URO

Jorge Navaza[1]
CNRS–GIF

Carmen Alvarez–Rúa & Javier Borge
Universidad de Oviedo

[1]Navaza, Lepault, Rey, Alvarez-Rúa & Borge (2002). *Acta Cryst.D***25**, 1820-1825.

$\mathcal{URO}$ is a package of programs for fitting atomic models into electron microscopy (EM) density maps.

The algorithms and the implementation are essentially those described for the program **FITING** [1], in the $\mathcal{AMoRe}$ [2] package, but substantially modified to deal with the (non-crystallographic) symmetry and the phases of the 3D EM reconstructions.

Special requirements

The following packages/programs have to be installed:

- Optional:
  * **O** (T. A. Jones, Uppsala)
  * **CCP4** (York)
  * **MAPMAN** (G. Kleywegt, Uppsala)

[1] Castellano *et al.*, 1992. *J. Appl. Cryst.* **25**, 281-284
[2] Navaza, 1994. *Acta Cryst.* A**50**, 157-163

Preliminary settings

- You need the environment variables **URO**, **COMPILE** and **BIN**:
  * **URO** defines the directory where the $\mathcal{URO}$ package is stored.
  * **BIN** defines the subdirectory within **$URO** where the binary libraries of compiled objects and executables are stored.
  * **COMPILE** defines the Fortran compiler which has been used to build the compiled objects. Usually, this information is obtained from the name of the directory containing binary files.
- If these variables are not defined, add them to your login file **.login**, e.g.

  setenv **URO** "/usr/local/packages/uro"

  setenv **BIN** "BIN_linux_g77"

  setenv **COMPILE** "g77 -Wall -O2"
- Define an alias, e.g. **uro**, associated with the script **setup**, and add it to your **.login** file:

  alias **uro** "csh **$URO/setup**"

Starting a job

Create a new working directory and move to it. Invoke the setup from the working directory:

uro

This setup procedure creates subdirectories in which all the required files and scripts necessary to run the $\mathcal{URO}$ package are stored.

Data preparation

- Move to **./d/**
  * Put in this directory the EM reconstruction, in EZD format.
  * Select the region of the EM image that is going to be used for model fitting (the whole reconstruction or a fraction of it). From now on, the selected region will be called the EM box. The EM box can be generated by executing the script **$URO/$BIN/e2e**.

    The EM box has to fulfill some special requirements:

    1. make sure that pixel $(0, 0, 0)$ of the EZD map corresponds to the origin of the operators which define the symmetry of the EM reconstruction. Adjust accordingly the header (ORIGIN) of the EZD file.
    2. the sampling points in the box (keyword EXTENT in the EZD file) have to satisfy the requirements of the FFT:

       2.1 nx, ny and nz must be even numbers;

       2.2 no prime factors higher than 19 are allowed.
  * The EM box must be named **emap.d**. Create a symbolic link between the EM box file and the name required by the program; e.g.

    ln -sf EM_box.ezd emap.d
  * Create two files containing the symmetry operators imposed on the EM reconstruction: **gs.sym** (**O** format) and **sym** ($\mathcal{URO}$ format). Scripts are available to generate these files for several symmetries:

d7.scr for D7 symmetry, i.e., a seven-fold axis along Z and two-folds axes perpendicular to Z; **$URO/$BIN/icosim** for most settings of icosahedral symmetry; **$URO/$BIN/tubes** for helical symmetry.

A script is available which transforms an (**O** format) file into an $\mathcal{URO}$ type symmetry file: **o2u.scr** .

Note: in order to avoid some problems while running **O**, the identity operator should appear in the first position in both files (**gs.sym** and **sym**). Check and modify these files if necessary.

* Define the resolution limits (in Å) for the fitting (last line in the file **data.d**; the other lines are automatically updated by the procedure).
* Put in this directory the PDB files corresponding to the independent search models (any position).
* Insert a FORMAT card at the head of all the files containing model coordinates. An example of this card for standard PDB files can be found in the file **fmt** .
* Data files with model coordinates must be named **xyz{n}.d** (**n** is the model identification number). Create symbolic links between the model files and the names required by the program; e.g.

```
ln -sf VP1.pdb xyz1.d
ln -sf VP7.pdb xyz2.d
```

- Move to **./** (the working directory)

  * The standard use of $\mathcal{URO}$ requires initial positions of the independent molecules to be fitted into the EM box. Each of these molecules corresponds to one of the model coordinates **./d/xyz{n}.d** . They have to be placed at tentative positions —the initial positions for the procedure— with the help of graphics. We need as many files of coordinates as independent molecules are going to be fitted into the EM box. The names of these files must have the extension **.pdb** .
    Note: insert a FORMATcard in all files of coordinates.
  * The correspondence between the model identification number and the molecule name is stored in the file **modlist**. This file has to be created by the user. An example follows:

Assume that the EM box contains three independent molecules, called **A**, **B** and **C**. Assume that the first two molecules (**A** and **B**) correspond to model number **1**, and that the last one corresponds to model number **2**. Then, the working directory must contain the three files of coordinates **A.pdb**, **B.pdb** and **C.pdb**, and the directory **./d** must contain the two files **xyz1.d** and **xyz2.d**.
The file **modlist** is, in this case,

```
#  1  A  B
#  2  C
```

---

or also

```
#  1  A
#  1  B
#  2  C
```

Note: a "#" symbol must appear at the beginning of each line. This symbol is immediately followed by the model identification number and the filenames (without extension) of the associated independent molecules.

Protocol

Programs are executed from the working directory (**./**).

- Calculate the Fourier transform of the EM box. Hereafter, the resulting Fourier coefficients of this transform will be called observed structure factors:

  ./e/emft [map] [scale]

  [map]: name of the file containing the EM box, usually **./d/emap.d**.

  [scale]: this factor is used to scale the dimensions of the EM reconstruction with the model dimensions (usually 1.).

  This command creates the binary file **./f/xudi** of packed and sorted $H, K, L,\ F^{obs}, Phi^{obs}$. The output file from this command containing some statistics is stored as **./o/sort.s**

- Calculate the molecular scattering factors of each of the models:

  ./e/scat [n]

  [n]: model identification number (corresponding to file **./d/xyz{n}.d**).

  This command creates a binary file containing the tabulated molecular scattering factor of model [n]. This file is stored as **./f/tabl[n]** . The output file from this command is stored as **./o/tabl[n].s**

- Each molecule that fits the EM reconstruction is considered as a rigid body and its position in the EM box is given by three angles (α, β, γ) and three translations (x, y, z). The reference for these positional variables (i.e., 0. 0. 0. 0. 0. 0.) corresponds to the model with the center of mass at the origin of the EM box and the principal axes of inertia parallel to the EM box axes. A file has to be created that contains the values of the positional variables corresponding to the initial positions of the independent molecules. If coordinates associated to each independent molecule in its initial position are available, plus the associated **modlist** file, this may be accomplished by executing the following command:

  ./e/c2pv [X]

[X]: name of the generated file containing the initial orientations and positions of the independent molecules. It is stored as **./o/[X].s**

This file has the following format (for the preceding example):

```
fiting: ** URO **
    3       1
 # 1  280.0   90.0    90.0  -0.6500  -0.1500  0.0000  1.0  1.0  1.0
 # 1   77.8   88.5   270.3  -0.6900  -0.0200  0.0900  1.0  1.0  1.0
 # 2  160.0   37.0    50.0   0.3000  -0.3100  0.4500  1.0  1.0  1.0
```

Description

1) Keyword (format A7) = fiting:

2) NBOD NSOL
   NBOD : number of independent molecules.
   NSOL : cabalistic number (always 1).

Then, NBOD cards corresponding to the optimized orientations and translations of the independent molecules.

3) $\#\ n\ \alpha\ \beta\ \gamma\ x\ y\ z\ Cf\ Rf\ Fm$
   n : model identification number. Same as in xyz{n}.d .
   α, β, γ : Euler angles.
   x, y, z : translations (fractionary).
   The other parameters, $Cf\ Rf\ Fm$, are irrelevant at this stage.

- The optimization procedure minimizes the misfit between the EM density in the EM box and the calculated electron density based on the independent molecules plus a certain number of their symmetry mates. Therefore, symmetry operators have to be chosen so that the generated molecules cover the entire density in the EM box. The numbers of the

selected operators (as they appear in the **./d/sym** file) must be included in a file called **symlist**. This file may be manually generated, with the help of a graphical display, or by following this procedure:

* Execute the following command:

  ./e/ctl [X]

  [X]: name of the file (**./o/[X].s**) containing the positional variables on which the generated **./symlist** file is based.

* It is important to check that the molecules generated by application of the symmetry operators listed in the **symlist** file to the independent molecules, cover the entire EM density and that none of them are placed out of the EM box. By running the following command:

  ./go.O

  a macro file **ono** for the graphical system **O** is generated. This file will display the EM density in the EM box and the molecules generated by application of the symmetry operators in **symlist**. Modify this last file if necessary.

- Given the initial (or current) values of the positional variables (file **./o/[X].s**) and the selected symmetry operators (file **symlist**), an input file to the optimization program has to be created. Execute:

  ./e/oic [X] [fit]

  [X] : name of the file (**./o/[X].s**) containing the positional variables to be optimized.

  [fit]: name of the generated file (without extension), input to the optimization program (**FITING**). This file is stored as **./i/[fit].i1** . It presents the following format:

```
fiting +*+*+*+*+*+*+*+*+*
4 1 1 0 1 :printing
10 9 :logical units
* URO **
  264.000   392.000   432.000   90.000   90.000   90.000
x,y,z * stop
1
        95.0       0.0
     400.00    20.00
ncs 6
 -174.76   0.00   0.00   -0.07576   0.41837   0.22995   #18
 -206.21   0.00   0.00   -0.07576   0.41837   0.24838   #19
 -160.78   0.00   0.00   -0.07576   0.41837   0.43266   #29
 -192.23   0.00   0.00   -0.07576   0.41837   0.45109   #30
 -146.80   0.00   0.00   -0.07576   0.41837   0.63539   #40
 -178.25   0.00   0.00   -0.07576   0.41837   0.65382   #41
inertia tensors 2
 * 1   563.56   208.62   208.62   0.00   0.00   0.00
 * 2   123.23    45.82   193.32   0.00   0.00   0.00
3 0 ++++++++++++++++++++
0 1 1 1 1 1 1
1 30 0.0020
 # 1    46.0     2.3     5.5   -0.6645   -0.1233   -0.0517   1.0   1.0   1.0
 # 1   123.8   131.9   235.8   -0.7840   -0.0239    0.0610   1.0   1.0   1.0
 # 2   156.2    36.2    19.0    0.4012   -0.2149    0.4201   1.0   1.0   1.0
```

Description

---

1) Keyword (format A7) = fiting:

2) Printing options.

3) LUN1 LUN2
   Logical units
   LUN1 : input binary file of packed and sorted H, K, L, F^{obs}, phase (**./f/xudi**).
   LUN2 : output of **FITING**.

4) Title (format A80).

5) Cell.

6) Space group symmetry operations (usually x,y,z).

7) NORT
   Code to define an orthogonal reference frame.

8) PERC BADD
   PERC : uses only the PERC % highest F^{obs}.
   BADD : B-factor added to F^{obs}.

9) DMIN, DMAX
   Resolution limits (in Å).

10) ncs NCS
    NCS : number of selected symmetry operators (equal to number of operators in **symlist** file).

11) NCS cards, each one corresponding to a selected symmetry operator in the **symlist** file. The format is similar to that of file **./d/sym**. The symmetry operators have been transformed from the EM image reference frame into the EM box.

12) inertia tensors NINT
NINT : number of models.

13) NINT cards, each one corresponding to the components of the inertia tensor of a model.

14) NBOD, PIVOT
NBOD : number of independent molecules.
PIVOT : term added to the diagonal of the normal matrix in the least-squares procedure (usually 0).

15) $Bf\ \alpha f\ \beta f\ \gamma f\ xf\ yf\ zf$
Refining flags corresponding to the following variables:
Bf : B factor
αf, βf, γf : Euler angles
xf, yf, zf : Translations
These flags must be set to 1 if the variable has to be refined and set to 0 if the corresponding variable should not be optimized.

16) NCYC, NITE, RMSS
NCYC : number of times the NBOD-bodies are alternately refined. If 0, only the last one in the list is refined.
NITE : number of iterations in the least-squares procedure.
RMSS : root-mean-square shift (in Å). Least-squares stops if the rms. correction to positions is less than RMSS.

17) $\#\ n\ \alpha\ \beta\ \gamma\ x\ y\ z$
n : model identification number. Same as in **xyz{n}.d** .
α, β, γ : Euler angles.
x, y, z : translations (fractionary).
The parameters appearing after coordinates z are, at this point, meaningless.

- Run the optimization program **FITING**:

  ./e/fiting [fit] [Y]

[fit] : name of the file (**./i/[fit].i1**) generated by **./e/oic**

[Y] : output file. It is stored as **./o/[Y].s**

This file has the same format as **./o/[X].s** previously described, and contains the optimized values of the positional variables plus the values of different criteria used for assessing the quality of the fit:

```
fiting: ** URO **
     3       1
 # 1   291.2   91.8    91.1  -0.6515  -0.1557  0.0015   1.0   1.0   1.0
 # 1    91.6   88.2   271.0  -0.6901  -0.0259  0.0906   1.0   1.0   1.0
 # 2   166.2   35.2    49.8   0.3012  -0.3189  0.4502  79.1  47.3  19.2
```

Description

---

1) Keyword (format A7) = fiting:

2) NBOD NSOL
   NBOD : number of independent molecules.
   NSOL : cabalistic number (always 1).

Then, NBOD cards corresponding to the optimized orientations and translations of the independent molecules.

3) $\#\ n\ \alpha\ \beta\ \gamma\ x\ y\ z\ Cf\ Rf\ Fm$

n	:	model identification number. Same as in xyz{n}.d.
α, β, γ	:	Euler angles.
x, y, z	:	translations (fractionary).

The following parameters have a meaning only for the last line:

Cf	:	correlation between observed and calculated complex structure factors (x 100).
Rf	:	crystallographic R-factor (x 100).
Fm	:	value of the optimized function (quadratic misfit) (x 100).

- Have a look at the final rms shifts of parameters at the standard output of the **FITING** program ("r-m-s shifts: rotation, translation, total"). If these values are far away from 0., the optimization process should be iterated until convergence. The following commands are required for iteration:

  ./e/oic [Y] [fit]
  ./e/fiting [fit] [Y]
  Both steps may be performed in a single run by executing

  ./e/oic_fiting [Y] [Y]

- Once the process has converged, generate the optimized coordinates of the independent molecules by invoking the following command:

  ./e/pv2c [Y]

  [Y]: output filename (**./o/[Y].s**) generated by **FITING**.

  This command makes a backup of the files containing the initial positions of the independent molecules,
  A.pdb B.pdb, ... → **BUNDLED_INI_#.pdb**
  and writes the optimized coordinates in the original files **A.pdb**, **B.pdb**, ...

- Create a macro file for visualizing the final results with the program **O**:

  ./go.O or **./go_instance.O**

Additional tasks

Creating a mask around the molecules that fit the EM density in the EM box

Once the optimization procedure (as described above) is completed, it is worth computing a mask of the EM reconstruction around the optimized molecules. This will generate a new EM box, which contains contributions of the EM reconstruction only around the molecules which are being fitted. If the optimization procedure is repeated with this new box, the final correlation coefficients will be better, since all the contributions from residual density will be eliminated from the computation of the Fourier transform of the EM density in the EM box. The complete process consists of the following steps:

* Run

  ./e/pv2c [Y]

  [Y] : output from a previous run of **FITING**.

* To create a mask, execute the following command, which requires **CCP4** and **MAPMAN**:

  ./go.MASK [mask.ezd]

  [mask.ezd]: name of the EZD file containing the mask.

* Calculate the Fourier transform of the mask:

  ./e/emft [mask.ezd]

* Create an input file and run the optimization program:

  ./e/oic_fiting [Y] [Z]

Splitting models

Imagine that, in the example given throughout this manual, model 2 consists of two different domains and a motion between them is likely to happen. Therefore, it is worth splitting model 2 into the constitutive domains and see what happens when refining their positions separately.
The working procedure consists of the following steps:

* Move to directory **./d/** .
  Split your model(s) into different PDB files.
  Assign to each fragment a new model identification number. E.g.:

  ln -sf frag1.pdb xyz4.d
  ln -sf frag2.pdb xyz5.d

* Move to the working directory (**./**).
  Calculate the molecular scattering factors of each new model:

  ./e/scat [n]

  [n]: model identification number; in this example, n = 4, 5.

* Edit the command file **splits** and substitute the line that appears between the two "EOF" cards by

  #2 > #4 #5

  which means: substitute model #2 by the constituents models #4 and #5. In the general case, insert as many lines as models are being split.

* The script **splits** creates a new file containing the initial orientations and positions of the fragments. Execute:

  ./splits [Y] [Z]

  [Y]: output file (**./o/[Y].s**) from a previous run of **FITING** (unsplit models).

  [Z]: new file created by splits. It is stored as **./o/[Z].s** (split models)

  In the commented example, if the input **./o/[Y].s** is

```
fiting: ** URO **
    3       1
# 1  291.2  91.8   91.1  -0.6515  -0.1557  0.0015   1.0   1.0   1.0
# 1   91.6  88.2  271.0  -0.6901  -0.0259  0.0906   1.0   1.0   1.0
# 2  166.2  35.2   49.8   0.3012  -0.3189  0.4502  79.1  47.3  19.2
```

---

the resulting output file **./o/[Z].s** will look like this:

```
fiting: ** URO **
    4       1
# 1  291.2    91.8    91.1   -0.6515  -0.1557   0.0015   1.0   1.0   1.0
# 1   91.6    88.2   271.0   -0.6901  -0.0259   0.0906   1.0   1.0   1.0
# 4  136.4    35.2   142.3    0.2032  -0.4159   0.3527  79.1  47.3  19.2
# 5   93.7   110.5    43.6    0.6366  -0.1473   0.2739  79.1  47.3  19.2
```

---

* Run the optimization program:

  ./e/oic_fiting [Z] [Z]

* Once the process has converged, edit the file **modlist** and modify appropriately the list of filenames corresponding to the new output from **FITING**. The order of the independent molecules in the file **./o/[Z].s** has to be preserved in the file **modlist**.
  In the commented example, imagine now that a filename **D.pdb** is to be assigned to model 4 and **E.pdb** to model 5. The file **modlist** has to be modified manually and should look like this:

  ```
  #  1  A
  #  1  B
  #  4  D
  #  5  E
  ```

  ---

* Generate the optimized coordinates of the independent molecules by running the following command:

  ./e/pv2c [Z]

  [Z] : output filename (**./o/[Z].s**) generated by **FITING**.

Estimation of radius of convergence of the optimization procedure / Search for other minima

Run this command and follow instructions:

./ncs_rms [Y]

[Y]: output filename (**./o/[Y].s**) from **FITING**.

This script prompts you to enter a rms shift which should be applied to the refined positions of the independent molecules. A number (determined by the user) of randomly generated shifts are applied to the refined coordinates of all or a subset of the independent molecules (also following the user's requirements). **FITING** is then run automatically for all the generated positions. This procedure gives an idea of the radius of convergence of the optimization program. It is also helpful for locating alternative minima of the function which is being optimized (quadratic misfit). The input file to this script is stored as **./o/rms0.s** and the final results are stored in the file **./o/rms1.s** .

An example of this output file follows:

```
fiting: ** URO ** * r-m-s shift [a] : 50.000 *
    3       1
 # 1  291.2  91.7   91.1  -0.6515  -0.1557  0.0015   0.0   0.0  0.0  1.0
 # 1   91.6  88.2  271.0  -0.6901  -0.0259  0.0906   0.0   0.0  0.0  1.0
 # 2  166.2  35.2   49.8   0.3012  -0.3189  0.4502  79.1  47.3  0.0  1.0
    3       1
 # 1  292.3  92.5   89.1  -0.6415  -0.1347  0.0095   0.0   0.0  1.0  1.0
 # 1   91.5  89.2  272.5  -0.6891  -0.0304  0.0905   0.0   0.0  1.5  1.0
 # 2  166.7  37.4   49.5   0.3013  -0.3199  0.4602  79.0  47.4  0.8  1.0
    3       1
 # 1  291.4  91.8   91.4  -0.6517  -0.1567  0.0014   0.0   0.0  1.3  1.0
 # 1   91.5  89.0  272.1  -0.6902  -0.0300  0.0906   0.0   0.0  0.4  2.0
 # 2  164.2  36.2   49.6   0.3012  -0.3198  0.4505  78.9  48.3  1.6  1.0
```

Description

---

1) Keyword (format A7) = fiting: ; * r-m-s shift [CODE] : RMS *

CODE : type of transformation applied to the input positions:
R → rotation
T → translation
A → rotation + translation

RMS : applied rms shift

Sections 2) and 3) are repeated in the output file as many times as random trials have been specified by the user.

2) NBOD NSOL
NBOD : number of independent molecules.
NSOL : cabalistic number (always 1).

Then, NBOD cards corresponding to the optimized orientations and translations of the independent molecules.

3) # n α β γ x y z Cf Rf $rmss$ $nums$
n : model identification number. Same as in xyz{n}.d .
α, β, γ : Euler angles.
x, y, z : translations (fractionary).
$rmss$: closest rms difference between the current solution and the rest of random trials in the file **rms1.s** .
$nums$: number of random trial closest to this solution.
For the following parameters only the last line is relevant.
Cf : correlation between observed and calculated complex structure factors (x 100).
Rf : crystallographic R-factor (x 100).
