install the readline development libraries with a command like:

```
yum install readline-devel         (RPM-based systems)
apt-get install libreadline5-dev   (Debian-based systems)
```

If this doesn't work, or you are using another Unix system then proceed as follows:

Download the latest version of GNU readline from `http://cnswww.cns.cwru.edu/php/chet/readline/rltop.html` At the time of writing, this is `readline-6.0.tar.gz`

Unpack the gzipped tar file under `/tmp`

```
cd /tmp
tar -zxvf readline-6.0.tar.gz
```

Change to the directory this creates and run configure:

```
cd readline-6.0
./configure
```

If you do not have write access to the /usr/local/ hierarchy, then you can install the files somewhere else:

```
./configure --prefix=/home/my-user-name/packages
```

Now build the readline library

```
make
```

and install (become superuser first if installing under /usr/local)

```
make install
```

Now, ensure that the directory where the library has been installed (`/usr/local/lib/` by default) is in the search path. You can do this by setting the environment variable `LD_LIBRARY_PATH`

```
(csh)   setenv LD_LIBRARY_PATH /usr/local/lib
(sh)    export LD_LIBRARY_PATH=/usr/local/lib
```

Alternatively, if you have root access, you can edit the file `/etc/ld.so.conf` to add the directory in which the library has been installed. Under recent Linux installations, there is another alternative which is to create a file `/etc/ld.so.conf.d/readline.conf` just containing a single line with the directory where the library has been installed. In either case, you must now (as root) type the command:

```
/sbin/ldconfig
```

Now, modify the Makefile, such that this directory is in the linker's library path. Change:

```
READLINELIB = -lreadline -lcurses
```

to:

```
READLINELIB = -L/usr/local/lib -lreadline -lcurses
```

Now build with `make` as usual, but endure that `LD_LIBRARY_PATH` is set whenever you want to run the program. Alternatively, install with

```
./configure --prefix=/usr/lib
```

to install in the main system directories and then it will be in the default search path. This isn't recommended unless you know what you are doing!

You can also link the readline library statically to ensure portability to machines with Linux machines having different versions of the readline library installed. In this case you will not need the `LD_LIBRARY_PATH` or changes to `/etc/ld.so.conf`. To do this, edit the Makefile and change:

```
READLINELIB = -lreadline -lcurses
```

to

```
READLINELIB = /usr/lib/libreadline.a -lcurses
```

(changing `/usr/lib/` as required to point to wherever libreadline has been installed).

18 Copyright

Please note that the program is called **ProFit** — not PROFIT, Profit or profit; this attempts to avoid confusion with the threading program known as PROFIT. **ProFit** was written first and released to the public around the same time.

ProFit is pronounced as it is written, i.e. 'pro' (as in 'protein') then 'fit' (not 'profit' as in 'make lots of money'!

ProFit was initially written by Dr. Andrew C.R. Martin while self-employed and trading as SciTech Software. Enhancements have been made since at UCL and at the University of Reading. Addition of iteration and multiple fitting was sponsored by Inpharmatica, Ltd. Enhancements in V2.6 and V3.0 were written by Dr. Craig Porter and made possible by a Tools and Resources grant from the BBSRC.

This program is not in the public domain.

It may not be copied or made available to third parties, but may be freely used by non-profit-making organisations and commercial companies who have obtained it directly from the author or by FTP or HTTP from the author's web sites.

If you did not register the program via the web site, you are requested to send EMail to the author to say that you are using this code so that you may be informed of future updates.

The code may not be made available on other FTP or Web sites without express permission from the author.

The code may be modified as required, but any modifications must be documented so that the person responsible can be identified. If someone else breaks this code, the author doesn't want to be blamed for code that does not work! You may not distribute any modifications, but are encouraged to send them to the author so that they may be incorporated into future versions of the code.

While the compiled **ProFit** program may be used by commercial companies, it may not be sold commercially or included as part of a commercial product. The source code or any derivative works may not be sold commercially or used for commercial purposes outside of **ProFit** without prior permission from the author.

While this software is provided "as is" and free of charge, I do appreciate hearing from people who use it and find it useful. An EMail or a postcard would be nice.

If you find **ProFit** useful, please tell your colleagues about it. Please *do not* pass copies of **ProFit** on to them directly; ask them to obtain it *via* my World Wide Web page (`http://www.bioinf.org.uk/software/profit/`)

19 How do I Reference ProFit?

No paper has been published describing **ProFit** itself since it is simply a convenient program (I hope) to let you use a standard fitting algorithm; consequently, it is a little difficult to reference. The exact wording is up to you and dependent on the context, but I suggest something similar to:

> Fitting was performed using the McLachlan algorithm (McLachlan, A.D., 1982 "Rapid Comparison of Protein Structres", Acta Cryst A38, 871-873) as implemented in the program ProFit (Martin, A.C.R. and Porter, C.T., `http://www.bioinf.org.uk/software/profit/`)

20 Acknowledgements

Inpharmatica Ltd. are acknowledged for funding development of V2.0 of ProFit. The BBSRC are acknowledged for funding development of V2.6 and V3.0 of ProFit.