$\mathcal{AM}o\mathcal{R}\varepsilon$
a package for
$\mathcal{A}$-utomatic $\mathcal{M}o$-lecular $\mathcal{R}\varepsilon$-placement

Tutorial

Jorge Navaza[1]
C.N.R.S. France

[1] Acta Crystallographica A50, 157-163. (1994) "$\mathcal{AM}o\mathcal{R}\varepsilon$: an Automated Package for Molecular Replacement".

Multi sunt vocati, pauci vero electi.

AMoRe is a suite of programs aimed at locating model electron densities within the crystal cell.

Contents

The package contains:

factor.d	=	ascii file of atomic form-factors.
symop.d	=	ascii file of space-group symmetry operations.
maitre.for	=	program and subroutines to be modified, if necessary, by the user.
setup	=	script to initiate a molecular replacement problem.
tutorial.pdf	=	pdf file with tutorial.
writeup.pdf	=	pdf file with manual.
login	=	example of variables to include into the .login file.

BIN_linRH_g77 BIN_SGI5.3_f77 BIN_···	=	platform and configuration dependent sub-directories containing the library of compiled subroutines and executables	esclave.a amore.exe cording.exe splits.exe getsym

The user must define the environment variable **AMORE** equal to the pathname of the directory where the AMoRe package is, and the environment variable **BIN** equal to the name of the sub-directory (within $AMORE) where the appropriate binary files (archive and executables) are. These binary files are machine/OS-version/compiler dependent; each configuration has its own particular sub-directory.

By default, the programs use the executables in $AMORE/$BIN, but the user has the possibility of producing its own local executables. For this, he needs to define the appropriate environment variable **COMPILE** (compiler options as used to generate esclave.a in the corresponding sub-directory $BIN),

Example of .login file (for csh). See also the ’login’ file.

```
setenv AMORE  ”/{pathname-of-AMoRe}”

switch (‘hostname -s‘)

case ’Tango’:
# PC under OS Linux Red-Hat and compiler g77
setenv BIN ”BIN_linRH_g77”
setenv COMPILE ”g77 -Wall -O2”
breaksw

case ’Milonga’:
# Silicon Graphics Indigo under OS 5.3 and compiler f77
setenv BIN ”BIN_SGI5.3_f77”
setenv COMPILE ”f77”
breaksw

...

endsw
```

Programs

Preliminary programs to cast input data into a suitable representation:

SORTING packs and sorts H,K,L,F^{obs}. The packing is cell and space-group dependent.

TABLING calculates the array of molecular scattering factors corresponding to the model coordinates (or electron densities). Puts the model in a small model-box. First translates the coordinates so that the center of mass – CoM – is at the origin, and rotates the coordinates so that the model's principal axes of inertia are parallel to the model-box axes. The model-box is put in a big cell in order to sample the model transform finely, to allow structure factors and gradients of the rotating model to be accurately interpolated from the array. The input may be an electron density map.

Main molecular replacement programs:

ROTING calculates spherical-harmonics coefficients associated to crystal and model Patterson functions and computes several rotation functions.

TRAING computes n-body fast translation functions. The output is, for each orientation, the correlation coefficients and R-factors of the top peaks of fast translation functions.

FITING performs least-squares fast rigid-body refinement.

Auxiliary programs:

PATTING calculates the Patterson function.

SELFING calculates the Self-Rotation function.

FUNKING reads fast rotation function output and calculates several figures of merits. It is automatically executed after ROTING.

JOB creates a starting set of inputs and a script with a tentative protocol to solve the molecular replacement problem.

OIC prepares inputs to main molecular replacement programs by selecting and combining intermediate results.

MR2IC works out the final rotations and translations to apply to the initial models.

CORDING produces coordinates corresponding to the best solution.

SPLITS produces positional variables corresponding to fragments of a molecule.

Automation

The package may be used at three different levels of automation:

- LEVEL-3: a starting set of inputs and a script with a tentative protocol to solve the molecular replacement problem, are created by program JOB. The input is general purpose control parameters (e.g. translation function options, number of peaks to select).

- LEVEL-2: inputs to the main molecular replacement programs are created by program OIC, which selects and combines available potential solutions. Information produced in previous steps may be used in order to create efficient inputs for OIC (e.g. cutoff values to skip potential solutions). The procedures oic_roting, oic_traing and oic_fiting are the concatenations of OIC with the main molecular replacement programs.
- LEVEL-1: inputs to the main programs are prepared by the user. In general, they need information produced in previous steps (e.g. positions to be refined).

Starting a Molecular Replacement Problem

To start a molecular replacement problem, it is recommended to move to an empty working-directory and execute

- csh $AMORE/setup

An alias may be defined to perform it. The setup procedure creates sub-directories, and puts files into them. The initial content of the working-directory and sub-directories is:

./	aide-memoire	=	succinct notice.
	./d/ ./e/ ./f/ ./i/ ./o/	=	sub-directories.
./d/	data.d	=	example of main AMoRε input.
	hkl.example xyz1.example	=	examples of diffraction data and coordinates files needed by AMoRε .
./e/	maitre.for	=	program and subroutines to fit data and memory requirements (copied from $AMORE).
	makeupd	=	script to make a new (local) executable, by compiling maitre.for and linking with $AMORE/$BIN/esclave.a .
	cording entorno fiting funking job mr2ic oic oic_fiting oic_roting oic_traing patting selfing sorting splits tabling traing	=	program commands.
./f/	empty	;	it will contain binary files created by programs.
./i/	dato.i3	=	example of LEVEL-3 input to run AMoRε automatically.
		;	it will contain inputs to programs; names with prescribed syntax (e.g. dato.i3 , sort.i1 , . . .).
./o/	empty	;	it will contain outputs of programs, named {$}.s .

Move to:

- ./d/

  adapt the data.d file. The symmetry cards must be lower case, finishing ’* end’ or ’ * stop’; no more than 80 characters per line; first equivalent position must be ’x,y,z’. They may be generated by executing

  ```
  $AMORE/getsym {space-group-name or space-group-number}
  ```

  Files with diffraction data and model coordinates must be named hkl.d (as hkl.example) and xyz{#}.d for model number{#} (as xyz1.example), respectively. Insert the FORMAT card in upper case, in hkl.d and xyz{#}.d files.

  When model electron densities are used instead of model coordinates, create the map{#}.d files, as explained in the write-up.

  It is not necessary to rename files; just create symbolic links:

  ```
  ln -sf FAB_LYS.HKL hkl.d
  ln -sf FAB.PDB xyz1.d
  ln -sf LYS.PDB xyz2.d
  ```

- ./i/

  Adapt dato.i3 , if necessary.

Programs are executed from the working-directory. Execute

- csh ./e/job dato

JOB creates default input cards (in ./i/) and a script (./job) with a tentative protocol to solve the molecular replacement problem. The script ./job is in fact a series of procedures which may be executed separately, as illustrated in the Example below.

To run *AMoRe* automatically, submit

- csh ./job >& log & (or csh ./job)

To generate the output coordinates corresponding to the best solution in the fiting-output file ./o/NAME.s , execute

- ./e/cording NAME

Example

The data correspond to a crystal with two molecules of the complex Fab-Lysozyme in the asymmetric unit, and two search models (xyz1.d = Fab, xyz2.d = Lysozyme). The input needed by AMoRe is:

Input:

file ./d/hkl.d

The FORMAT card should allow reading [H, K, L, F^{obs}] – four items. All header cards are ignored, till the FORMAT card (format A5,*) is found. If phases are available (to use in 'p-t-f' phased translation), insert PHASE card (format A5) after FORMAT, which should now allow reading five items.

```
remark: this is a fab + lysozyme complex. it may have two molecules.
remark: therefore it may have two fab and two lysozyme.
remark: coordinates sent fab d1.3 hel, residues 601 up are lysozyme.
remark: the following format may not correspond to the data.
FORMAT (3I4,F12.2)
   0   0   6   894.10   38.40
   0   0   8   507.50   15.70
   0   0  10   116.10    9.70
...
```

file ./d/xyz1.d

The FORMAT card should allow reading [ATOM-type, X, Y, Z, B-factor] – 5 items. If fractionnary coordinates, insert CELL card (format A4) after FORMAT (see subroutine lecatc in file ./e/maitre.for).

```
  remark: FAB FRAGMENT (ANTI-LYSOZYME ANTIBODY D1.3)
  remark: the following format (typical of PDB) may not correspond to
  the data.
  FORMAT (12X,A4,14X,3F8.3,6X,F6.2)
  ATOM   1   N     ASP   1   4.060   7.307   5.186   1.00   51.58
  ATOM   2   CA    ASP   1   4.042   7.776   6.553   1.00   48.05
 ...
```

file ./d/xyz2.d

```
  remark: LYSOZYME
  remark: the following format (typical of PDB) may not correspond to
  the data.
  FORMAT (12X,A4,14X,3F8.3,6X,F6.2)
  ATOM   3314   N     LYS   1   0.829   -12.809   39.408   1.00   35.17
  ATOM   3315   CA    LYS   1   0.556   -13.062   38.012   1.00   33.11
 ...
```

Note that atom types are read with format A4. However, only the first two characters (Mendeleyev notation) are used to get the atomic scattering factors from file $AMORE/factor.d (see subroutine lecatc and lgauss in ./e/maitre.for). This may need shifting of some names. For example, Calcium, Carbon, Zinc and Oxygen should appear as:

```
  ATOM   311   Ca   EXA   1   0.545   -10.000   38.000   1.00   30.01
  ATOM   312    C   EXA   1   0.556   -13.062   38.012   1.00   33.11
  ATOM   313   Zn   EXA   1   0.829   -12.809   39.408   1.00   35.17
  ATOM   314    O   EXA   1   0.655   -11.030   39.020   1.00   36.20
```

file ./d/data.d

```
* D44HEL **
99.7   167.3   84.7   90.   90.   90.
x,y,z * 1/2+x,1/2-y,-z * 1/2-x,-y,1/2+z * -x,1/2+y,1/2-z * end
0
95.   0.
15.   3.5
2   2   2
```

Description

1) Title (format A80).

2) Cell.

3) Symmetry operations (lower case), finishing '* end'; first equivalent position must be 'x,y,z'; no more than 80 characters per line. Centering translation may be explicitly introduced (see file $AMORE/symop.d); for example, the following are both licit inputs (though the latter is recommended) for space group C2:
x,y,z * -x,y,-z * x+1/2,y+1/2,z * -x+1/2,y+1/2,-z * end
or
x,y,z * -x,y,-z * 1/2,1/2,0 * end
These cards may be generated by executing
$AMORE/getsym C2

4) NORT
Code to define an orthogonal reference frame.

5) PERC BADD

PERC	=	uses only the PERC % highest F^{obs} within the selected resolution range.
BADD	=	B-factor added to F^{obs} (e.g. -5 to sharpen data).

6) DMAX DMIN
Resolution limits used for the molecular replacement problem (in Å).

7) NTYP MOL1 ... MOLn

NTYP	=	number of different models.
MOL{#}	=	number of molecules of model-type {#} in the a.u.; {#} = 1,NTYP.

file ./i/dato.i3

```
job +*+*+*+*+*+*+*+*+*
xyz
1.   2   10   0.5   2.5
c-o   50   0.3   30
p-t   10   0.5   30
10   20
20.
```

Description

1) Keyword (format A4) = ’job ’.

2) AKEY (format A5)
Keyword defining mode: if ’xyz ’ reads coordinates; if ’map ’ reads electron density map.

3) RATE LMINs LMINf CUTR STEP
RATE = defines the integration radius as RATE × Molrad, where Molrad is radius of the smallest sphere, with origin at CoM, containing the whole molecule.
LMINs
LMINf = several rotation functions are calculated, where the spherical-harmonics expansions begin with LMINs $\leq \ell \leq$ LMINf; the ℓ-expansion controls the angular resolution.
CUTR = cutoff in rotation function output; first selects all peaks above CUTR × maximum-peak-height.
STEP = step size for ϕ, θ and ψ (in degrees).

4) TKEY NUMR CUTT NPIC (format A5,*)
TKEY =
’c-o’ computes centered-overlap;
’p-t’ computes phased-translation (’p-t-f’ when phases are available);
’h-l’ computes Harada-Lifchitz translation function;
’c-c’ computes correlation-coefficient.
NUMR = selects up to a maximum of NUMR orientations for input to one-body translations.
CUTT = cutoff in fast translation function output; first selects all peaks above CUTT × maximum-peak-height of ’c-o’, ’p-t’, ’h-l’ or ’c-c’ fast translation function.
NPIC = number of peaks to output of translation function. The program computes correlations and R-factors.

5) Same as previous card, but for n-body translations (it must be present, even if not used).

6) NUMT NITE
NUMT = selects up to a maximum of NUMT positions to refine.
NITE = number of iterations in the least-squares procedure.

7) CUTD
Cutoff to eliminate positions with CoM-CoM distance less than CUTD Å.

For the above data, JOB proposes the following protocol (for a description of the different programs, see the write-up):

file ./job

```
# amore

./e/sorting
   set m=1
   while (${m} <= 2)
./e/tabling ${m}
   set m=‘expr ${m} + 1‘
   end
   set m=1
   while (${m} <= 2)
./e/oic_roting oicrd ${m} o${m}r
   set k=‘expr ${m} + 1‘
   end
cat ./o/o*r.s > ./o/or1.s
./e/oic_traing oicto or1 ot1
./e/oic_fiting oicfd ot1 of1
   set k=1
   while (${k} < 4)
   set n=‘expr ${k} + 1‘
./e/oic_traing oictn ot${k} of${k} ot${n}
./e/oic_fiting oicfd ot${n} of${n}
   set k=‘expr ${k} + 1‘
   end
```

Output:

In the case of normal termination, directory ./o/ will then contain the following ascii files:

sort.s	=	statistics of the input reflections.
tabl1.s	=	information about the reference position of model 1.
tabl2.s	=	information about the reference position of model 2.
o1r.s	=	cross-rotation peaks for model 1.
o2r.s	=	cross-rotation peaks for model 2.
or1.s	=	cross-rotation peaks for models 1 and 2.
ot1.s	=	one-body translation peaks for models 1 and 2.
of1.s	=	fast rigid-body refinement of selected solutions in ot1.s .
ot2.s	=	two-body translation peaks for models 1 and 2 (with best solution in ot1.s kept fixed).
of2.s	=	fast rigid-body refinement of selected solutions in ot2.s .
ot3.s	=	three-body translation peaks for models 1 and/or 2 (with best solution in ot2.s kept fixed).
of3.s	=	fast rigid-body refinement of selected solutions in ot3.s .
ot4.s	=	four-body translation peaks for models 1 and/or 2 (with best solution in ot3.s kept fixed).
of4.s	=	fast rigid-body refinement of selected solutions in ot4.s .

The outputs of ROTING, TRAING and FITING follow a same pattern; the most general case is the n-body translation output:

file ./o/ot4.s = 4-body translation function (3 bodies kept fixed).

```
traing: ** D44HEL **
    4     10
  > 1  103.0  26.2   12.7  0.0754  0.3067  0.4549  24.8  50.4  26.6  53.90
  > 1  135.5  43.8  319.1  0.4601  0.2047  0.0151  33.0  48.0  37.6  41.52
  > 2  107.2  70.6   15.8  0.4264  0.4266  0.6109  39.2  46.1  43.6  42.13
  # 2  107.2  70.6   15.8  0.2885  0.0210  0.8054  37.0  46.7  41.9   9.30
  # 2  107.2  70.6   15.8  0.3529  0.6612  0.9675  36.8  46.7  40.5   8.72
  # 2  107.2  70.6   15.8  0.6897  0.7021  0.8633  36.7  46.6  40.9   9.13
  # 2  107.2  70.6   15.8  0.5238  0.3561  0.7624  36.6  46.6  40.7  10.00
  # 2  107.2  70.6   15.8  0.4997  0.3047  0.9067  36.6  46.5  40.1   9.73
  # 2  107.2  70.6   15.8  0.1544  0.8305  0.0475  36.5  46.7  40.6   8.90
  # 2  107.2  70.6   15.8  0.5273  0.1935  0.9461  36.5  46.9  41.1   8.35
  # 2  107.2  70.6   15.8  0.4019  0.4385  0.7297  36.5  46.8  40.7   8.19
  # 2  107.2  70.6   15.8  0.4901  0.3063  0.5681  36.4  46.6  41.7   9.50
  # 2  107.2  70.6   15.8  0.4397  0.8707  0.6853  36.4  46.8  41.7   9.72
    4      1
  > 1  103.0  26.2   12.7  0.0754  0.3067  0.4549  24.8  50.4  26.6  53.90
  > 1  135.5  43.8  319.1  0.4601  0.2047  0.0151  33.0  48.0  37.6  41.52
  > 2  107.2  70.6   15.8  0.4264  0.4266  0.6109  39.2  46.1  43.6  42.13
  # 2  105.6  61.1   48.4  0.8012  0.5030  0.1681  42.1  44.9  47.3   9.80
    4     10
...
```

Description

1) Keyword (format A7) = 'traing:'.

2) NBOD NPIC

NBOD	=	number of molecules (n-body).
NPIC	=	number of translations for the given orientation.

Then NBOD-1 cards corresponding to the fixed positions:

3) $> \ \mu\ \phi\ \theta\ \psi\ x\ y\ z\ C_f\ R_f\ C_i\ D_m$

The last four items (descriptors) recall the genesis of the fixed position.

μ	=	model identification number; also logical-unit identifier for molecular scattering factors; usually, logical-unit = $\mu + 10$.
ϕ, θ, ψ	=	Euler angles.
x, y, z	=	translations (fractionnary).
C_f	=	correlation of amplitudes (× 100).
R_f	=	crystallographic R-factor (× 100).
C_i	=	correlation of intensities (× 100).
D_m	=	minimal CoM-CoM distance with current symmetry related and preceding positions.

Then NPIC cards corresponding to the orientation that was translated:

4) $\#\ \mu\ \phi\ \theta\ \psi\ x\ y\ z\ C_f\ R_f\ C_i\ T_f$

The last three descriptors correspond to the whole configuration
(i.e. fixed positions plus translated orientation).

μ	=	model identification number; also logical-unit identifier for molecular scattering factors; usually, logical-unit = $\mu + 10$.
ϕ, θ, ψ	=	Euler angles.
x, y, z	=	translations (fractionnary).
C_f	=	correlation of amplitudes (× 100).
R_f	=	crystallographic R-factor (× 100).
C_i	=	correlation of intensities (× 100).
T_f	=	fast-translation-function value (in σ units excepting for c-c).

Repeat 2) to 4) for other positions.

The meaning of the descriptors in the output of ROTING is however different:

file ./o/o1r.s = cross-rotation function.

```
roting:  ** D44HEL **
    1       89
  # 2    15.1   71.5   236.4   0.0000   0.0000   0.0000    9.2   54.2   15.5   11.5
  # 2   107.2   70.6    15.8   0.0000   0.0000   0.0000   10.2   54.1   18.0   10.9
  # 2    73.0   59.1    86.8   0.0000   0.0000   0.0000   10.0   54.1   19.3   10.2
  # 2    25.5   82.8   242.3   0.0000   0.0000   0.0000    8.8   54.1   14.0   10.0
  # 2   174.5   12.1    30.5   0.0000   0.0000   0.0000    8.8   54.4   13.3    9.9
. . .
```

Description

1) Keyword (format A7) = 'roting:'.

2) NBOD NPIC
   NBOD = number of molecules (always 1).
   NPIC = number of orientations.

Then NPIC cards:

3) $\# \; \mu \; \phi \; \theta \; \psi \; x \; y \; z \; C_f \; R_f \; C_i \; C_p$
   μ = model identification number; also logical-unit identifier for molecular scattering factors; usually, logical-unit = $\mu + 10$.
   ϕ, θ, ψ = Euler angles.
   x, y, z = translations (fractionnary); set to zero.
   C_f = correlation of amplitudes, in P1 (× 100).
   R_f = crystallographic R-factor (× 100), in P1.
   C_i = correlation of intensities (× 100). It includes all symmetry related orientations, but no intermolecular contribution.
   C_p = truncated Patterson correlation (× 100).

Output coordinates

The coordinates corresponding to the best solution in ./o/of4.s are generated by executing

- ./e/cording of4

By default, the molecules will be put so as to produce the closest pack, starting from the molecule nearest to $(0.5, 0.5, 0.5)$, taking into account Cheshire and space group symmetry. The output of CORDING is the pdb file 'sol1' which contains all the independent molecules. By executing it, i.e. 'csh sol1', the pdb files of the independent molecules are generated.

NCS detection

The Patterson function and the Self-Rotation function can be calculated by executing

- ./e/patting

  and

- ./e/selfing {radius-of-comparison} {output}

respectively. The output of PATTING is ./o/patt.s and that of SELFING is ./o/output.s . {radius-of-comparison} is the radius of the spherical volume of integration, in Å.

Troubleshooting

It is recommended to execute the programs SORTING and TABLING – the interface with the user – separately,

```
csh ./e/sorting
csh ./e/tabling 1
csh ./e/tabling {#}
```

Message errors appear in the log file or standard output.

- **Missing files:** The 'INPUT ERROR' means that a file-name passed as argument does not exist, or that the file does not match the option of the program called.
- **Dimension:** Errors give, if possible, explicit messages (e.g. "set mi >> . . . "). Otherwise they just indicate the problem (e.g. "insufficient memory for . . . ").

  In all cases, fix dimensions in file ./e/maitre.for (FORTRAN code). This file contains the main calling program (amore) with a "parameter" card to define dimensions. Then

  ```
  csh ./e/makeupd
  ```

  it creates a new (local) executable (./e/amore.exe).
- **Format:** The FORMAT card of diffraction data and coordinates should be carefully checked (see Example above). Also, it is worth displaying the input model on a graphics system to check that the model does not contain atoms in extravagant positions. Occupation numbers are not used.
- **Searches with electron densities:** The overall strategy of automation is the fast – and reasonably accurate – computation of structure factors, which is possible if the TABLING program is correctly used (see the write-up). The user should be able to produce an electron density within a rectangular box of any desired dimension, at any desired resolution (within 15 % shift from theoretical requirements).