PDB_EXTRACT

Extracting Information from Output of X-ray Crystallographic Applications

RCSB Protein Data Bank

(June, 18, 2004; last modified Jan. 13, 2005)

Contents

1 What does PDB_EXTRACT do?

PDB_EXTRACT is used to extract information about data processing, heavy atom phasing, molecular replacement, density modification, and final structure refinement from the output files produced by many X-ray crystallographic applications. PDB_EXTRACT merges the information into two mmCIF (macromolecular Crystallographic Information File) files, one with structure factors and one with coordinate and statistic. These two files are ready for PDB deposition.

PDB_EXTRACT can work with/without ADIT (http://deposit.pdb.org/adit) to deposit complete data. When working with ADIT (recommended), users just upload the extracted mmCIF files to ADIT. Enter any additional information into ADIT and submit your file directly from there. When working without ADIT, users just fill any additional information to a plain text file (called *data_template.text*) and run the program. Users can email the extracted two files to the Protein Data Bank (PDB) 'deposit@rcsb.rutgers.edu'.

The advantage of using pdb_extract:

a. Faster to prepare your deposit mmCIF file. User just provide the LOG files from the software used and filled some information to the *data_template.text* file from your local computer. Some items are pre-calculated for you. For example, Matthews coefficient and solvent constant, molecular entities.

b. Complete and accurate to deposit your file. pdb_extract automatically extract information from your LOG file. This reduces many typing errors.

c. Great for multiple structural deposition. Users just keep one *data_templete.text* file and copy and paste the unchanged information to the newly generated one for individual structural deposition.

Collectively, these software tools reduce the human effort required to assemble very complete and validated protein structure entries ready for PDB deposition.

IMPORTANT NOTES:

1. If you have several structures ready to be deposited to the PDB site, you need to apply the pdb_extract program to each individual structure, since each structure requires a single PDB ID for deposition.

2. You may have a lot of trials for each step (data processing, heavy atom phasing, or density modification, or final structure refinement), but information extracted from each step should be only from the best trial that leads to next step toward solving your structure.

3. You may use different programs for heavy atom phasing solution. For example, you used program A to locate heavy atom positions and you used program B to refine heavy atom parameters (like x, y, z, occupancy and B factors etc.). Phasing statistics information will be extracted from the output of program B; therefore, pdb_extract should be applied to the output of program B. However, if you want to give credit to program B, you can type '-p program-name' without giving LOG files. For example, ARP/Warp is a intermediate step, you may just type '-p Warp'.

4. You may also use different programs for final structure refinement, but pdb_extract should be only applied to the program which leads to your final structure deposition.

2 Program access

The source and binary versions of PDB_EXTRACT can be downloaded from the address http://deposit.pdb.org/software. The source is available under an Open Source license. The binary distributions are available for Intel-Linux, SGI-IRIX, DEC-Alpha, and Sun-Solaris.

The web interface can be accessed at http://pdb-extract.rutgers.edu for PDB_EXTRACT and http://deposit.pdb.org/adit/ for ADIT.

PDB_EXTRACT has been integrated into CCP4 (http://www.ccp4.ac.uk/main.html) and the CCP4i interface(Version 5.0 and above). Users can run PDB_EXTRACT under the CCP4 environment.

3 Installations

NOTE: if you have installed CCP4 (version 5.0 and above), you do not have to install pdb_extract. You just use the CCP4 environment to run the pdb_extract program.

Requirements:

1. plantforms Intel-Linux, SGI-IRIX, DEC-Alpha, and Sun-Solaris:
2. disk space great than 140 Mbytes for source distribution and 30 Mbytes for binary distribution
3. C/C++ compilers

3.1 Installation of binary distribution

It is recommended to install the binary distribution, since it is very fast to install and it takes very small space. The binary distributions are available for Intel-Linux, SGI-IRIX, DEC-Alpha, and Sun-Solaris.

Step 1 Uncompress and unbundle the distribution using the following command:

zcat pdb-extract-vX.XXX-XXX.tar.gz | tar -xf -

The result of this command is a subdirectory pdb-extract-vX.XXX-XXX in the current directory, which contains the following:

bin - subdirectory that contains application executables
pdb-extract-vX.X - subdirectory that contains an example in its "examples" subdirectory.
data - subdirectory that contains some data files needed by the application.
etc - subdirectory that contains utility scripts and application software license agreement.

Step 2 Set up the environment variables.

A. Define RCSBROOT environment variable to point to the installation directory.

Assuming that the installation directory is
/home/username/pdb-extract-vX.XXX-XXX, execute in the shell:

For C shell users:
setenv RCSBROOT /home/username/pdb-extract-vX.XXX-XXX
For Bourne shell users:
RCSBROOT=/home/username/pdb-extract-vX.XXX-XXX; export RCSBROOT

B. Add "bin" subdirectory to the PATH environment variable.

Assuming the installation directory is /home/username/pdb-extract-vX.XXX-XXX,

For C shell users:
setenv PATH "/home/username/pdb-extract-vX.XXX-XXX/bin:"$PATH
For Bourne shell users:
PATH="/home/username/pdb-extract-vX.XXX-XXX/bin:"$PATH; export PATH

Step 3 Make binary data from ASCII data

Position in the pdb-extract-vX.XXX-XXX/etc directory and run the script binary.sh:

```
cd pdb-extract-vX.XXX-XXX/etc
./binary.sh
```

This command will create certain binary data files, using the ASCII data files in data/ascii directory. The resulting files are stored in data/binary directory. Note that it may take several minutes for this step to complete. This step must be executed before the tool can be utilized.

3.1 Installation of source code distribution

Step 1. Uncompress and unbundle the distribution using the following command:

zcat pdb-extract-vX.XXX-XXX.tar.gz | tar -xf -

The result of this command is a subdirectory pdb-extract-vX.XXX-XXX in the current directory. It contains subdirectories of various source modules and the following items important for the user:

bin - subdirectory in which the application executables will be placed after the build process.
pdb-extract-vX.X - subdirectory that contains several examples in its "examples" subdirectory.
data - subdirectory that contains some data files needed by the application.

Step 2. Set up the environment variables.

A. Define RCSBROOT environment variable to point to the installation directory.

Assuming that the installation directory is
/home/username/pdb-extract-vX.XXX-XXX, execute in the shell:

For C shell users:

```
setenv RCSBROOT /home/username/pdb-extract-vX.XXX-XXX
```

For Bourne shell users:

```
RCSBROOT=/home/username/pdb-extract-vX.XXX-XXX; export RCSBROOT
```

B. Add "bin" subdirectory to the PATH environment variable.
Assuming that the installation directory is /home/username/pdb-extract-vX.XXX-XXX,

For C shell users:

```
setenv PATH "/home/username/pdb-extract-vX.XXX-XXX/bin:"$PATH
```

For Bourne shell users:

```
PATH="/home/username/pdb-extract-vX.XXX-XXX/bin:"$PATH; export PATH
```

Step 3. Building the Application (compile the program)

Position in the pdb-extract-vX.XXX-XXX directory and run "make" command:

```
cd pdb-extract-vX.XXX-XXX
make
```

The application executables will be placed in the "bin" subdirectory.

NOTE: The users who are working on Sun platform are advised to check the compiler flags in etc/make.platform.sunos5 file. Depending on the compiler version, users may be required to make modifications to those compiler flags.

Step 4. Make binary data from ASCII data

Position in the pdb-extract-vX.XXX-XXX directory and run "make" command as follows:

make binary

NOTE: This command will create certain binary data files, using the ASCII data files in data/ascii directory. The resulting files are stored in data/binary directory. Note that it may take several minutes for this step to complete. This step must be executed before the tool can be utilized.

4 Run the program

There is an example included in this distribution. User can find more examples from the location http://pdb.rutgers.edu/mmcif/PDB_EXTRACT/index.html.

4.1 quick run the program

This example is located in the subdirectory of "pdb-extract-vX.X/examples/Example_1".

The directory contains the following:

input_data - contains the input data for the example
deposit - contains the resulting files (after running the program):
validation_result - contains all validation results (after running the program)

To execute the example, position in the appropriate directory and invoke test.sh and test_script.sh scripts. For example:

```
cd pdb-extract-vX.XXX-XXX/pdb-extract-vX.X/examples/Example_1
./test.sh
./test_script.sh
```

4.2 Explanations of arguments and input/output files

The output files:

After you run the above commands (for example ./test.sh), you will get the following files in the directory pdb-extract-vX.X/examples/Example_1/deposit/

Example_1.cif - this is the merged mmCIF file created by "pdb_extract"
Example_1_deposit.cif - this is the new mmCIF file by "maxit".
Example_1.sf.cif - this is the structure factor created by "pdb_extract_sf"

You can deposit the two files Example_1.sf.cif and either Example_1.cif or Example_1_deposit.cif

The files generated by the validation program are in the directory.
pdb-extract-vX.X/examples/Example_1/validation_result

Example_1.ps – this is the postscript file for a picture of molecules in asymmetric unit and crystal packing in a unit cell.
Example_1.-asym.wrl – this is a VRML file for a picture of molecules in asymmetric unit (viewed dynamically by internet browsers).
Example_1.-packing.wrl – this is a VRML file for a picture of molecular crystal packing (viewed dynamically by internet browsers).
Example_1.html – this is a HTML file for essential information by the structure.
Example_1.letter – An **important** file. It contains all the diagnostic errors of the structure.

User should read this file (**Example_1.letter**) with care and correct any geometrical errors before submission of the structure.

The input files:

MAD experiment

- Phasing calculation by program CNS (version 1.1).
- Density modification by program CNS (version 1.1).
- Final structure refinement by program CNS (version 1.1).

Data files:

- pdb-extract-vX.X /examples/Example_1/input_data/mad_sdb.dat
    - File format: CNS log format.
    - File source: run CNS (mad_phase.inp)
    - Data to be extracted: heavy atom coordinates, B factors, etc.

- pdb-extract-vX.X /examples/Example_1/input_data/mad_summary.dat
    - File format: CNS log format.
    - File source: run CNS (mad_phase.inp)
    - Data to be extracted: all the phasing statistics

- pdb-extract-vX.X /examples/Example_1/input_data/mad_fp.dat
    - File format: CNS log format.
    - File source: run CNS (mad_phase.inp)
    - Data to be extracted: wavelengths, f_prime, f_double_prime.

- pdb-extract-vX.X /examples/Example_1/input_data/density_modify.dat
    - File format: CNS log format.
    - File source: run CNS (fourier_map_dm.inp)
    - Data to be extracted: FOM after density modification, dm method

- pdb-extract-vX.X /examples/Example_1/input_data/deposit_cns.mmcif
    - File format: mmCIF
    - File source: run CNS (deposit_mmcif.inp)
    - Data to be extracted: the atom coordinates and B factors and structure refinement statistics.

- pdb-extract-vX.X /examples/Example_1/input_data/data_template.text
    - File format: mmCIF
    - File source: Generated by ' extract -pdb pdb_file_name'.
    - Data to be extracted: a complete chemical sequence.

The script to run the programs (test.sh)

The script for “test.sh” is a combination of various Unix command line programs.

The script is in the directory “pdb-extract-vX.X /examples/Example_1/” and the content of the script is the following:

```
#!/bin/sh

# use pdb_extract to extract the required statistics and get a mmcif file.
pdb_extract  -e  MAD -r CNS -iCIF input_data/deposit_cns.mmcif  \
-iENT input_data/data_template.text \
-p  CNS -iLOG  input_data/mad_sdb.dat input_data/mad_summary.dat input_data/mad_fp.dat  \
-d CNS -iLOG input_data/density_modify.dat  -o Example_1.cif

# use pdb_extract_sf to convert the structure factor to mmCIF format.
pdb_extract_sf  -dt I -dp HKL -c 1 -w 1 -idat input_data/w1.sca  \
-c 1 -w 2 -idat input_data/w2.sca
-c 1 -w 3 -idat input_data/w3.sca -o Example_1.sf.cif

# use validation-v8 to validate the mmcif file.
validation-v8 -f Example_1.cif -o 2 -public -exchange -adit

# use maxit-v8.01-O to reorder the mmcif format
maxit-v8.01-O -i Example_1.cif -o 8 -exchange_in -exchange_out

# move the files to some directory and delete some log files.
mv Example_1.cif.cif Example_1_deposit.cif
mv Example_1.cif deposit
mv Example_1_deposit.cif deposit
mv Example_1.sf.cif deposit
mv Example_1* validation_result
rm -f *log *err procheck* SEQUENCE.DAT *ERR validation.alignment
```

The script to run the programs (test_script.sh)

The script for “test_script.sh” is an alternative way to obtain the same result as above. It is also a combination of various programs. The difference is that it used “extract” instead of the “pdb_extract” and “pdb_extract_sf”. All the information is included in the file “log_script.inp”.

The script is in the directory “pdb-extract-vX.X /examples/Example_1/” and the content of the script is the following:

```
#!/bin/sh

# use extract to run everything in example_1.inp and get a mmcif file.
extract -ext input_data/log_script.inp

# use validation-v8 to validate the mmcif file.
validation-v8 -f script_example_1.cif -o 2 -public -exchange

# use maxit-v8.01-O to reorder the mmcif format
```

```
maxit-v8.01-O -i script_example_1.cif -o 8 -exchange_in -exchange_out

# move the files to some directory and delete some log files.
mv script_example_1.cif.cif script_example_1_deposit.cif
mv script_example_1.cif deposit/
mv script_example_1_deposit.cif deposit/
mv script_example_1_sf.cif deposit/
mv *.html  *.ps  *.letter  *.wrl  validation_result/
rm -f *log *err procheck* SEQUENCE.DAT *ERR validation.alignment
```

5 Tutorials

There are four ways to extract crystallographic information and deposit complete data to the Protein Data Bank.

1. Use CCP4i (http://www.ccp4.ac.uk/main.html/)
2. Use the Web interface (http://pdb-extract.rutgers.edu/).
3. Use Unix Command Line Interface.
4. Use Script Interface.

Three interfaces have different features. For example, The CCP4i or Web interface provide a simple graphic interface. Users just select the program name and output file names to do the job. The full Unix command line method provides the greatest flexibility. User need to read the command options to run the program. The script input method provides a simple local interface.

Here, we give a concrete example to show how to use **pdb_extract** for complete data extraction.

In this example, the experimental method for solving the protein structure was multiple anomalous diffraction (MAD). The information for the experiment is as the following:

- One crystal was used for data collection
- Three wavelengths (e.g. inflection, peak, remote edge) were tuned for diffraction.
- The program HKL2000 was used for indexing and data scaling.
- The program SOLVE was used for heavy atom phase determination and phase refinement.
- RESOLVE was used for density modification.
- All three reflection data files were used for phasing.
- REFMAC5 was used for final structure refinement.

The LOG files generated from above programs are the following:

- HKL2000 program generated three reflection data files (scale1.sca, scale2.sca, scale3.sca) and three log files (scale3.log, scale2.log, scale2.log) from scaling corresponding to the three data sets.
- SOLVE program generated one log file (solve.prt) containing phasing statistics and one PDB file (ha.pdb) containing heavy atom (Se in this case) coordinates.
- RESOLVE program generated one log file (resolve.log) containing statistics
- REFMAC5 program generated one PDB file (refmac.pdb) containing atomic coordinates and one mmCIF file (native.refmac) containing refinement statistics.

5.1 Using CCP4i interface

Step 1. From the main window of CCP4i, select the '*Data Harvesting Management Tool*' option.

Step 2. From the option of '*Run program to*', select the '*Extract additional information for deposition*'

Step 3. Select the '*Generate a data template file*' from various steps

- Type (or select using browse) in the yellow boxes either the PDB or mmCIF file name obtained from the final structure refinement and the output file name. In this case, the output coordinate file is refmac.pdb.
- Run the **pdb_extract** program to obtain the data template file. Edit this file according to the instruction in the text file.

Step 4. Select the '*Generate a complete mmCIF file for PDB deposition*' from various steps

- Select program names and log file names generated from the selected programs.
    - Select the scaling program HKL and select the log file scale1.log to extract scaling statistics (data used for refinement).
    - Select phasing method MAD and program SOLVE. Give the log file solve.prt to obtain phasing statistics.
    - Select the density modification program RESOLVE and the log file resolve.log to obtain density modification statistics.
    - Select the structure refinement program REFMAC5 and the PDB coordinate file refmac.pdb and the data harvest file native.refmac to obtain the PDB coordinates and refinement statistics
    - Select the data template file generated from step 3 to obtain the chemical sequence and the non-electronically extracted information.
- Run the **pdb_extract** program to obtain a complete data in mmCIF format. The final output file can be uploaded to **ADIT** (http://deposit.pdb.org/adit/) for on line structure validation and submission.

NOTE: The characters of file name should always start from beginning of each yellow box. There should be no white space in each box, even no file name is typed in.

5.2 Using the Web interface (http://pdb-extract.rutgers.edu/)

Follow on line tutorial!

5.3 Using Unix Command Line Interface

STEP 1. Obtain the *template data file* 'data_template.text' using the command

extract -pdb coordinate_PDB_file_name

or

extract -cif coordinate_CIF_file_name

After running the program, you will get a file called 'data_template.text'. There are 19 CATEGORIES for data entries in the file 'data_template.text'. CATEGORY 1-2 contains the extracted unit cell parameters and the unique molecular chemical sequence group. Please modify the two CATEGORIES as necessary. Additional structure information can be filled into CATEGORIES (3-19).

The content of the *data template file* 'data_template.text' is given in **Appendix**

STEP 2. Obtain coordinates and all the statistics

Run the **pdb_extract** program:

```
pdb_extract -e MAD \
        -p SOLVE -iLOG                    solve.prt \
        -d RESOLVE  -iLOG resolve.log \
        -r refmac5 -icif peak.refmac -ipdb refmac.pdb\
        -s HKL –iLOG scale-refine.log \
       -sp HKL scale1.log scale2.log scale3.log \
      -iENT date_template.text \
        -o output.cif
```

STEP 3. Obtain structure factors

Run **pdb_extract_sf** to convert HKL format to mmCIF format and merge all the files to one file.

```
pdb_extract_sf -rt F -rp refmac5 -idat refmac_sf.mmcif \    (for refinement)
          -dt I -dp HKL \                  (for phasing)
          -c 1 -w 1 -idat scale1.sca \
          -c 1 -w 2 -idat scale2.sca \
          -c 1 -w 3 -idat scale3.sca \
          -o output_sf.cif
```

Since REFMAC5 use MTZ file for refinement, the reflection data file must be converted to CIF format using ccp4i or mtz2various. The file name is refmac_sf.mmcif.

The output file (output_sf.cif) contains one reflection data block for refinement and one data block for protein phasing.

STEP 4 Validation and deposition

Note: the PDB-Validation program is not provided in CCP4 package.

Either upload the two files (output.cif, output_sf.cif) to ADIT, or use the following commands:

validation-v8 -f output.cif -o 2 –public -exchange
maxit-v8.01-O -i output.cif -o 8 -exchange_in -exchange_out -keep_contact_author

Then upload the validated file and the structure factor file to ADIT for on-line submission. The alternative is to email the file to 'deposit@rcsb.rutgers.edu'

5.4 Using the script interface

This script is similar to the CNS script input. Users just enter all the LOG file name and the program names to the plain text file ‘log_script.inp’ and run the program.

STEP 1 obtain the plain text file ‘log_script.inp’

extract -pdb refmac.pdb

You will get one script file called 'log_script.inp' and one data template file ‘data_template.text’. Edit the data template file according to the instruction in the file. Fill all the Log file names and the program names as well as the data_template file to the script file 'log_script.inp'.

The content of the file ‘log_script.inp’ is shown in the **Appendix**.

STEP 2 run the program:

extract -ext log_script.inp

You will get the same results as using the Unix command line option.

STEP 3 Validation and deposition: (as in the Unix command line option).

6 Some helpful hints to get LOG (or output) files from various crystallographic applications

Listed below are the programs used from data collection to structure determination.

6.1 Data collection/reduction/scaling

This is the early stage of solving a crystal structure. The statistics for integrating, scaling the data tells the quality of the structure factors. Information can be extracted from this step is the following:

- Intensities (or amplitude) and standard deviations
- Data completeness (overall, resolution shells)
- Redundancy (overall, resolution shells), mosaicity.
- R-merge, R-sym (overall, resolution shells)
- <I>/<sigma> (overall, resolution shells)
- Total and unique reflections collected.
- Resolution range

6.1.1 Using HKL/HKL2000/scalepack (http://www.lnls.br/infra/linhasluz/denzo-hkl.htm)

HKL is a package by Otwinowski for data collection/reduction/scaling. You can use graphic interface or the scalepack script to scale your data. The LOG file (e.g. scale1.log) containing statistics is from the step of scaling data.

To extract statistics:
pdb_extract –s HKL **–ilog** scale1.log (one data for refinement)
pdb_extract –sp HKL **–ilog** scale1.log scale2.log … (multiple data for phasing)

6.1.2 Using D*trek (http://www.msc.com/protein/dtrek.html)

D*trek is a package by Rigaku/MSC for data collection/reduction/scaling. You can use graphic interface to scale (or merge/average) your data. The LOG file (e.g. scale1.log) containing statistics is from the step of scaling data.

To extract statistics:
pdb_extract –s Dtrek **–ilog** scale1.log (one data for refinement)
pdb_extract –sp Dtrek **–ilog** scale1.log scale2.log … (multiple data for phasing)

6.1.3 Using SAINT (http://xray.chm.bris.ac.uk/facilities/smart.html)

SAINT is a package by Bruker (Siemens Molecular Analytical Research Tool) for data collection/reduction/scaling. The LOG file (e.g. scale1.ls) containing statistics is from the step of scaling data.

To extract statistics:
pdb_extract –s SAINT **–ilog** scale1.ls (one data for refinement)
pdb_extract –sp SAINT **–ilog** scale1.ls scale2.ls … (multiple data for phasing)

6.1.4 Using 3DSCALE (?)

3DSCALE is a program by Fu et al. for data scaling. The LOG file (e.g. scale1.log) contains statistics.

To extract statistics:
pdb_extract –s 3DSCALE **–ilog** scale1.log (one data for refinement)
pdb_extract –sp 3DSCALE **–ilog** scale1.log scale2.log … (multiple data for phasing)

6.1.5 Using SCALA (http://www.ccp4.ac.uk/dist/html/scala.html)

SCALA is the CCP4 supported program. It scales together multiple observations of reflections. SCALA can generate mmCIF file containing useful statistics. When you run the programs, you must ask the program to export the data harvest file. For example the file will be name.scala. If you do not use the mmCIF file, you can use the LOG file generated by SCALA.

To extract statistics:
pdb_extract –s scala **–icif** name.scala (one data for refinement)
or
pdb_extract –s scala **–ilog** scala.log (one data for refinement)

6.2 Programs for molecular replacement

If your structure was solved by molecular replacement, you can use the **pdb_extract** program to extract information from the LOG files. The information may be the followings:

- Low and high resolution used in rotation and translation.
- Rotation and translation methods
- Reflection cut off criteria, reflection completeness.
- Correlation coefficients for I or F between observed and calculated.
- R_factor, packing information, and model details.

6.2.1 Using CNS/XPLOR (http://cns.csb.yale.edu/v1.1/)

CNS can be used to do molecular replacement. After you finish the translation search, you can get a log file called translation.list which contains all the information of molecular replacement.

The command to extract information is:
pdb_extract -o test.mmcif **–e** MR **–m** CNS **–ilog** translation.list

6.2.2 Using Amore (CCP4 version 4.1- above): (http://www.ccp4.ac.uk/dist/html/INDEX.html)

Amore is a popular program for molecular replacement. It is distributed in the CCP4 package. After rotation and translation search, you will generate two log files rotation.log and translation.log. You may extract information from both log files

The command to extract information is:
pdb_extract –e MR **–m** amore **–ilog** rotation.log translation.list **-o** test.mmcif

6.2.3 Using Morep (CCP4 version 4.1-above): (http://www.ccp4.ac.uk/dist/html/INDEX.html)

Morep is a program for molecular replacement. It is distributed in the CCP4 package. When you run the script, you can give a log file name like morep.log. All the statistic information will be recorded in the log file.

The command to extract information is:
pdb_extract –e MR **–m** morep **–ilog** morep.log **-o** test.mmcif

6.2.4 Using EPMR: (http://www.msg.ucsf.edu/local/programs/epmr/epmr.html)

EPMR is a Unix command line program for molecular replacement.
When you run the program, please give a log file name like the following
Epmr [options] files > epmr.log
All the statistic information will be recorded in the log file.

The command to extract information is:
pdb_extract –e MR **–m** epmr **–ilog** epmr.log **-o** test.mmcif

6.3 Programs for heavy atom position location and protein phasing

The whole issue of protein crystallography may be the phase problem. Heavy atom phasing is performed at earlier stage of structure determination. Some log files generated from phasing step contain important statistic information (don't throw them away). The pdb_extract program can be used to extract the following information from log files.

- Wavelength, f',f" , resolution range
- FOM (acentric, centric, overall, resolution shells)
- R-Cullis (acentric, centric, overall, resolution shells)
- R-Kraut (acentric, centric, overall, resolution shells)
- Phasing power (acentric, centric, overall, resolution shells)
- Number of heavy atom sites, heavy atom type.
- Heavy atom location method.
- Heavy atom B-factor, occupancies, and xyz coordinates.

6.3.1 Using CNS/XPLOR (http://cns.csb.yale.edu/v1.1/)

CNS is a complete software system for protein crystallography. The scripts for heavy atoms location and phasing refinement are 'mad_phase.inp' or 'ir_phase.inp'. When you run these scripts, you will get the output files like 'phase_final.summary', 'phase_final.sdb' or 'mad_phase.fp'.

The output file phase_final.summary has all the phasing statistics. The output file phase_final.sdb has all the heavy atom coordinates, occupancies and B factors. The output file mad_phase.fp has refined f_prime and f_double_prime.

(Note: The refined heavy atom coordinates and the B factors and the occupancies can be found in a file like 'phase_final.sdb'. If you prefer to convert to the PDB format, you can run the script sdb_to_pdb.inp. You will get a file 'phase_final.pdb' with PDB format.)

To extract phasing information, you need the following description:
pdb_extract -o test.mmcif **–e** MAD **–p** CNS \
–iLOG phase_final.summary phase_final.sdb mad_phase.fp

or, if you have the heavy atom coordinates in PDB format:
pdb_extract -o test.mmcif **–e** MAD **–p** CNS \
–iLOG phase_final.summary mad_phase.fp \
–iPDB phase_final.pdb

6.3.2 Using MLPHARE (http://www.ccp4.ac.uk/dist/html/INDEX.html)

MLPHARE is a program in the CCP4 suit. It is used for refining heavy atom parameters:

If you use the CCP4i graphical interface or the script mode, you need to ask the program to write a harvesting file. Select the **data harvest** button, when you use the CCP4i interface. Do not give command word NOHARV, when you use script. After you finished running this program, you will get a file (e.g. name.mlphare) which is in mmCIF format. It contains statistics information for heavy atom phasing refinement.

For extracting the wavelength information, you need to run program REVISE in the CCP4 (version 4.1-4.2). You may get a file (like prephadata.log which is in LOG format)

To extract phasing information, you need the following description:
pdb_extract -o test.mmcif **–e** method **–p** MLPHARE \
–iCIF name.mlphare **–iLOG** prephadata.log

6.3.3 Using SOLVE (http://www.solve.lanl.gov/)

Solve is a popular program for finding heavy atom and refining heavy atom parameters. The summary information will be written to a file called "solve.prt" (default name used by the program). The program also exports the heavy atom coordinates, called "ha.pdb".

The pdb_extract program works for any one of the following situations:

- one data set for SAD
- one data set for MAD
- one data set for MIR
- one data set for MIR plus anomalous scatters in the native. (e.g. two derivatives Hg, plus native data has Fe with anomalous)
- one MAD data set with two anomalous scatters. (e.g. both Se and Fe have anomalous signals)
- two data sets (MAD + MIR)
- two data sets for MAD
- two data sets for MIR

To extract phasing information, you need the following descriptions:
pdb_extract –e method **–p** SOLVE **–iLOG** solve.prt **-ipdb** ha.pdb **-o** test.mmcif

6.3.4 Using SHARP (http://babinet.globalphasing.com/sharp/)

SHARP is a program for finding heavy atom positions and refining heavy atom parameters. When you run SHARP or autoSHARP, the log files containing useful information are normally in the directory sharpfiles/logfiles_local/*dirs*, where *dirs* are all the subdirectories for your various structures. One should be advised that the location of generated log files might depend on how the program is installed!

SHARP produces many output files. You need to get the following files:
For version 1.3.x:

- Heavy.pdb which contains the heavy atom coordinates.
- FOMstats.html which contains figure of merit statistics.
- Name.sin which is a generated input scripts. It has all the input information.
- Otherstat.html which contains Rcullis, Rkraut, phasing power.

For version 2.0 or above:

- Heavy.pdb which contains the heavy atom coordinates.
- FOMstats.html which contains figure of merit statistics.
- Name.sin which is a generated input scripts. It has all the input information.
- RCullis_?.html which contains Rcullis.
- PhasingPower_?.html which contains phasing power

The easiest way to obtain these files is to run the program from the SUSHI interface. Review all the log files from the internet browser and save the files as plain txt (or html) files.

To extract phasing information, you need the following description:
pdb_extract -o test.mmcif **–e** method **–p** SHARP **–iPDB** heavy.pdb \
–iLOG FOMstats.html Otherstat.html Name.sin

6.3.5 Using SnP (http://www.hwi.buffalo.edu/SnB/)

SnB is graphic interface software using the *Shake-and-Bake* algorithm. It has no heavy atom parameter refinement, and it has no corresponding statistics. However, SnB gives the heavy atom or substructure coordinates (e.g. heavy.pdb) in PDB format.

You can use the following command to extract the heavy atom PDB coordinates:
pdb_extract -o test.mmcif **–e** method **–p** SNB **–iPDB** heavy.pdb

Note: Sometimes the coordinates and other parameters can be refined by other programs like MLPHARE or CNS. Therefore, the heavy atom phasing information as well as the heavy atom coordinates will be extracted from MLPHARE or CNS instead of SnB even though SnB may have been used to find the initial heavy atom positions.

6.3.6 Using BnP (http://www.hwi.buffalo.edu/BnP/)

BnP is a combination of program SnB and Phases by Furry. The heavy atom positions are located by SnB and the heavy atom parameters will be refined by Phases.

The log file (for example auto.log) can be found from the directory ~/PHASES/*. Log file normally contains phasing power for each phasing set.

You can use the following command to extract the heavy atom PDB coordinates:
pdb_extract -o test.mmcif **–e** method **–p** BnP **–ilog** auto.log **–iPDB** heavy.pdb
or
pdb_extract -o test.mmcif **–e** method **–p** phases **–ilog** auto.log **–iPDB** heavy.pdb

6.3.7 Using SHELXD or SHELXS (http://shelx.uni-ac.gwdg.de/SHELX/)

They are pretty much the same as SnB. Only heavy atom or substructure coordinates are produced in PDB format (e.g. heavy.pdb).

You can use the following command to extract the heavy atom PDB coordinates:
pdb_extract -o test.mmcif **–e** method **–p** SHELXD **–iPDB** heavy.pdb
or

pdb_extract -o test.mmcif **–e** method **–p** SHELXS **–iPDB** heavy.pdb

6.3.8 Using PHASES (http://imsb.au.dk/~mok/phases/phases.html)

PHASES is a package developed by Furry. It can be used to located heavy atom positions and refine the heavy atom parameters.

The log file (for example name.log) can be found from the directory ~/PHASES/*. Log file normally contains phasing power for each phasing set.

You can use the following command to extract the heavy atom PDB coordinates:
pdb_extract -o test.mmcif **–e** method **–p** Phases **–ilog** name.log **–iPDB** heavy.pdb

6.4 Programs for density modification

Density modification is normally applied after obtaining phase (or heavy atom coordinates). If you use density modification in you structure determination, you can apply the pdb_extract program to extract some statistics from the generated log files.

The following items may be extracted:
- Density modification method.
- FOM after density modification (overall, resolution shells)
- Solvent mask determination method.
- Structure solution software.

6.4.1 Using CNS/XPLOR (http://cns.csb.yale.edu/v1.1/)

The CNS user may need to run the input script like ‘density_modify.inp’. You will get a log file called ‘density_modify.list’.

The command to extract density modification statistics is:
pdb_extract -o test.mmcif **–e** method **–d** CNS **–iLOG** density_modify.list

6.4.2 Using DM (http://www.ccp4.ac.uk/dist/html/INDEX.html)

DM is a popular density modification program in the CCP4 suit. When you run DM either by using the CCP4i graphic interface or the script, you will get a log file like ‘dm.log’.

The command to extract density modification statistics is:
pdb_extract -o test.mmcif **–e** method **–d** DM **–iLOG** dm.log

6.4.3 Using SOLOMON (http://www.ccp4.ac.uk/dist/html/INDEX.html)

SOLOMON is also a popular density modification program CCP4 suit. When you run DM either by using the CCP4i graphic interface or the script, you will get a log file like 'Solomon.log'.

The command to extract density modification statistics is:
pdb_extract -o test.mmcif **–e** method **–d** SOLOMON
–iLOG solomon.log

6.4.4 Using RESOLVE (http://www.solve.lanl.gov/)

RESOLVE is a density modification program in the solve/resolve package. Normally it runs together with SOLVE, but one can run it separately. When you run RESOLVE, (resolve ? > resolve.log) you will get a log file like 'resolve.log'.

The command to extract density modification statistics is:
pdb_extract -o test.mmcif **–e** method **–d** RESOLVE **–iLOG** resolve.log

6.4.5 Using SHARP (http://babinet.globalphasing.com/sharp/)

Density modification used in SHARP is actually the DM (version 2.2) or solomon. When you run density modification in SHARP, you will get a log file like 'dm.log'.

The command to extract density modification statistics is:
pdb_extract -o test.mmcif **–e** method **–d** SHARP **–iLOG** dm.log
or
pdb_extract -o test.mmcif **–e** method **–d** dm **–iLOG** dm.log

6.5 Programs for final structure refinement

The structure refinement is performed at the end of structure determination. Normally the atom coordinates are generated in PDB format and the statistics are generated in log files. The pdb_extract program can be applied to extract the following information:

- Number of reflections used in refinement, and in R-Free set.
- Resolution range (highest res. shell)
- R-factor (overall, resolution shells)
- Number of atoms refined
- Cell parameters and space group.
- The xyz coordinates of all the atoms.

- RMS Bond Distances, Bond Angles, Chiral Volume, Torsion Angles
- Isotropic temperature factor restraints
- Non-crystallographic symmetry restraints
- Solvent model used
- Overall Average Isotropic B Factor
- Overall Anisotropic B Factor
- Overall Isotropic B Factor
- Topology/parameter data used to refine deposited model
- Refinement software

6.5.1 Using CNS/XPLOR (http://cns.csb.yale.edu/v1.1/)

CNS is a popular program for final structure refinement. After you finish the structure refinement, you need to run script (deposit_mmcif.inp). It produces a file (e.g. deposit.mmcif) in CIF format. This file contains rich statistic information.

The command to extract refinement statistics and model details is:
pdb_extract -o test.mmcif **–e** method **–r** CNS **–iCIF** deposit.mmcif

6.5.2 Using REFMAC5 (http://www.ccp4.ac.uk/dist/html/INDEX.html)

REFMAC5 is a popular program for structure refinement (also supported in the CCP4 suite). If you run this program using CCP4i or the script, you need to ask the program to write a harvesting file. When you use the CCP4i interface, select the data harvest button. When you use script mode, do not give command word NOHARV. After you finish running this program, you will get a file (e.g. name.refmac) which is in CIF format. It contains all the information for structure refinement. You will also get a PDB file (e.g. name.pdb). This file contains all the atom coordinates, Bfactors, etc.

The command to extract refinement statistics and model details is:
pdb_extract -o test.mmcif **–e** method **–r** REFMAC5 **–iCIF** \
name.refmac **–iPDB** name.pdb

6.5.3 Using SHELXL (http://shelx.uni-ac.gwdg.de/SHELX/)

SHELXL is a sub_program in the SHELX package. It is used for structure refinement.
After you finish structure refinement, you need to run the shelxpro interactive program and use option B. After going through the shelxpro, you will get a PDB file (e.g. name.pdb) with header information.

The command to extract refinement statistics and model details is:

pdb_extract -o test.mmcif **–e** method **–r** SHELXL **–iPDB** name.pdb

6.5.4 Using TNT (http://www.uoxray.uoregon.edu/tnt/welcome.html)

TNT is a crystal structure refinement program. Data from this program can be extracted from the output PDB file and some LOG files. You can use the **to_pdb** command to convert coordinates in TNT format (name.cor) to the PDB format (name.pdb).

The command is: **to_pdb** name.cor

After finishing refinement, you must use command **rfactor** to generate a log file (e.g. rfactor.log) which contains the refinement statistics.

The command is: **rfactor** name.cor > rfactor.log

To extract the symmetry information, user must provide the symmetry file (e.g. p6122.dat). The information is in the control file name.tnt

The complete command to extract information from TNT should be the following:
pdb_extract –r TNT **–iLOG** p6122.dat rfactor.log **–iPDB** name.pdb

6.5.5 Using ARP/wARP (http://www.embl-hamburg.de/ARP/)

ARP/wARP is an automatic program for structure solution and refinement. REFMAC5 is used for the structure refinement step.

The new version (6.0) can use CCP4i as graphic interface. You can run this program either by CCP4i or by script. You will get a log file (for example warpNtrace_refine.log). You also get a PDB file like warpNtrace.pdb.

To extract phasing information, you need the following description:

pdb_extract -o test.mmcif **–e** method **–r** WARP **–iLOG** warpNtrace_refine.log \
–iPDB warpNtrace.pdb

6.5.6 Using RESTRAIN (http://www.ccp4.ac.uk/dist/html/INDEX.html)

RESTRAIN is a program for structure refinement (also used in the CCP4 suite). Do not give command word NOHARV, when you use script mode. After you finish running this program, you will get a file (e.g. name.restrain) which is in CIF format. It contains all the information for

structure refinement. You will also get a PDB file (e.g. name.pdb). This file contains the atom coordinates, Bfactors, etc.

The command to extract refinement statistics and model details is:

pdb_extract -o test.mmcif **–e** method **–r** RESTRAIN **–iCIF** name.restrain \
–iPDB name.pdb

7 Program argument description

There are three components (**pdb_extract**, **pdb_extract_sf**, and **extract**) for the program.

7.1 Unix argument description and options for pdb_extract

NAME **pdb_extract**

SYNOPSIS **pdb_extract** [OPTIONs]... [FILEs]...

DESCRIPTION

PDB_EXTRACT is used to extract information about data processing, heavy atom phasing, molecular replacement, density modification, and final structure refinement from the output files produced by many X-ray crystallographic applications, and merges the information into macromolecular Crystallographic Information File (mmCIF) data files.

User can get help by typing 'pdb_extract –h or pdb_extract –help' to get information how to do extractions and deposition to PDB

OPTIONS

-o Followed by a given output file name.

For example: **-o** outfile.mmcif

NOTE: if you do not give this description, the default output file name (pdb_extract.mmcif) will be used.

-e Followed by one of the following experimental methods:

- **MR** molecular replacement.
- **SAD** single anomalous diffraction.
- **MAD** multiple anomalous diffraction.
- **SIR** single isomorphous diffraction.
- **SIRAS** single isomorphous with anomalous diffraction.
- **MIR** multiple isomorphous diffraction.
- **MIRAS** multiple isomorphous with anomalous diffraction.

example: **-e MAD**

Note: If your experiment is solved by combinations of above methods (e.g. **MR** with **MAD**), you may extract things from both methods (e.g. **-e** MR **–m** program_mr **–ilog** Log_file **–e** MAD **–p** program_mad **–ilog** file_name)

-m Followed by the one of following programs for molecular replacement:

- **CNS** (versions 1.0 and 1.1).
- **Amore** from CCP4 suite (versions 4.1- above).
- **EPMR** (versions 2.5).
- **MOREP** from CCP4 suite (versions 4.1-above).
- **Phaser** from CCP4 suite (versions 4.1-above**).**

For example: **-m amore**

-p Followed by the one of following program names for phasing:

- **CNS** (versions 1.0 and 1.1).
- **MLPHARE** from CCP4 suite (versions 4.0- above).
- **SOLVE** (versions 2.00-2.06).
- **SHARP** (versions 1.3.x – 2.03).
- **SHELXS** (version 97).
- **SHELXD** (version 97).
- **SnB** (version 2.2).
- **BnP** (version 0.93-0.96).
- **PHASES** (version 0.97).

For example: **-p CNS**

Note: if the program that you used for phasing is not in the above list, you may still give the program name. Some information (like heavy atom coordinates) may still be extracted, if the produced file is in PDB or mmCIF format. (use **–p** program_name)

-d Followed by the one of following program names for density modification:

- **CNS** (versions 1.0 and 1.1).
- **DM** from CCP4 suite (CCP4 versions 4.0~5.0).
- **SOLOMON** from CCP4 suite (CCP4 versions 4.0~5.0).
- **RESOLVE** (versions 2.01~2.06).
- **SHELXE** (version 97).
- **SHARP** (version 1.3.x-2.03. using DM version 2.2 for density modification).

For example: **-d CNS**

-r Followed by one of the following program names for final structure refinement.

- **CNS** (versions 1.0 and 1.1).
- **REFMAC5** from CCP4 suite version 4.1-5.0 (REFMAC version 5.2).

- **RESTRAIN** from CCP4 suite version 4.1-5.0 (RESTRAIN v4.6).
- **SHELXL** (version 97).
- **TNT** (version 5F).
- **WARP** (version 6.0, It uses REFMAC5 for refinement)

For example: **-r CNS**

Note: if the program that you used for final structure refinement is not in the above list, you may still give the program name. Some information (like atom coordinates) may still be extracted, if the produced file is in PDB or CIF format. (use **–r** program_name)

-s Followed by one of the following programs for data scaling (for refinement):

- **HKL/HKL2000/SCALEPACK** (versions 1.30 ~ 1.96).
- **SCALA** (version 3.1.4 ~3.2.3) or from CCP4 suite version 4.1-5.0
- **D*trek** (version 7.0SSI)
- **SAINT** (version 6.35A)
- **3DSCALE**

For example: **-s HKL**

Note: The **–s** option here is only used to get statistics from data reduction. The reflection data must be used to the final structure refinement.

-sp Followed by one of the following programs for data scaling (for refinement):

- **HKL/HKL2000/SCALEPACK** (versions 1.30 ~ 1.96).
- **SCALA** (version 3.1.4 ~3.2.3) or from CCP4 suite version 4.1-above
- **D*trek** (version 7.0SSI)
- **SAINT** (version 6.35A)
- **3DSCALE**

For example: **-sp HKL**

Note: The option is similar to **–s**, but it is used to extract statistics from multiple data reductions. The reflection data sets must be used to protein phasing solutions (SAD, MAD, SIR, MIR …). Normally, there are multiple data sets.

-iPDB Followed by a input file with PDB format.

For example: **-iPDB** test1.pdb

NOTE: The PDB files are usually generated from heavy atom phasing (heavy atom coordinates) or the final structure refinement.

-iCIF Followed by a input file with CIF format.

For example: **-iCIF** deposit_cns.cif

NOTE: This file can be produced during crystal structural determination. For instance: if you use MLPHARE for locating heavy atom position and do heavy atom phasing refinement, a file in mmCIF format will be generated. This file will contain statistics for heavy atom phasing. Another instance, if you use CNS for final structure refinement, running the deposit.inp macro will produce a CIF file containing the model coordinates and refinement statistics.

-iLOG Followed by one or more input LOG files

For example: **-iLOG** mad_sdb.dat mad_summary.dat

NOTE: Log files are usually generated during crystal structural determination. The format depends on the program used. They may contain phasing statistics or heavy atom coordinates. For instance, when people use CNS for heavy atom phasing, they will generate a file (e.g. mad_sdb.dat) which contains the heavy atom coordinates and a file (e.g. mad_summary.dat) which contains phase refinement statistics.

-iENT Followed by the either a mmCIF file or the pdb_extract generated plain text file.

For example: **-iENT** data_template.text or **-iENT** entity-poly.mmcif

NOTE: This file should be generated by the program **extract**. It contains the full chemical sequence and related information to be filled for each macromolecule in the solved structure. The format is either mmCIF (entity-poly.mmcif) or a plain text (data_template.text). The chemical sequences are grouped in each molecular entity which is defined as a unique monomer in asymmetric unit. For example, if there are several copies (chain A, B, ..) of a molecule in asymmetric unit (e.g. dimmer, trimmer ..), the molecular entity is only one. This file can be found in each example. For details of the definition, please see the website: http://pdb.rutgers.edu/mmcif.

The file (data_template.text or entity-poly.mmcif) have to be editored or modified (as necessary) before it is used with **pdb_extract**.

For convenience, user can use the following commands to generate the file “entity-poly.text” from either your PDB file or your mmCIF file.

extract –pdb pdb_file_name (coordinate file in PDB format)
or
extract –cif cif_file_name (coordinate file in mmCIF format (e.g. CNS))

7.2 Examples for using pdb_extract options

Note: you can extract statistics separately from each step of structure determination applications (data processing, heavy atom phasing, density modification, molecular replacement and final structure refinement), or you can put all the steps together, which is a complete deposition.

Command for extracting information about heavy atom phasing:
(The experimental_method must be given for this step)
pdb_extract -e experimental_method **-p** program_name_phasing \
-iPDB pdb_files **–iLOG** log_files \
–iCIF mmCIF_files **-o** output_file_name

Command for extracting information about density modification (output from this program is normally the LOG file):
pdb_extract -d program_name_for_dm **–iLOG** log_files **-o** output_file_name

Command for extracting information about molecular replacement (output from this program is normally the LOG file):
pdb_extract -m program_name_for_mr **–iLOG** log_files **-o** output_file_name

Command for extracting information from final structure refinement:
pdb_extract -r program_name_for_refinement **-iPDB** pdb_files \
–iLOG log_files **–iCIF** mmCIF_files **-o** output_file_name

Command for extracting information from data scaling LOG files (for refinement):
pdb_extract -s program_name_scaling **–iLOG** log_file **-o** output_file_name

Note: HKL/SCALEPACK does not export <I/SimgaI>, but it is needed in the PDB deposition. pdb_extract can calculate this for you when providing the data for refinement. The command is
pdb_extract -s program_name_scaling **–iLOG** log_file **-o** output_file_name \
-idat reflection_data_file

Command for extracting information from data scaling LOG files (for phasing):
pdb_extract -sp program_name_scaling **–iLOG** log_file1 log_file2 **-o** output_file_name

Command for extracting information for a complete structure:

pdb_extract -e experimental_method **-r** rogram_name_for_refinement \
-iPDB pdb_files **–iLOG** log_files **–iCIF** mmCIF_files \
-p program_name_for_phasing **-iPDB** pdb_files \
–iLOG log_files **–iCIF** mmCIF_files \
-d program_name_for_dm **–iLOG** log_files \
-s program_name_for_scaling **–iLOG** log_files \
-sp program_name_for_scaling **–iLOG** log_files \
-iENT data_template.text **-o** output_file_name

7.3 Unix argument description and options for pdb_extract_sf

NAME **pdb_extract_sf**

SYNOPSIS **pdb_extract_sf** [OPTIONs]... [FILEs]...

DESCRIPTION

This program can be used to capture
1. Reflection data used for final structure refinement.
2. Multiple reflection data (eg. MAD, MIR …) processed by the software at the data collection site.

OPTIONS

-o Followed by an output file name.

For example: **-o** outfile.cif

NOTE: if you do not specify an output file, a default output file name (pdb_extract_sf.mmcif) will be used.

-dt Data type for initial data processing (normally intensity).
It is followed by F (Amplitude) or I (Intensity)

For example: **-dt** I

-dp Data format for initial data processing.
It is followed by one of the following program names:
HKL/SCALEPACK, DTREK, SAINT, XPREP, 3DSCALE, SCALA, OTHER.
For example: **-dp** HKL

NOTE1: If the program for data scaling is not in the above list, please give OTHER. Then you must provide either a plain text file with reflection data in the order of H, K, L, (F or I), (sigmaF or sigmaI) separated by spaces in each row, or a mmCIF file: e.g. **–dt** I **–dp** OTHER **–idat** file_name

NOTE2: If the output data is in mtz format (processed by MOSFILM and SCALA), you must convert the data either to CNS format or scalepack format or mmCIF format. If it is CNS format, you will use **-dp** CNS **–idat** file-name. If it is scalepack format, you will use **-dp** HKL **–idat** file-name. If it is mmCIF format, you can use **-dp** SCALA **–idat** file-name.

-c crystal index.

It is followed by crystal number (integers, like 1,2,3, ..)

Example: **-c** 2
(It means the reflection was from the second crystal).

-w wavelength index.

It is followed by wavelength number (integers, like 1, 2, 3)

Example: **-w** 2
(This means the data was collected from the crystal using the second wavelength).

-idat reflection data file

It is followed by data file name

Example: **-idat** scalepack.sca

NOTE: You should always give the combination **–c** *i*, **-w** *j* **–idat** *file_name* in the right order! Here *i* is the crystal index, *j* is wavelength index, and *file_name* is the file name containing the reflections.

-rt data type used for final structure refinement.

It is followed by F (Amplitude) or I (Intensity)

For example: **-dt** F

-rp data format in the final structure refinement.

It is followed by the data format name: CNS/XPLOR, REFMAC5, SHELX, TNT, HKL/SCALEPACK, DTREK, SAINT, XPREP, 3DSCALE, SCALA,

Example: **-rp** CNS

NOTE: If you used REFMAC5 to do final structure refinement, you need convert the MTZ file to mmCIF format or CNS format.

If it is converted to mmCIF format, you can use the command
pdb_extract_sf –rt I **–rp** REFMAC5 **–idat** data-file-name

If it is converted to CNS format, you can use the command
pdb_extract_sf –rt I **–rp** CNS **–idat** data-file-name

NOTE1: If the program for refinement is not in the above list, please give OTHER**.** Then you must provide either a plain text file with reflection data in the order of H, K, L, (F or I), (sigmaF or sigmaI) separated by spaces in each row, or a mmCIF file: e.g. **–rt** F **–rp** OTHER **–idat** file_name

-imgCIF file name for imgCIF format.
It is followed by the input file name:

Example: **-imgCIF** example.cbf
NOTE: Only the head information of the imgCIF file can be extracted!

7.4 Examples for using pdb_extract_sf options

Extracting reflection data used for final structure refinement:
pdb_extract_sf -rt data-type **-rp** data-format-for-refinement \
-idat data-file-name **–o** output-file-name

NOTE: Normally, there is only one data set. If you have several data set used for final refinement, you need to merge all the data in one file.

Extracting reflection data from initial data process (e.g. scaling …):
pdb_extract_sf -dt data_type **-dp** program_name_for_scaling \
-c crystal_number_1 **-w** wavelength_number_1 **-idat** data_file_name_1 \
-c crystal_number_2 **-w** wavelength_number_2 **-idat** data_file_name_2 \
…
–o output_file_name

NOTE: Normally, there are several data sets (e.g. in MAD, MIR …). These reflections are used for protein phasing. The formats are from the initial data process.

Converting all the reflection data in one mmCIF file (just combine the above two steps):

pdb_extract_sf **-rt** data-type_refine **-rp** data-format-for_refine \
-idat data-file-name_refine **-dt** data_type_scaling \
-dp program_name_for_scaling\
-c crystal_number_1 **-w** wavelength_number_1 **-idat** data_file_name_1 \
-c crystal_number_2 **-w** wavelength_number_2 **-idat** data_file_name_2 \
…
–o output_file_name

The output_file_name contains the reflections for refinement and the reflections for protein phasing.

7.5 Unix argument description and options for extract

NAME **extract**

SYNOPSIS **extract** [OPTIONs] [FILE]

DESCRIPTION

This program can be used to do the following:

1. Generate the plain text file (data_template.text) used with pdb_extract.
   Generated the plain text file (log_script.inp) used by extract itself.
2. Run all the commands in the script input file (log_script.inp).

OPTIONS

-pdb Followed by the coordinate PDB file name

example **-pdb** pdb_file_name

NOTE: it will generate two plain text files (data_template.text and log_script.inp) with the chemical sequences extracted from the coordinate PDB file.

-cif Followed by the coordinate mmCIF file name

example **-cif** mmCIF_file_name

NOTE: it will generate two plain text files (data_template.text and log_script.inp) with the chemical sequences extracted from the coordinate mmCIF file.

-ext Followed by the generated file log_script.inp

example **-ext** log_script.inp

NOTE: you just fill the LOG files and some additional information to the log_script.inp. Then use '**extract –ext** log_script.inp ' to get a complete mmCIF file and structure file.

7.6 Examples for using extract options

Obtain the input file entity-poly.inp
extract -pdb pdb_file_name or **extract -cif** cif_file_name

Get a complete mmCIF file for deposition

extract -ext log_script.inp

8 Tables

8.1 Unix command options

Unix command line options for the three components of PDB_EXTRACT. **pdb_extract_sf** is used to capture structure factors, and **pdb_extract** is used to capture the details of molecular replacement, heavy atom phasing, density modification and structure refinement, and e**xtract** is used to generate entity-poly.text and entity-poly.inp.

pdb_extract_sf [OPTION]... [FILE]...	
Option	Argument descriptions
-o	output file name (default name is pdb_extract_sf.mmCIF).
-dt	data type (I or F) after data processing at beam line.
-dp	program for processing data (e.g. HKL/Scalepack, D*Trek, SCALA).
-rt	data type (I or F) used for final structure refinement.
-rp	program for structure refinement (e.g. CNS|REFMAC5|SHELX|TNT).
-c	crystal number (like 1, 2, 3 …) for diffraction.
-w	wavelength number (like 1, 2, 3 …) for diffraction.
-idat	data file name used for phasing or structure refinement.
-ilog	log file name obtained from data processing.
-icif	file name obtained from data processing (in mmCIF format).
pdb_extract [OPTION]... [FILE]...	
Option	Argument descriptions
-o	output file name (default name is pdb_extract.mmcif)
-e	experimental method (eg. MR|SAD|MAD|SIR|MIR|SIRAS|MIRAS)
-m	program for molecular replacement (e.g. CNS|AMORE|MOLREP|EPMR)
-p	program for heavy atom phasing (e.g. CNS|MLPHARE|SOLVE|SHARP|SHELXD|SnB|BnP)
-d	program for density modification (e.g. CNS|DM|SOLOMON|RESOLVE)
-r	program for final structure refinement (e.g. CNS|REFMAC5|RESTRAIN|SHELXL|TNT|WARP)
-s	program for reflection data scaling (only for refinement) (e.g. HKL/Scalepack, D*Trek, SAINT, SCALA, 3DSCALE).
-sp	program for reflection data scaling (only for phasing) (e.g. HKL/Scalepack, D*Trek, SAINT, SCALA, 3DSCALE).
-ilog	the input file with format corresponding to the program used.
-ipdb	the input file with PDB format.
-icif	the input file with mmCIF format.
-ient	the input file entity-poly.text. (for complete sequence.)
extract [OPTION] [FILE]	
Option	Argument descriptions
-pdb	input coordinate file name (PDB format)
-cif	input coordinate file name (mmCIF format)
-ext	input script file name entity-poly.inp

8.2 Crystallographic software list

Crystallographic software applications supported by PDB_EXTRACT.

Category	Software	Versions	Authors
Data collection and reduction	HKL/SCALEPACK	1.30 - 1.96	Otwinowski & Minor (1997)
	d*TREK	7.0SSI	Pflugrath (1997)
	SAINT	V6.35A	Siemens (1994)
	SCALA	3.1.4 - 3.2.3	Evans (1997)
Molecular replacement	CNS	0.9 - 1.1	Brunger et al. (1998)
	Amore	CCP4 (4.0 - 5.0)	Navaza (1994)
	Morep	7.5.01	Vagin & Teplyakov (1997)
	EPMR	2.5	Kissinger et al. (1999)
Heavy atom phase determination	CNS	0.9 - 1.1	Brunger et al. (1998)
	SOLVE	2.0 - 2.06	Terwilliger & Berendzen. (1999)
	MLPHARE	CCP4 (4.0 - 5.0)	CCP4 (1994)
	SHARP/autoSHARP	1.3.x - 2.02	Fortelle & Bricogne (1997)
	SHELXD/SHELXS	97	Sheldrick (1997)
	PHASES	95	Furry (1997)
	SnB	2.0 - 2.2	Weeks & Miller (1999).
	BnP	0.93 - 0.94	Weeks et al. (2002)
Density modification	CNS	0.9 - 1.1	Brunger et al. (1998)
	DM	2.0 - 2.1	Cowtan (1994)
	Solomon	CCP4 (4.0 - 5.0)	Abrahams & Leslie (1996)
	RESOLVE	2.0 - 2.06	Terwilliger (2000)
	SHELXE	97	Sheldrick (1997)
Structure refinement	CNS	0.9 - 1.1	Brunger et al. (1998)
	REFMAC5	5.0 - 5.2	Murshudov (1997)
	RESTRAIN	4.7.7	CCP4 (1994)
	SHELXL	97	Sheldrick (1997)
	TNT	5F	Tronrud (1997)
	WARP	5.0 - 6.0	Lamzin & Wilson, (1997)

9 References

H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne(2000), "The Protein Data Bank". Nucleic Acids Research, 28 235-242

P.W. Haebel, V.L. Arcus, E.N. Baker, and P. Metcalf(2001). Acta Cryst. D57, 1341-1343

M. Harris and T.A. Jones(2002). Acta Cryst. D58, 1889-1891

Z. Otwinowski and W. Minor, " Processing of X-ray Diffraction Data Collected in Oscillation Mode ", Methods in Enzymology, Volume 276: Macromolecular Crystallography, part A, p.307-326, 1997

Pflugrath JW (1999). "The finer things in X-ray diffraction data collection." Acta Cryst. Section D-Biological Crystallography. 55 1718-25

Evens, P. R. (1997). "the Scala" Joint CCP4 and ESF-EACBM Newsletter 33, 22-24

Brunger, A.T., Adams, P.D., Clore, G.M., DeLano, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.-S., Kuszewski, J., Nilges, N., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T., and Warren, G.L. (1998). Crystallography and NMR
system (CNS): A new software system for macromolecular structure determination, Acta Cryst. D54, 905-921.

Terwilliger, T.C. and J. Berendzen. (1999) "Automated MAD and MIR structure solution". Acta Crystallographica D55, 849-861.

COLLABORATIVE COMPUTATIONAL PROJECT, NUMBER 4. 1994. ``The CCP4 Suite: Programs for Protein Crystallography". Acta Cryst. D50, 760-763

E. de La Fortelle & G. Bricogne (1997) "Maximum-Likelihood Heavy-Atom Parameter Refinement for the Multiple Isomorphous Replacement and Multiwavelength Anomalous Diffraction Methods" Methods in Enzymology 276 472-494

Abrahams J. P. and Leslie A. G. W., Acta Cryst. D52, 30-42 (1996)

K. Cowtan (1994), Joint CCP4 and ESF-EACBM Newsletter on Protein Crystallography, 31, p34-38.

Terwilliger, T. C. (2000) "Maximum likelihood density modification," Acta Cryst. D56, 965-972.

Weeks, C.M. & Miller, R. (1999). 'The design and implementation of SnB v2.0', J. Appl. Cryst.32, 120-124.

Tronrud, D, E., (1997). " The TNT Refinement Package" in Macromolecular Crystallography, Part B, Methods Enzymol. 277, 306-318

Lamzin, V.S. & Wilson, K.S. (1997). "Automated refinement for protein crystallography", Methods Enzymol. (Carter, C. & Sweet, B. eds.) 277, 269-305

G.N. Murshudov, A.A.Vagin and E.J.Dodson, (1997) "Refinement of Macromolecular Structures by the Maximum-Likelihood Method" Acta Cryst. D53, 240-255.

Weeks, C.M., Blessing, R.H., Miller, R., Mungee, S., Potter, Rappleye, A., Simith, G.D. Xu, H., Furey, W. (2002), "Towards automated protein structure determination: BnP, the SnB-PHASES Interface ". Z. Kristallogr. 217, 686-693

Furey, W. & Swaminathan, S. (1997), " PHASES-95: A Program Package for the Processing and Analysis of Diffraction Data from Macromolecules". Methods in Enzymology, 277, 590-620

Sheldrick G. (1997) "The SHELX-97 homepage" http://shelx.uni-ac.gwdg.de/SHELX/

Navaza J. (1994) "AMoRe: an Automated Package for Molecular Replacement",*Acta. Cryst.* **50**, 157-163.

Vagin A. , Teplyakov A. (1997) , " MOLREP: an automated program for molecular replacement" *J. Appl. Cryst.* **30**, 1022-1025.

Charles R. Kissinger, Daniel K. Gehlhaar & David B. Fogel, (1999) "Rapid automated molecular replacement by evolutionary search", *Acta Cryst*. , **D55**, 484-491

10 Frequently asked questions

1. Question: What should I do, if the program that I used for solving a structure is not checked by pdb_extract?

Answer: If the program exports log files in mmCIF format or the PDB format for atomic coordinates, you just give the program name, some information will be extracted. However, if the unknown program only generates LOG file which is neither mmCIF no PDB format, please send us help@rcsb.rutgers.edu the log file and the program name. We will add the program to our list.

2. Question: If it takes really long time between each crystallographic step (like from phasing to refinement), I may not keep the old log files.

Answer: I suggest you apply the pdb_extract program as soon as you finished this step. Then, you will generate one mmCIF file for this step. You may only keep this mmCIF file somewhere in your disk. Finally, you just use the same program to merge all the steps together. (Your options should all be **–icif** cif_file_name …).

3. Question: How do I know that I obtained the correct mmCIF file?

Answer: Normally the program gives a warning message. But it is a good idea to check if the mmCIF file has the right PDB coordinates (_atom_site. ?). If you encounter an error when running the program, please take a look if you used the correct options. Otherwise, send a message to help@rcsb.rutgers.edu

4. Question: I have installed the CCP4 suit. do I have to install the pdb_extract again.

Answer: You do not have to install the standalone version of pdb_extract, if you prefer to do validation by the ADIT server. In addition to using the CCP4i interface, you can also do all the Unix command line option under the CCP4 environment.

11 Appendix (*data_template.text*)

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        THE DATA_TEMPLATE.TEXT FILE
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                           NOTES AND REMINDER
The data template file contains data entries for unique chemical sequences
present in the structure and other non-electronically captured information.

PLEASE CHECK CATEGORIES 1 & 2: Before proceeding any further, make necessary
corrections here so that all information in these categories are complete
and correct.

You may choose to fill in CATEGORIES (3-19) either here or later in ADIT.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                        GUIDELINES FOR USING THIS FILE
  1. Only strings included between the 'lesser than' and 'greater than'
     signs (<.....>) will be parsed for evaluation by the program. Therefore,
     DO NOT write either on the left or right of the 'less than' and 'greater
     than' signs respectively.

  2. All alphanumeric values or strings that you include in the different
     categories should be within double-quotes. Blank spaces or carriage
     returns within a pair of double quotes are ignored by the program.
     DO NOT use double quotes (") within strings that you enter.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

~~~~~~~~~~~~~~~~~~~~~~~~~~~~START INPUT DATA BELOW~~~~~~~~~~~~~~~~~~~~~~~~

================CATEGORY 1:   Crystallographic Data=======================
Enter crystallographic data

<space_group = "       C 2"> (use International Table conventions)
<space_group_number = "? ">

<unit_cell_a     = " 108.742 " >
<unit_cell_b     = "  61.679 " >
<unit_cell_c     = "  71.652 " >
<unit_cell_alpha = " 90.00 " >
<unit_cell_beta  = " 97.15 " >
<unit_cell_gamma = " 90.00 " >


================CATEGORY 2:   Sequence Information ======================
Enter one letter sequence for each polymeric entity in asymmetric unit

-------------------------------------------------------------------------
                          SOME DEFINITIONS
     An ENTITY is defined as any unique molecule present in the asymmetric
     unit. Each unique biological polymer (protein or nucleic acids) in the
     structure is considered an entity. Thus, if there are five copies of
     a single protein in the asymmetric unit, the molecular entity is still
     only one. Water and non-polymers like ions, ligands and sugars are
     also entities.

     Here we only consider the sequences of polymeric entities (protein or
     nucleic acid).

                GUIDELINES FOR COMPLETING THIS CATEGORY
      * In a PDB or mmCIF format file, all residues of a single polymeric
      entity should have one chain ID. Multiple copies of the same entity
      should each be assigned a unique chain ID. The multiple chain IDs
      should be separated by commas as 'A,B,C,...'. If incorrect chain IDs
      are used the entity groups extracted by this program will not be
```

correct. To avoid this, make necessary corrections in the PDB or mmCIF
file used to generate the data_template file and regenerate the
data_template.text file. Alternatively, edit the extracted sequence
in this file to correctly represent the sequence and chain IDs of each
polymeric entity.

* In addition to chain IDs, this program uses distance geometry to
asses if there are any breaks in the polymer sequence. These breaks
may occur due to missing residues (not included in the model due to
missing electron density) or due to poor geometry. Four question marks
'????' are used to denote these chain breaks. Replace these question
marks with the sequence of residues missing from the coordinates. Also
add any residues missing from the N- and/or C-termini here.

* If there are non-standard residues in the coordinates, this program
lists them according to the three letter code used in the coordinate
file as (ABC). If all the residues in your sequence are nonstandard,
check and edit the sequence manually to represent it correctly in this
file.

* If any residue was modeled as Ala or Gly due to lack of the side-chain
density, the sequence extracted here will represent them as A or G
respectively. Correct this to the original sequence that was present in
the crystal.

Below is the one letter chemical sequence extracted from your PDB
coordinate file. The molecular entities are grouped and listed
together.

PLEASE CHECK THE SEQUENCE of each entity carefully and modify it, as necessary.
Make sure that you REVIEW THE FOLLOWING:
* chain breaks due to missing residues,
* missing residues in the N- and/or C-termini,
* non-standard residues and
* cases of residues modeled as Ala or Gly due to missing side-chain density.

<molecule_entity_id="1" >
<molecule_entity_type="polypeptide(L)" >
<molecule_one_letter_sequence="
SKPLLTKREREVFELLVQDKTTKEIASELFISEKTVRNHISNAMQKLGVKGRSQAVVELLRMGELEL" >
< molecule_chain_id="1,2,3,4,5,6" >

<molecule_entity_id=" " >
<molecule_entity_type=" " >
<molecule_one_letter_sequence=" " >
<molecule_chain_id=" " >

==========CATEGORY 3: Contact Authors==========
Enter information about the contact authors.

Information about the Principal investigator (PI) should be given.

For principal investigator
<contact_author_PI_name = " "> (Surname, F.M.)
<contact_author_PI_email = " ">
<contact_author_PI_phone = " ">
<contact_author_PI_fax = " ">
<contact_author_PI_address = " ">

For other contact authors
<contact_author_name_1 = " ">
<contact_author_email_1 = " ">
<contact_author_phone_1 = " ">
<contact_author_fax_1 = " ">
<contact_author_address_1 = " ">

<contact_author_name_2 = " ">

```
<contact_author_email_2 = " ">
<contact_author_phone_2 = " ">
<contact_author_fax_2 = " ">
<contact_author_address_2 = " ">

...(add more if needed)...

================CATEGORY 4:   Release Status=============================
Enter release status for the coordinates, constraints and sequence

   Status should be chosen from one of the following:
  (release now, hold for publication,  hold for 6 weeks,  hold for 6 months,
   hold for 1 year)

<Release_status_for_coordinates = " ">
<Release_status_for_structure_factor = " ">
<Release_status_for_sequence = " ">

================CATEGORY 5:   Title======================================
Enter the title for the structure

<structure_title = " ">

================CATEGORY 6:   Citation Authors===========================
Enter citation authors (e.g. Surname, F.M.)

      The primary citation is the article in which the deposited coordinates
      were first reported. Other related citations may also be provided.

For the primary citation
<primary_citation_author_name_1 = " ">
<primary_citation_author_name_2 = " ">
<primary_citation_author_name_3 = " ">
<primary_citation_author_name_4 = " ">
<primary_citation_author_name_5 = " ">
...add more if needed...

For other related citations (if applicable)
<citation_1_author_name_1 = " ">
<citation_1_author_name_2 = " ">
<citation_1_author_name_3 = " ">
<citation_1_author_name_4 = " ">
<citation_1_author_name_5 = " ">
...add more if needed...

<citation_2_author_name_1 = " ">
<citation_2_author_name_2 = " ">
<citation_2_author_name_3 = " ">
<citation_2_author_name_4 = " ">
<citation_2_author_name_5 = " ">
...add more if needed...

...(add more citations if needed)...

================CATEGORY 7:   Citation Article===========================
Enter citation article (journal, title, year, volume, page)

      If the citation has not yet been published, use 'To be published'
      for the category 'journal_abbrev'. The order of citations in this
      category should correspond to that is CATEGORY 6.

For primary citation
<primary_citation_journal_abbrev = " ">
<primary_citation_title = " ">
<primary_citation_year = " ">
<primary_citation_journal_volume = " ">
<primary_citation_page_first = " ">
<primary_citation_page_last = " ">

For other related citation (if applicable)
```

```
<citation_1_journal_abbrev = " ">
<citation_1_title = " ">
<citation_1_year = " ">
<citation_1_journal_volume = " ">
<citation_1_page_first = " ">
<citation_1_page_last = " ">

<citation_2_journal_abbrev = " ">
<citation_2_title = " ">
<citation_2_year = " ">
<citation_2_journal_volume = " ">
<citation_2_page_first = " ">
<citation_2_page_last = " ">

...(add more citations if needed)...

================CATEGORY 8:   Molecule Names==============================
Enter the name of the molecule for each entity

      The name of molecule should be obtained from the appropriate
      sequence database reference, if available. Otherwise the gene name or
      other common name of the entity may be used.
      e.g. HIV-1 integrase for protein
           RNA Hammerhead Ribozyme for RNA
      The number of entities should be the same as in CATEGORY 1.

<molecule_name_1 = " ">    (entity 1)
<molecule_name_2 = " ">    (entity 2)
<molecule_name_3 = " ">    (entity 3)

...(add more if needed)...


================CATEGORY 9:   Molecule Details============================
Enter additional information about each entity

      Additional information would include details such as fragment name
      (if applicable), mutation, and E.C.number.

For entity 1
<Molecular_entity_id_1 = " ">       (e.g. 1, 2, ...)
<Fragment_name_1 = " ">             (e.g. ligand binding domain, hairpin)
<Specific_mutation_1 = " ">         (e.g. C280S)
<Enzyme_Comission_number_1 = " ">   (if known: e.g. 2.7.7.7)

For entity 2
<Molecular_entity_id_2 = " ">
<Fragment_name_2 = " ">
<Specific_mutation_2 = " ">
<Enzyme_Comission_number_2 = " ">

For entity 3
<Molecular_entity_id_3 = " ">
<Fragment_name_3 = " ">
<Specific_mutation_3 = " ">
<Enzyme_Comission_number_3 = " ">

...(add more if needed)...


================CATEGORY 10:   Genetically Manipulated Source=============
Enter data in the genetically manipulated source category

      If the biomolecule has been genetically manipulated, describe its
      source and expression system here.

For entity 1
<Manipulated_entity_id_1 = " ">              (e.g. 1, 2, ...)
<Source_organism_scientific_name_1 = " ">    (e.g. Homo sapiens)
<Source_organism_gene_1 = " ">               (e.g. RPOD, ALKA...)
```

```
<Expression_system_scientific_name_1 = " ">  (e.g. Escherichia coli)
<Expression_system_strain_1 = " ">         (e.g. BL21(DE3))
<Expression_system_vector_type_1 = " ">           (e.g. plasmid)
<Expression_system_plasmid_name_1 = " ">     (e.g. pET26)
<Manipulated_source_details_1 = " ">          (any other relevant information)

For entity 2
<Manipulated_entity_id_2 = " ">
<Source_organism_scientific_name_2 = " ">
<Source_organism_gene_2 = " ">
<Expression_system_scientific_name_2 = " ">
<Expression_system_strain_2 = " ">
<Expression_system_vector_type_2 = " ">
<Expression_system_plasmid_name_2 = " ">
<Manipulated_source_details_2 = " ">

For entity 3
<Manipulated_entity_id_3 = " ">
<Source_organism_scientific_name_3 = " ">
<Source_organism_gene_3 = " ">
<Expression_system_scientific_name_3 = " ">
<Expression_system_strain_3 = " ">
<Expression_system_vector_type_3 = " ">
<Expression_system_plasmid_name_3 = " ">
<Manipulated_source_details_3 = " ">

...(add more if needed)...

================CATEGORY 11:   Natural Source=============================
Enter data in the natural source category

      If the biomolecule was derived from a natural source, describe
      it here.

For entity 1
<natural_source_entity_id_1 = " ">          (e.g. 1, 2, ...)
<natural_source_scientific_name_1 = " ">    (e.g. Homo sapiens)
<natural_source_details_1 = " ">            (any other relevant information
                                       e.g. organ, tissue, cell ..)

For entity 2
<natural_source_entity_id_2 = " ">
<natural_source_scientific_name_2 = " ">
<natural_source_details_2 = " ">

for entity 3
<natural_source_entity_id_3 = " ">
<natural_source_scientific_name_3 = " ">
<natural_source_details_3 = " ">

...(add more if needed)...

================CATEGORY 12:   Keywords===================================
Enter a list of keywords that describe important features of the deposited
structure.

      For example, beta barrel, protein-DNA complex, double helix,
      hydrolase, structural genomics etc.

<structure_keywords = " ">

================CATEGORY 13:   Biological Assembly========================
Enter data in the biological assembly category

      Biological assembly describes the functional unit(s) present in the
      structure. There may be part of a biological assembly, one or more
      than one biological assemblies in the asymmetric unit.
      Case 1
      * If the asymmetric unit is the same as the biological assembly
        nothing special needs to be noted here.
```

```
      Case 2
      * If the asymmetric unit does not contain a complete biological unit.
        Please provide symmetry operations including translations required
        to build the biological unit.
        (example:
        The biological assembly is a hexamer generated from the dimer
        in the asymmetric unit by the operations:  -y, x-y-1, z-1 and
        -x+y, -x-1, z-1.)
      Case 3
      * If the asymmetric unit has multiple biological units
        Please specify how to group the contents of the asymmetric unit into
        biological units.
        (example:
        The biological unit is a dimer. There are 2 biological units in the
        asymmetric unit (chains A & B and chains C & D).

For biological unit 1
<biological_assembly_1 = " ">

For biological unit 2
<biological_assembly_2 = " ">

....(add more if needed)....

================CATEGORY 14:   Crystals====================================
Enter the number of crystals used for diffraction

<number_of_crystals = " ">

================CATEGORY 15:   Methods and Conditions=====================
Enter the crystallization conditions for each crystal

For crystal 1:
<crystal_number_1 = " ">            (e.g. 1, 2, ...)
<crystallization_method_1 = " ">      (e.g. vapor diffusion, hanging drop)
<crystallization_pH_1 = " ">          (e.g. 7.5 ...)
<crystallization_temperature_1 = " "> (e.g. 100) (in Kelvin)
<crystallization_components_1 = " ">  (e.g. PEG 4000, NaCl etc.)

For crystal 2:
<crystal_number_2 = " ">
<crystallization_method_2 = " ">
<crystallization_pH_2 = " ">
<crystallization_temperature_2 = " ">
<crystallization_components_2 = " ">

...(add more if needed)...

================CATEGORY 16:   Crystal Property============================
Enter solvent content, Matthews coefficient
      These values were calculated based on the sequence as shown in
      CATEGORY 2. If there are missing residues, you need to add the
      missing residues and re-run the program to get accurate values.
      (The command to re-run is 'extract -sol data_template.text')

For crystal 1:
<crystals_number_1 = " 1 ">                  (e.g. 1, 2, ...)
<crystals_solvent_content_1 = "52.2 ">
<crystals_matthews_coefficient_1 = "2.6 ">
<crystals_mosaicity_1 = " ">    (e.g. 0.5 ...)

...(add more if needed)...

================CATEGORY 17:   Radiation Source============================
Enter the details of the source of radiation, the X-ray generator,
and the wavelength for each diffraction.

For experiment 1:
<radiation_experiment_1 = " ">   (e.g. 1, 2, ...)
<radiation_source_type_1 = " ">  (e.g. rotating-anode, synchrotron ...)
```

```
<radiation_source_name_1= " ">   (e.g. Rigaku RU200, CHESS Beamline A1 ...)
<radiation_wavelengths_1= " ">   (e.g. 1.502 ...)
<radiation_protocol_1= " ">      (e.g. MAD, SINGLE WAVELENGTH ...)
<radiation_detector_type_1 = " ">     (e.g. CCD, IMAGE PLATE ...)
<radiation_detector_name_1= " ">  (e.g. SIEMENS-NICOLET, RIGAKU RAXIS ...)

For experiment 2:
<radiation_experiment_2 = " ">
<radiation_source_type_2 = " ">
<radiation_source_name_2 = " ">
<radiation_wavelengths_2 = " ">
<radiation_protocol_2= " ">
<radiation_detector_type_2 = " ">
<radiation_detector_name_2= " ">

....(add more if needed)....

================CATEGORY 18:   Collection Temperature====================
Enter the temperature for data collection (in Kelvin)

<collection_temperature_crystal_1 = " ">  (for crystal  1:)
<collection_temperature_crystal_2 = " ">  (for crystal  2:)

....(add more if needed)....


================CATEGORY 19:   Structure Genomics=========================
If it is the structure genomics project, give the information

<SG_project_id = " 1">
<SG_project_name = " ">
<full_name_of_SG_center = " ">
<initial_of_SG_center = " ">

====================================END=====================================
```

12 Appendix (*log_script.inp*)

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        THE LOG_SCRIPT.INP FILE
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                         NOTES AND REMINDER
This script file is used to enter the names of the crystallographic
software used for structure determination and the log, PDB, mmCIF or
text files generated by them.

PLEASE COMPLETE the ENTRY FIELDS according to the type of your experiment
and use the command 'extract -ext log_script.inp' to obtain the completed
structure data ready for validation and deposition.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

                         GUIDELINES FOR USING THIS FILE
  1. Only strings included between the 'lesser than' and 'greater than'
     signs (<.....>) will be parsed for evaluation by the program. Therefore,
     DO NOT write either on the left or right of the 'less than' and 'greater
     than' signs respectively.

  2. All alphanumeric values or strings that you include in the different
     categories should be within double-quotes. Blank spaces or carriage
     returns within a pair of double quotes are ignored by the program.
     DO NOT use double quotes (") within strings that you enter.

  3. Log files used for generating the deposition should be generated from
     the best (usually the last) trial for each crystallographic software.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

~~~~~~~~~~~~~~~~~~~~~~~~~~~START INPUT DATA BELOW~~~~~~~~~~~~~~~~~~~~~~~

===============PART 1: Structure Factor for Final Refinement==============
Enter reflection data file used for final structure refinement

     NOTE:
     * Usually the highest resolution or best data set is used for the
       refinement. Use that structure factor file here.

     *  In some cases, it may not be possible to collect a complete dataset
       from a single crystal. Thus, multiple data sets have to be scaled
       and merged together for refinement. Use the merged reflection file
       here.

     * If the reflection data format is not one of those listed below,
       please use OTHER for the data format, and provide an ASCII file
       that has at least five values [H, K, L, I (or F), sigmaI (or sigmaF)]
       for each reflection and seperate each item by one or more spaces.
       Include the test flags as the sixth column in the file (if available).

     * If the reflection file is in mtz format (e.g. using REFMAC5), convert
       it to mmCIF format using the mtz2various application provided by CCP4.

       Reflection data format:
       CNS|SHELX|TNT|REFMAC5|HKL|SCALEPACK|DTREK|SAINT|SCALA|3DSCALE

<reflection_data_type = "F" >      [enter I (intensity) or F (amplitude)]
<reflection_data_format = "CNS" >
<reflection_data_file_name = " input_data/gere-nat.cv" >

==============PART 2: Structure Factors for Protein Phasing===============
Enter reflection data files used for heavy atom or MAD phasing

     NOTE:
     *  Enter this category if you have more than one complete reflection
       file (e.g. in the case of MAD,SIRAS, MIR). The LOG files generated
```

```
   from data scaling software for all these data sets are also needed.

*  If the scaling program is not one of those listed below
   (HKL|SCALEPACK|DTREK|SAINT|3DSCALE), enter OTHER for the program
   name and provide an ASCII file with five values
   [H, K, L, I (or F), sigmaI (or sigmaF)] for each reflection and
    seperate each item by a space

*  If the same crystal was used for collecting multiple data sets, the
   crystal number will remain '1' as the wavelength numbers change.
   However, if multiple crystals were used, for the data collections,
   the corresponding crystal numbers should be used for each data set.

*  IT IS IMPORTANT THAT THE LOG FILE AND DATA FILE COME FROM THE
   SAME PROGRAM.

<scale_data_type = "I" >          [enter I (intensity) or F (amplitude)]
<scale_program_name = "HKL" >

For data set 1:
<crystal_number  = "1" >
<diffract_number = "1" >
<scale_data_file_name  = " input_data/w1.sca" >
<scale_log_file_name   = " " >

For data set 2:
<crystal_number  = "1" >
<diffract_number = "2" >
<scale_data_file_name  = "input_data/w2.sca  " >
<scale_log_file_name   = " " >

For data set 3:
<crystal_number  = "1" >
<diffract_number = "3" >
<scale_data_file_name  = "input_data/w3.sca " >
<scale_log_file_name   = " " >

==================PART 3: Statistics for Indexing=====================
Enter log file and software name for data indexing

     NOTE:
     *  This is only for the data of final structure refinment.

        Software for indexing is one of the following:
        (HKL|DENZO|DTREK|MOSFLM)

<data_indexing_software = "HKL" >
<data_indexing_LOG_file_name = " " >
<data_indexing_CIF_file_name = " " >  (if mmCIF format)

==================PART 4: Statistics for Data Scaling====================
Enter log file and software name for data scaling

     NOTE:
     *  The log file included here should have scaling statistics of
        the file used for the final structure refinement. If multiple data
        sets were scaled and merged for refinement (as described in Part 1
        above) use the log file generated during merging of the data sets.

        Software for scaling is one of the following:
        (HKL|SCALEPACK|DTREK|SAINT|3DSCALE|SCALA)

<data_scaling_software = "HKL" >
<data_scaling_LOG_file_name = "input_data/sclepack1.log " >
<data_scaling_CIF_file_name = " " >  (if  mmCIF format)

==============PART 5: Statistics for Molecular Replacement===============
Enter log files and software name for molecular replacement

     NOTE:
```

```
        Software is one of the following:
        (CNS|AMORE|MOLREP|EPMR|PHASER)
        The log file should be from the best trial of MR.

<mr_software = " " >
<mr_log_file_LOG_1 = " " >
<mr_log_file_LOG_2 = " " >

=================PART 6: Statistics for Protein Phasing==================
Enter log files and software name for heavy atom phasing

     NOTE:
        The phasing method should be one of (SAD|MAD|SIR|SIRAS|MIR|MIRAS).
        Software is one of the following:
        (CNS|MLPHARE|SOLVE|SHELXS|SHELXD|SNB|BNP|SHARP|PHASES)
        The log file should be from the best trial of phasing.

<phasing_method = "MAD" >        (SAD|MAD|SIR|SIRAS|MIR|MIRAS)
<phasing_software = "CNS" >

<phasing_log_file_LOG_1 = "input_data/mad_sdb.dat " >
<phasing_log_file_PDB_1 = " " >    (in PDB format (heavy atom coordinates))
<phasing_log_file_CIF_1 = " " >    (in mmCIF format)

<phasing_log_file_LOG_2 = "input_data/mad_summary.dat " >
<phasing_log_file_PDB_2 = " " >
<phasing_log_file_CIF_2 = " " >

<phasing_log_file_LOG_3 = "input_data/mad_fp.dat  " >
<phasing_log_file_PDB_3 = " " >
<phasing_log_file_CIF_3 = " " >

... add more if needed ...

===============PART 7: Statistics for Density Modification===============
Enter log files and software name for density modification

     NOTE:
        Software is one of the following:
        (CNS|DM|RESOLVE|SOLOMON|SHELXE)
        The log file should be from the best trial of density modification.

<dm_software = "CNS " >
<dm_log_file_LOG_1 = " input_data/density_modify.dat" >
<dm_log_file_CIF_1 = " " >         (in mmCIF format)

===============PART 8: Statistics for Structure Refinement===============
Enter log files and software name used for final structure refinement

     NOTE:

        Software is one of the following:
        (CNS|REFMAC5|SHELXL|TNT|PROLSQ|NUCLSQ|RESTRAIN)
        The log file should be from the final trial of structure refinement.

<refine_software = "CNS" >

<refine_log_file_PDB_1 = " " >  (coordinate file in PDB format)
<refine_log_file_CIF_1  =  "input_data/deposit_cns.mmcif  "  >    (mmCIF  file  containing  refinement
statistics)
<refine_log_file_LOG_1 = " " >



=======================PART 9: Data Template File========================
Enter file name of the data template file

     NOTE:
        This file 'data_template.text' was generated by using the
        command 'extract -pdb pdb_file' or 'extract -cif cif_file'. It
```

```
        contains the sequences of all unique polymers (protein or nucleic
        acid) present in the structure. It also contains other
        non-electronically captured information. Please complete the
        data template file before running pdb_extract.

<data_template_file = "input_data/data_template.text" >


==========================PART 10: Output Files============================
Enter the output file names

     NOTE:
        If you do not give the output file names, the default names
        pdb_extract_sf.mmcif containing structure factors and
        pdb_extract.mmcif containing coordinates will be assigned
        by the program


<sf_output= " script_example_1_sf.cif" >            (for structure factors)
<statistics_output= "script_example_1.cif " >    (for coordinates and statistics)

===================================END=====================================
```