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

Writeup

Jorge Navaza [1] [2]
C.N.R.S. France

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

[2] Acta Cryst. D57, 1367-1372. (2001) "Implementation of Molecular Replacement in $\mathcal{AM}o\mathcal{R}\varepsilon$" by Jorge Navaza.

SETUP

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

- csh ${AMORE}/setup

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 AMoRe input.
	hkl.example xyz1.example	=	examples of diffraction data and coordinates files needed by AMoRe .
./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 roting 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 AMoRe 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 .

PROGRAMS

JOB

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

Calling command: ./e/job dato

Input:	**./i/dato.i3**	=	input described below.
	./d/data.d	=	main AMoRe input described below.
Output:	**./job**	=	script with protocol for the actual molecular replacement problem.
	./i/sort.i1	=	input for SORTING.
	./i/patt.i1	=	input for PATTING.
	./i/tabl.i1	=	input for TABLING.
	./i/funk.i1	=	input for FUNKING.
	./i/oicrd.i2	=	input for OIC (ROTING mode).
	./i/oicto.i2	=	input for OIC (TRAING mode, one-body).
	./i/oictn.i2	=	input for OIC (TRAING mode, n-body).
	./i/oicfd.i2	=	input for OIC (FITING mode).

Input: 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'.

4) NORT
   Code to define an orthogonal reference frame.

5) PERC BADD
   PERC = uses only the PERC % highest F^{obs}.
   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.

Input: 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:
’xyz ’ reads coordinates;
’map ’ reads electron density map.

3) RATE LMINs LMINf CUTR STEP

RATE = defines the integration radius as RATE × Molrad, where Molrad = 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.

Example

The data.d file corresponds to a crystal with two molecules of the complex Fab-Lysozyme in the asymmetric unit, and two search models (Fab = 1 , Lysozyme = 2). JOB proposes the following protocol:

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 m=‘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${n} ot${n}
   set k=‘expr ${k} + 1‘
   end
```

Note that all the one-body rotation outputs (./o/o{#}r.s) have been concatenated into the single file ./o/or1.s . In this particular example we expect the Fab positions to be determined more easily than the Lysozymes. Thus, we may execute the programs in a different order:

```
# 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 m=‘expr ${m} + 1‘
   end
./e/oic_traing oicto o1r ot1
./e/oic_fiting oicfd ot1 of1
./e/oic_traing oictn ot1 of1 ot2
./e/oic_fiting oicfd ot2 of2
./e/oic_traing oictn o2r of2 ot3
./e/oic_fiting oicfd ot3 of3
./e/oic_traing oictn ot3 of3 ot4
./e/oic_fiting oicfd ot4 of4
```

SORTING

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

Calling command: ./e/sorting

Input:	**./i/sort.i1**	=	input described below.
	./d/hkl.d	=	ascii file of H,K,L,F^{obs}.
Output:	./f/xudi	=	binary file of packed and sorted H,K,L,F^{obs}.
	./o/sort.s	=	SORTING output.

Input: file ./i/sort.i1

```
sorting +*+*+*+*+*+*+*+*+*+*+*
1   10   9   :logical units
* 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
15.00   3.50
1.00
```

Description

1) Keyword (format A7) = 'sorting'.

2) LUN1 LUN2 LUN3

   Logical units (see script ./e/sorting):

   LUN1 = input ascii file of H,K,L,F^{obs} (./d/hkl.d).

   LUN2 = output binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).

   LUN3 = SORTING output (./o/sort.s).

3) Title (format A80).

4) Cell.

5) Symmetry operations (lower case), finishing '* end'.

6) DMAX DMIN

   Resolution limits. Performs statistics for data within this resolution range.

7) SCAL

   Dividing scale factor.

PATTING

PATTING calculates the Patterson function using the expression

$$P(\mathbf{r}) = \frac{1}{V} \sum_{\mathbf{H}} |F_{\mathbf{H}}|^2 \; e^{-2\pi i \mathbf{Hr}}$$

Calling command: ./e/patting

Input:	**./i/patt.i1**	=	input described below.
	./f/xudi	=	binary file of packed and sorted H,K,L,F^{obs}.
Output:	**./o/patt.s**	=	PATTING output.

To recover the Patterson function map, erase in script ./e/patting the symbol “#” in line “#mv $AMOREF/fort.91 patmap”. ./patmap is a binary file with several registers:

```
real patmap(nx,ny,nz)
write(file) nx,ny,nz,a,b,c,alpha,beta,gamma
 do iz=1,nz
write(file) ((patmap(ix,iy,iz),ix=1,nx),iy=1,ny)
 enddo
```

where:

nx,ny,nz	=	number of sampling points in each direction.
a,b,c,alpha,beta,gamma	=	cell parameters (in Å and degrees).

Input: file ./i/patt.i1

```
patting +*+*+*+*+*+*+*+*+*+*+*+*+*+*
2   1   1   :printing
10   9   :logical units
* 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
90.   0.
15.00   3.50   1.00
0.10   100   1.e-4
```

Description

1) Keyword (format A7) = 'patting'.

2) Printing options.

3) LUN1 LUN2

   Logical units (see script ./e/patting):

   LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
   LUN2 = PATTING output (./o/patt.s).

4) Title (format A80).

5) Cell.

6) Symmetry operations (lower case), finishing '* end'.

7) PERC BADD

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

8) DMAX DMIN SHAR

   DMAX,DMIN = data resolution limits.
   SHAR = Shannon rate (the greater SHAR, the finer the mesh). Defines the crystal cell sampling as
   NX ≈ 2 × a / (DMIN/SHAR)
   NY ≈ 2 × b / (DMIN/SHAR)
   NZ ≈ 2 × c / (DMIN/SHAR)

9) CUTP NPIC DELT

   CUTP = cutoff in Patterson function; selects all peaks above CUTP × maximum-peak-height.
   NPIC = maximum number of peaks to output of Patterson function.
   DELT = used in peak-search to avoid spurious peaks.

TABLING

TABLING calculates the array of molecular scattering factors corresponding to the model coordinates [1]. Puts the model in a small model-box: first translates the coordinates so that the center of mass 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 Fourier transform finely, to allow structure factors and gradients of the rotating model to be accurately interpolated.

The contribution of search model m to the calculated structure factors is expressed in terms of the model scattering factors, f_m , calculated by TABLING, by

$$F_{\mathbf{H}}^{cal\,(m)}(\mathbf{R},\mathbf{T}) = \sum_{g=1}^{G} \; f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}\mathbf{O}_m) \; e^{2\pi i\mathbf{H}\left(\mathbf{M}_g\mathbf{T}+\mathbf{t}_g\right)} \; .$$

$\mathbf{R}$ and $\mathbf{T}$ are the orientation and translation that specify the position of the search model. $\mathbf{O}_m$ and $\mathbf{D}$ denote the orthogonalizing and deorthogonalizing matrices, respectively. $\mathbf{M}_g$, $\mathbf{t}_g$ denote the transformation associated to the g^{th} space-group symmetry operation.

Calling command: ./e/tabling {#}

Input:	**./i/tabl.i1**	=	input described below.
	$AMORE/factor.d	=	ascii file of atomic form-factors.
	./d/xyz{#}.d	=	ascii file of coordinates.
	./d/map{#}.d	=	binary file of electron density.
Output:	./f/tabl**{#}**	=	binary file of molecular scattering factors.
	./o/tabl{#}.s	=	TABLING output.

[1] Acta Cryst. A58, 568-573. (2002) "On the Computation of Structure Factors by FFT Techniques" by Jorge Navaza.

Input: file ./i/tabl.i1

```
tabling +*+*+*+*+*+*+*+*+*+*+*
3   1   0   1   :printing
1   2   10   9   :logical units
molecular scattering factors
xyz
0   0.0
3.50
3.5   2.00   1   60.5
```

Description

1) Keyword (format A7) = ’tabling’.

2) Printing options.

3) LUN1 LUN2 LUN3 LUN4
Logical units (see script ./e/tabling):
LUN1 = input ascii file of coordinates – ./d/xyz{#}.d
LUN2 = input binary file of electron density – ./d/map{#}.d
LUN3 = output binary file of molecular scattering factors – ./f/tabl{#}
LUN4 = TABLING output – ./o/tabl{#}.s

4) Title (format A80).
The procedure uses “sed” to add “model {#} -” to the title.

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

6) BFLG BREP
B-factor flags:
BFLG = 1, replace all atomic B-factors by BREP.
BREP = to set all atomic B-factors to BREP.

7) DMIN
Resolution limit of generated Fourier coefficients. In fact program generates past this point to allow for interpolationat the frontier.

8) SBOX SHAR TFLG BADD
SBOX = Defines the model-cell as SBOX × model-box.
SHAR = Shannon rate (the greater SHAR, the finer the mesh).
Defines the model-cell sampling as
NX ≈ 2 $\times \tilde{a}$ / (DMIN/SHAR)
NY ≈ 2 $\times \tilde{b}$ / (DMIN/SHAR)
NZ ≈ 2 $\times \tilde{c}$ / (DMIN/SHAR)
where $\tilde{a}$,$\tilde{b}$,$\tilde{c}$ are the model-cell parameters.
TFLG = 0, use optimal values for SHAR and BADD.
BADD = “B-factor” to smear the atomic electron density. It is added in direct space (when reading atomic coordinates) and subtracted in reciprocal space (in all cases).

Maps

If the search model is an electron density (e.g. a low resolution envelop), TABLING translates it as previously described, but does not rotate it. The input must be a binary file – ./d/map{#}.d – with several registers:

```
real rho(mx,my,mz)
write(file) mx,my,mz,xlw,ylw,zlw,xup,yup,zup,alpha,beta,gamma
 do iz=1,mz
write(file) ((rho(ix,iy,iz),ix=1,mx),iy=1,my)
 enddo
```

where:

mx,my,mz	=	number of sampling points in each direction.
xlw,ylw,zlw,xup,yup,zup	=	lower and upper limits (in Å) of the model-box.
alpha,beta,gamma	=	model-box angles.

The model-box edges are thus

$$\begin{array}{lcl} \tilde{a} & = & \text{xup-xlw} \\ \tilde{b} & = & \text{yup-ylw} \\ \tilde{c} & = & \text{zup-zlw} \end{array}$$

This should be, as far as possible, the smallest box containing the model density. The sampling must be sensibly equal to

$$\begin{array}{lcl} \text{mx} & \approx & 2 \times \tilde{a} \; / \; (\text{DMIN/SHAR}) \\ \text{my} & \approx & 2 \times \tilde{b} \; / \; (\text{DMIN/SHAR}) \\ \text{mz} & \approx & 2 \times \tilde{c} \; / \; (\text{DMIN/SHAR}) \end{array}$$

Fourier Coefficients

It is also possible to provide the output of TABLING without executing the program. It must be a binary file – ./f/tabl{#} – with two registers:

```
complex fto(-1:hsup,-ksup:ksup,-lsup:lsup)
write(file) a,b,c,alpha,beta,gamma,nort,hsup,ksup,lsup,sqhsup
write(file) fto
```

where:

a,b,c,alpha,beta,gamma	=	model-cell.
nort	=	orthogonalising code.
hsup,ksup,lsup	=	maximum indices of the array; the (h,k,l) indices run, respectively, within the limits (-1:hsup), (-ksup:ksup), (-lsup:lsup)
sqhsup	=	1/DMIN^2.
fto	=	array of complex molecular scattering factors.

In this case, it may be useful to provide also the ./o/tabl{#}.s file, which is used by program OIC to define the integration radius and the model-cell for cross-rotation function calculations.

ROTING

ROTING calculates fast rotation functions (self-, cross- and locked-rotation functions) [2]. The rotation function (Patterson Overlap) is defined by

$$PO(\mathbf{R}) = \frac{1}{v} \int_{\Omega} P^{(t)}(\mathbf{r})\ P^{(s)}(\mathbf{R}^{-1}\mathbf{r})\ d^3\mathbf{r}$$

where $P^{(t)}$ and $P^{(s)}$ are the target and search Patterson functions, respectively. Note that the rotation applied to the search model is $\mathbf{R}$. By expanding the Patterson functions in spherical harmonics, PO may be casted into the form

$$PO(\phi,\theta,\psi) = \sum_{\ell=LMIN}^{LSUP} \ \sum_{m,m'=-\ell}^{\ell} \left[\sum_{n=1}^{N} \overline{e^{(t)}_{\ell,m,n}}\ e^{(s)}_{\ell,m',n} \right] \ d^{\,\ell}_{m,m'}(\theta)\ e^{i(m\phi+m'\psi)}$$

with

$$e_{\ell,m,n} = \sqrt{12\pi(2(\ell+2n)-1)}\ \sum_{\mathbf{H}} \frac{|F_{\mathbf{H}}|^2}{V} Y_{\ell,m}(\mathbf{H}/H) \frac{j_{\ell+2n-1}(2\pi HR)}{2\pi HR}$$

[2] International Tables for Crystallography (2001). Volume F: Crystallography of Biological Macromolecules". Ed. Rosmann, M.G. and Arnold, E., Dordrecht, Kluwer Academic Publishers: "Rotation Functions" by Jorge Navaza.

Calling command: ./e/roting {INP} {#} {OUT}

Input:	**./i/{INP}.i1**	=	input described below.
	./f/xudi	=	binary file of packed and sorted H,K,L,F^{obs}.
	./f/tabl**{#}**	=	binary file of molecular scattering factors for model **{#}**.
Output:	./f/elmx**{#}**	=	binary file of spherical-harmonics expansion coefficients for the crystal.
	./f/elmn**{#}**	=	binary file of spherical-harmonics expansion coefficients for model **{#}**.
	./o/{OUT}.s	=	rotation function output.

To recover the rotation function map, erase in script ./e/roting , the symbol "#" in the line "#mv $AMOREF/fort.93 rotmap". ./rotmap is a binary file with several registers:

```
real rotmap(np,nf,nt),theta(nt)
write(file) np,nf,nt,p,f,(theta(n),n=1,nt)
 do n=1,nt
write(file) ((rotmap(ip,if,n),ip=1,np),if=1,nf)
 enddo
```

where:

np,nf,nt	=	number of sampling points in each direction.
p,f	=	ψ and ϕ Euler cell dimensions (in degrees).
(theta(n), n=1,nt)	=	θ sampling set (in degrees).

Input: file ./i/{INP}.i1 (general form)

```
roting +*+*+*+*+*+*+*+*+*
3   0   1   0   :printing
gene ++++++++++++++++++++++
. . .
elmn ++++++++++++++++++++++
. . .
rota ++++++++++++++++++++++
. . .
```

Description

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

2) Printing options.

3) MKEY (format A4)

Keyword defining mode:
'gene' generates model structure factors in a suitable cell;
'elmn' calculates spherical- harmonics expansion coefficients for crystal or models;
'rota' calculates rotation functions.

The three modes may be used independently.

Different modes:

“gene” mode:

```
gene ++++++++++++++++++++++
11   92   :logical units
title : model fragment number 1
127.790   98.030   90.050   90.000   90.000   90.000
x,y,z * end
1
15.00   3.50
0
```

Description

1) MKEY (format A4) = 'gene'.

2) LUN1 LUN2
   Logical units (see script ./e/roting):
   LUN1 = input binary file of molecular scattering factors (./f/tabl{#}).
   LUN2 = output binary file of packed and sorted Fourier coefficients (as produced by SORTING).

3) Title (format A80).

4) Model cell for rotation function (default cell = model-box + resolution + integration-radius).

5) Symmetry operations (lower case), finishing '* end' (usually P1).

6) NORT
   Code to define an orthogonal reference frame (usually 1).

7) DMAX DMIN
   Resolution limits. Generates data within this range (in Å).

8) NBOD
   Option to generate structure factors with molecules placed at desired positions (usually 0).

"elmn" mode:

```
elmn ++++++++++++++++++++++
10   91   :logical units
* D44HEL **
99.700   167.300   84.700   90.000   90.000   90.000
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   0.0
15.00   3.50   0.00   42.40
```

Description

1) MKEY (format A4) = 'elmn'.

2) LUN1 LUN2
   Logical units (see script ./e/roting):
   LUN1 = input binary file of packed and sorted Fourier coefficients (as produced by SORTING or ROTING -mode "gene").
   LUN2 = output binary file of spherical-harmonics expansion coefficients (./f/elmx{#} or ./f/elmn{#}).

3) Title (format A80).

4) Cell.

5) Symmetry operations (lower case), finishing '* end'.

6) NORT
   Code to define an orthogonal reference frame.

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

8) DMAX DMIN RMIN RMAX
   DMAX
   DMIN = resolution limits.

   RMIN
   RMAX = integration radii (in Å).

“rota” mode:

```
rota ++++++++++++++++++++++
91   92   9   1   :logical units
* D44HEL **
cross
6   500
2.5
0   0
0.5   1000   0.1E-04
```

Description

1) MKEY (format A4) = 'rota'.

2) LUN1 LUN2 LUN3 LUN4

   Logical units (see script ./e/roting):

LUN1	=	input binary file of spherical-harmonics expansion coefficients for fixed crystal (usually ./f/elmx{#}).
LUN2	=	input binary file of spherical-harmonics expansion coefficients for rotating crystal (usually ./f/elmn{#}).
LUN3	=	rotation function output (./o/{OUT}.s).
LUN4	=	model identification number. Also logical-unit identifier for molecular scattering factors. Usually, logical-unit = LUN4+10.

3) Title (format A80).

4) RKEY (format A5)

   Keyword defining mode:

'cross'	computes cross-rotation function;
'self'	computes self rotation.

5) LMIN LSUP

   Expansions between LMIN and LSUP are used. The ℓ-expansion controls the angular resolution. Low order terms are governed by the crystal symmetry; excluding them may reduce the final peak heights, but make the rotation parameters more precise and make multiple solutions have more equal heights.

6) STEP

   Step size for ϕ, θ and ψ (in degrees).

7) NRS1 NRS2

   Used to produce several shifts prior to the rotation function calculation (usually 0 0). This allows locked-rotation function calculations.

8) CUTR NPIC DELT

CUTR	=	cutoff in rotation function output; first selects all peaks above CUTR × maximum-peak-height.
NPIC	=	maximum number of peaks to output of rotation function.
DELT	=	used in peak-search to avoid spurious peaks.

Example

Self Rotation:

The self rotation may be calculated by using the following input cards:

```
roting +*+*+*+*+*+*+*+*+*
3   0   1   0   :printing
elmn ++++++++++++++++++++++
10   91   :logical units
* D44HEL **
99.700   167.300   84.700   90.000   90.000   90.000
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   0.0
15.00   3.50   0.00   42.40
rota ++++++++++++++++++++++
91   0   9   0   :logical units
* D44HEL ** self
self
6   500
2.5
0   0
0.2   1000   0.1E-04
```

If the expansion-coefficients already exist (./f/elmx{#}), then the following input may be used:

```
roting +*+*+*+*+*+*+*+*+*
3   0   1   0   :printing
rota ++++++++++++++++++++++
91   0   9   0   :logical units
* D44HEL ** self
self
6   500
2.5
0   0
0.2   1000   0.1E-04
```

Locked Cross-Rotation:

The locked cross-rotation is the average of the values of the cross rotation at orientations related by the NCS elements $\mathbf{S}_n, n = 1, \ldots, N$:

$$PO_L(\mathbf{R}) = \sum_{n=1}^{N} PO(\mathbf{S}_n\mathbf{R}) \,/N \ .$$

By redefining the target function, it can be computed as an ordinary cross rotation,

$$\begin{aligned} PO_L(\mathbf{R}) &= \sum_{n=1}^{N} \frac{1}{v} \int_\Omega P^{(t)}(\mathbf{r}) P^{(s)}(\mathbf{R}^{-1}\mathbf{S}_n^{-1}\mathbf{r}) \ d^3\mathbf{r} \,/N \\ &= \frac{1}{v} \int_\Omega \left(\sum_{n=1}^{N} P^{(t)}(\mathbf{S}_n\mathbf{r}) \,/N \right) P^{(s)}(\mathbf{R}^{-1}\mathbf{r}) \ d^3\mathbf{r} \ , \end{aligned}$$

with the target Patterson function substituted by the average over the NCS of the rotated target functions. Let ϕ , θ , ψ be Euler angles of a self-rotation peak; the parameters of the inverse rotation are $\phi' = 180 - \psi$, $\theta' = \theta$, $\psi' = 180 - \phi$. Then, the locked cross-rotation may be calculated by using the following input ($N = 2$):

```
roting +*+*+*+*+*+*+*+*+*
3   0   1   0   :printing
rota ++++++++++++++++++++++
91   92   9   1   :logical units
* D44HEL ** locked
cross
6   500
2.5
2   0
0.    0.    0.
φ′   θ′   ψ′
0.5   1000   0.1E-04
```

TRAING

TRAING calculates fast translation functions [3]. The Fourier coefficient of the search model, rotated by $\mathbf{R}$ and placed at $\mathbf{T}$, is given by the expression:

$$\begin{aligned} F^{cal}_{\mathbf{H}}(\mathbf{T}) &= \sum_{g=1}^{G} \left(f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}\mathbf{O}_m)\; e^{2\pi i \mathbf{H}\mathbf{t}_g} \right)\; e^{2\pi i \mathbf{H}\mathbf{M}_g\mathbf{T}} \\ &= \sum_{g=1}^{G} u^m_g(\mathbf{H})\; e^{2\pi i \mathbf{H}\mathbf{M}_g\mathbf{T}} \end{aligned}$$

and the corresponding intensity by

$$\begin{aligned} I^{cal}_{\mathbf{H}}(\mathbf{T}) &= \sum_{g,g'=1}^{G} f_m(\mathbf{H}\mathbf{M}_g\mathbf{D}\mathbf{R}_m\mathbf{O}_m)\; \overline{f_m(\mathbf{H}\mathbf{M}_{g'}\mathbf{D}\mathbf{R}_m\mathbf{O}_m)}\; e^{2\pi i \mathbf{H}\left(\left(\mathbf{M}_g - \mathbf{M}_{g'}\right)\mathbf{T} + \mathbf{t}_g - \mathbf{t}_{g'}\right)} \\ &= \sum_{g,g'=1}^{G} u^m_g(\mathbf{H})\; \overline{u^m_{g'}(\mathbf{H})}\; e^{2\pi i \mathbf{H}\left(\mathbf{M}_g - \mathbf{M}_{g'}\right)\mathbf{T}}. \end{aligned}$$

Several translation functions may be calculated (“centered” variables are denoted by $\overline{\overline{I_{\mathbf{H}}}} = I_{\mathbf{H}} - \langle I_{\mathbf{H}} \rangle$):

- centered-overlap

$$\begin{aligned} CO(\mathbf{T}) &= \sum_{\mathbf{H}} \overline{\overline{I^{obs}_{\mathbf{H}}}} \times \overline{\overline{I^{cal}_{\mathbf{H}}(\mathbf{T})}} \\ &\propto \sum_{g,g'=1}^{G} \sum_{\mathbf{H}} \overline{\overline{I^{obs}_{\mathbf{H}}}}\; \overline{u^m_g(\mathbf{H})}\; u^m_{g'}(\mathbf{H})\; e^{-2\pi i \mathbf{H}\left(\mathbf{M}_g - \mathbf{M}_{g'}\right)\mathbf{T}} \end{aligned}$$

- Harada-Lifchitz

$$HL(\mathbf{T}) = \left(\sum_{\mathbf{H}} \overline{\overline{I^{obs}_{\mathbf{H}}}} \times \overline{\overline{I^{cal}_{\mathbf{H}}(\mathbf{T})}} \right) \Big/ \sum_{\mathbf{H}} I^{cal}_{\mathbf{H}}(\mathbf{T})$$

- correlation-coefficient

$$CC(\mathbf{T}) = \left(\sum_{\mathbf{H}} \overline{\overline{I^{obs}_{\mathbf{H}}}} \times \overline{\overline{I^{cal}_{\mathbf{H}}(\mathbf{T})}} \right) \Big/ \sqrt{\left(\sum_{\mathbf{H}} \overline{\overline{I^{obs}_{\mathbf{H}}}}^{\,2} \right) \times \left(\sum_{\mathbf{H}} \overline{\overline{I^{cal}_{\mathbf{H}}(\mathbf{T})}}^{\,2} \right)}$$

[3] Acta Cryst. A51, 445-449. (1995) ”On the fast translation functions for molecular replacement” by Jorge Navaza & Elena Vernoslova.

- phased-translation

  1. without “external” phases

$$\begin{aligned} PT(\mathbf{T}) &= \sum_{g,g'=1}^{G} \sum_{\mathbf{H}} \Big(|F^{obs}_{\mathbf{H}}/u^m_g(\mathbf{H})| + |F^{obs}_{\mathbf{H}}/u^m_{g'}(\mathbf{H})| - 2\ SCAL \Big) \\ &\times \ \overline{u^m_g(\mathbf{H})}\ u^m_{g'}(\mathbf{H})\ e^{-2\pi i \mathbf{H}(\mathbf{M}_g - \mathbf{M}_{g'})\mathbf{T}} \end{aligned}$$

  2. with “external” phases

$$PTF(\mathbf{T}) = \sum_{\mathbf{H}} |F^{obs}_{\mathbf{H}}|\ e^{i\phi^{ext}_{\mathbf{H}}} \times \overline{F^{cal}_{\mathbf{H}}(\mathbf{T})}$$

- many-body translation

  1. phased-translation

$$PTN(\mathbf{T}) = \sum_{\mathbf{H}} \left(\ \left| \frac{F^{obs}_{\mathbf{H}}}{F^{fix}_{\mathbf{H}}} \right| - SCAL \right) F^{fix}_{\mathbf{H}} \times \overline{F^{cal}_{\mathbf{H}}(\mathbf{T})}$$

  2. for all others, replace

$$F^{cal}_{\mathbf{H}}(\mathbf{T}) \rightarrow F^{cal}_{\mathbf{H}}(\mathbf{T}) + F^{fix}_{\mathbf{H}}$$

Calling command: ./e/traing {INP} {OUT}

Input:	**./i/{INP}.i1**	=	input described below.
	./f/xudi	=	binary file of packed and sorted H,K,L,F^{obs}.
	./f/tabl{#}	=	binary files of molecular scattering factors; {#} = 1,NTYP.
Output:	**./o/{OUT}.s**	=	translation function output.

To recover the last fast translation function map, erase in script ./e/traing , the symbol “#” in line “#mv $AMOREF/fort.94 tramap”. ./tramap is a binary file with several registers:

```
real tramap(nx,ny,nz)
write(file) nx,ny,nz,a,b,c,alpha,beta,gamma
 do iz=1,nz
write(file) ((tramap(ix,iy,iz),ix=1,nx),iy=1,ny)
 enddo
```

where:

nx,ny,nz = number of sampling points in each direction.

a,b,c,alpha,beta,gamma = cell parameters (in Å and degrees).

Input: file ./i/{INP}.i1

```
traing +*+*+*+*+*+*+*+*+*
7  1  1  0  1  0  1  1  :printing
10  9  :logical units
* D44HEL **
99.700  167.300  84.700  90.000  90.000  90.000
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  0.0
15.00  3.50  0.88  1.00
3 ++++++++++++++++++++
c-o  0.0
0.50  30  1.e-4
 > 1  102.3  25.7   12.8  0.0755  0.3081  0.4560  27.6  52.1  54.0  42.00
 > 1  135.5  43.0  320.5  0.4613  0.2053  0.0096  38.6  48.1  41.1  37.00
 # 2   73.6  68.9  346.3  0.0000  0.0000  0.0000   7.6
3 ++++++++++++++++++++
c-o  0.0
0.50  30  1.e-4
 > 1  102.3  25.7   12.8  0.0755  0.3081  0.4560  27.6  52.1  54.0  42.10
 > 1  135.5  43.0  320.5  0.4613  0.2053  0.0096  38.6  48.1  41.1  37.20
 # 2   74.8  61.6  309.9  0.0000  0.0000  0.0000   4.0
. . .
```

Description

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

2) Printing options.

3) LUN1 LUN2

   Logical units (see script ./e/traing):

   LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).

   LUN2 = translation function output (./o/{OUT}.s).

4) Title (format A80).

5) Cell.

6) Symmetry operations (lower case), finishing '* end'.

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} (e.g. -5 to sharpen data).

9) DMAX DMIN TMIN SHAR

   DMAX,DMIN = data resolution limits.

   TMIN = fast translation function resolution (default value is DMIN/4).

   SHAR = Shannon rate (the greater SHAR, the finer the mesh). Defines the crystal cell sampling as

   $NX \approx 2 \times a / (TMIN/SHAR)$

   $NY \approx 2 \times b / (TMIN/SHAR)$

   $NZ \approx 2 \times c / (TMIN/SHAR)$

Then add this information which is extracted from ROTING, TRAING or FITING outputs:

10) NBOD

Number of molecules (n-body mode).

11) TKEY SCAL (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.

SCAL = for phased-translation, to scale the phasing-model subtraction.

12) CUTT NPIC DELT

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 = maximum number of peaks to output of translation function. The program computes correlations and R-factors.

DELT = used in peak-search to avoid spurious peaks.

13) $> |\# \mu_m \phi_m \theta_m \psi_m x_m y_m z_m$ (NBOD cards; m = 1,NBOD)

This is a formatted line. The format is decided by the user when installing the package. If NBOD = 1 search is done over the Cheshire cell. Otherwise last will be translated over whole cell while keeping the others fixed.

μ_m = model identification number; also logical-unit identifier for molecular scattering factors corresponding to the m-th molecule. Usually, logical-unit = $\mu_m + 10$.

ϕ_m θ_m ψ_m = Euler angles for m-th molecule.

x_m y_m z_m = translations (fractionnary) for m-th molecule. For the last one, read but not used.

Repeat 10) — 13) for other positions and orientations.

FITING

FITING performs rigid-body refinement [4]. Minimizes

$$\sum_{\mathbf{H}} \left(F^{obs}_{\mathbf{H}} - \frac{e^{+B|\mathbf{H}|^2}}{Scale} |\sum_{m} F^{cal\,(m)}_{\mathbf{H}}(\phi_m, \theta_m, \psi_m, x_m, y_m, z_m)| \right)^2$$

with respect to B, $Scale$ and rotation and translation parameters.

Calling command: ./e/fiting {INP} {OUT}

Input:	**./i/{INP}.i1**	=	input described below.
	./f/xudi	=	binary file of packed and sorted H,K,L,$\mathrm{F}^{\mathrm{obs}}$.
	./f/tabl{#}	=	binary files of molecular scattering factors; {#} = 1,NTYP.
Output:	**./o/{OUT}.s**	=	fast rigid-body refinement output.

[4] J. Appl. Cryst. 25, 281-284. (1992) ”Fast Rigid-body Refinement for Molecular Replacement Techniques”Pino Castellano, Glaucius Oliva & Jorge Navaza (1992).

Input: file ./i/{INP}.i1

```
fiting +*+*+*+*+*+*+*+*+*
7   1   1   0   1   0   1   1   :printing
10   9   :logical units
* D44HEL **
99.700   167.300   84.700   90.000   90.000   90.000
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   0.0
15.00   3.50
inertia tensors   2
 ∗ 1   140.30   100.20   78.10   0.00   0.00   0.00
 ∗ 2    80.00    66.10   50.00   0.00   0.00   0.00
3 ++++++++++++++++++++
0   1   1   1   1   1   1
1   20   0.04
  # 1   102.3   25.7    12.8   0.0755   0.3081   0.4560   27.6   52.1   54.0   42.00
  # 1   135.5   43.0   320.5   0.4613   0.2053   0.0096   38.6   48.1   41.1   37.00
  # 2    74.8   61.6   309.9   0.2951   0.9939   0.8339   43.9   45.5   41.8   28.00
...
```

Description

1) Keyword (format A7) = 'fiting '.

2) Printing options.

3) LUN1 LUN2
   Logical units (see script ./e/fiting):
   LUN1 = input binary file of packed and sorted H,K,L,F^{obs} (./f/xudi).
   LUN2 = fast rigid-body refinement output (./o/{OUT}.s).

4) Title (format A80).

5) Cell.

6) Symmetry operations (lower case), finishing '* end'.

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} (e.g. -5 to sharpen data).

9) DMAX DMIN
   DMAX,DMIN = resolution limits.

10) NTYP (format *, after column 15)
    Number of different models whose positions are to be refined.

11) $* \; m \; Q^m_{xx} \; Q^m_{yy} \; Q^m_{zz} \; Q^m_{xy} \; Q^m_{xz} \; Q^m_{yz}$ (NTYP cards; m = 1,NTYP)
    This is a formatted line. The format is decided by the user when installing the package.
    m = model identification number.
    Q^m_{ij} = principal moments of inertia for m-th model.

Then add this information which is extracted from TRAING or FITING outputs (or ROTING for space group P1):

12) NBOD

Number of molecules (n-body mode).

13) Refinement flags (7 flags).

Flags for refinement of B-factor, Euler angles and translations.

$= \left| \begin{array}{l} 0 \text{ (no refinement);} \\ 1 \text{ (refinement).} \end{array} \right.$

14) NCYC NITE RMSS

NCYC = number of times the NBOD bodies are alternately refined.
= 0 sets NCYC = NBOD.

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.

15) # μ_m ϕ_m θ_m ψ_m x_m y_m z_m (NBOD cards; m = 1,NBOD)

This is a formatted line. The format is decided by the user when installing the package. If NBOD > 1, last is refined first, while the others are kept fixed; then proceeds cyclically.

μ_m = model identification number; also logical-unit identifier for molecular scattering factors corresponding to the m-th molecule. Usually, logical-unit = $\mu_m + 10$.

ϕ_m θ_m ψ_m = Euler angles for m-th molecule.

x_m y_m z_m = translations (fractionnary) for m-th molecule.

Repeat 12) to 15) for other positions.

MR2IC

MR2IC reads the rotation and translation applied by TABLING to the input models, and the refined rotations and translations output from FITING, to work out the final rotations and translations parameters to apply to the initial models.

Calling command: ./e/mr2ic {INP} {OUT}

Standard input:	**./e/mr2ic**		
Input:	**./o/{INP}.s**	=	ascii file of FITING output.
	./d/data.d	=	ascii file of main AMoRe input.
	./o/tabl{#}.s	=	ascii file of TABLING outputs; {#} = 1,NTYP.
Output:	**./o/{OUT}.s**	=	output positional parameters.

Standard Input: (file ./e/mr2ic)

```
mr2ic   1
```

Description

1) OKEY NUMF (format A6,*)
   OKEY = 'mr2ic '
   NUMF = selects up to a maximum of NUMF positions to output (in decreasing order of correlation).

CORDING

CORDING applies rotations and translations from a FITING output to the search models and generates the pdb files solution of the molecular replacement problem. 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, the pdb files of the independent molecules are generated.

Calling command: ./e/cording {INP}

Input:	**./o/{INP}.s**	=	ascii file of FITING output.
	./d/data.d	=	ascii file of main AMoRe input.
	./d/xyz{#}.d	=	ascii file of input coordinates.
	./o/tabl{#}.s	=	ascii file of TABLING outputs; {#} = 1,NTYP.
Output:	**./sol1**	=	output pdb file.

Standard Input: (file ./e/cording)

```
pack
0.5 0.5 0.5
1
pdb
```

Description

1) CKEY (format A4)

CKEY =
- 'pack' = first molecule closest to CENTER, following closest to precedent ones;
- 'cent' = all molecules closest to CENTER;
- 'okay' = molecules at FITING positions.

2) O_x O_y O_z

CENTER position in fractionnary coordinates.

3) NSOL

Number of solutions to output; usually one.

4) PKEY (format A3)

PKEY =
- 'pdb' = generates PDB files;
- 'shl' = generates coordinates in SHELX format.

OIC

OIC creates inputs to the main molecular replacement programs by selecting and combining potential solutions, which are sorted according to the values of the correlation coefficient.

Calling command:

```
./e/oic  {NOM}  {#}    {OUT}            ; 3 arguments.

./e/oic  {NOM}  {IN1}  {OUT}            ; 3 arguments.

./e/oic  {NOM}  {IN1}  {IN2}    {OUT}   ; 4 arguments.
```

Five situations are possible according to **{NOM}** and number of arguments:

- for ROTING inputs, model dimensions are read from file ./o/tabl**{#}**.s ; 3 arguments.
- for one-body TRAING inputs, orientations to translate are read from file ./o/**{IN1}**.s ; 3 arguments.
- for n-body TRAING inputs, orientations are read from file ./o/**{IN1}**.s , and the fixed position from file ./o/**{IN2}**.s . If orientations are read from a n-body output, only those corresponding to the last body are taken into account; 4 arguments.
- for n-body TRAING inputs, where the orientations and the fixed position are read from the same file ./o/**{IN1}**.s ; 3 arguments.
- for FITING inputs, positions to refine are read from file ./o/**{IN1}**.s ; 3 arguments.

Input:	**./i/{NOM}.i2**	=	input described below.
	./d/data.d	=	ascii file of main AMoRe input.
	./o/tabl{#}.s	=	ascii files of TABLING outputs; {#} = 1,NTYP.
	./o/{IN1}.s **./o/{IN2}.s**	=	ascii files of ROTING, TRAING or FITING outputs.
Output:	**./i/{OUT}.i1**	=	OIC output, i.e. input to main molecular replacement programs.

The OIC program needs the ./o/tabl{#}.s file to define the integration radius and the model cell for cross-rotation function calculations. This file is not created if the molecular scattering factors of the search model electron density is not calculated with TABLING.

Three different input modes:

Input: file ./i/{NOM}.i2 (ROTING mode)

```
oic roting +*+*+*+*+*+*+*+*+*
* D44HEL **
model:   1
15.00   3.50
1.0
2 10
2.5
0.5   1000
```

Description

1) Keyword defining mode (format A10) = 'oic roting'.

2) Title (format A80).

3) MTYP (format *, after column 7)
   Model-identification number.

4) DMAX DMIN
   Resolution limits (in Å).

5) RATE
   Defines the integration radius as RATE × Molrad, where Molrad is the radius of the smallest sphere, with origin at CoM, containing the whole molecule.

6) 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.

7) STEP
   Step size for ϕ, θ and ψ (in degrees).

8) CUTR NPIC
   CUTR = cutoff in rotation function output; first selects all peaks above CUTR × maximum-peak-height.
   NPIC = maximum number of peaks to output of rotation function.

Input: file ./i/{NOM}.i2 (TRAING mode)

```
oic traing +*+*+*+*+*+*+*+*+*+*
* D44HEL **
n-body-s   1
p-t   1.0
0.0   10
15.00   3.50
0.5   30
fuzz:    2.0
over:    20.0
```

Description

1) Keyword defining mode (format A10) = 'oic traing'.

2) Title (format A80).

3) BKEY RANK (format A10,*)

   BKEY = keyword defining mode. Three possibilities: | 'one-body '
 'n-body '
 'n-body-s ' |

   The last one is for the particular case of n-body translations where the orientations to translate and the fixed position are read from the same file. When the number of arguments is 4, the OIC procedure uses "sed" to erase '-s'.

   RANK = takes as fixed position the one that appears in RANK-th order of decreasing correlation (for n-body).

4) TKEY SCAL (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.

   SCAL = for phased-translation, to scale the phasing-model subtraction.

5) CORR NUMR

   CORR = cutoff to select only those solutions with correlation coefficient greater than CORR.

   NUMR = selects up to a maximum of NUMR orientations for input to translation (in decreasing order of correlation).

6) DMAX DMIN

   Resolution limits (in Å).

7) CUTT NPIC

   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.

8) CUTO (format *, after column 5)

   Cutoff to eliminate orientations to translate differing by less than CUTO degrees.

9) CUTD (format *, after column 5)
Cutoff to eliminate fixed positions with CoM-CoM distance less than CUTD Å (for n-body).

Input: file ./i/{NOM}.i2 (FITING mode)

```
oic fiting +*+*+*+*+*+*+*+*+*
* D44HEL **
0.0   10
15.00   3.50
1   20   0.01   0
fuzz:   2.0
over:   20.0
```

Description

1) Keyword defining mode (format A10) = 'oic fiting'.

2) Title (format A80).

3) CORR NUMT
CORR = cutoff to select only those solutions with correlation coefficient greater than CORR.
NUMT = selects up to a maximum of NUMT positions to refine (in decreasing order of correlation).

4) DMAX DMIN
Resolution limits (in Å).

5) NCYC NITE RMSR BREF
NCYC = number of times the n-bodies are alternately refined.
= 0 sets NCYC = NBOD.
NITE = number of iterations in the least-squares procedure.
RMSR = least-squares stops if the rms. correction to positions is less than RMSR × DMIN.
BREF = flag for refinement of B-factor.

6) CUTO (format *, after column 5)
Cutoff to eliminate orientations differing by less than CUTO degrees (for space group P1).

7) CUTD (format *, after column 5)
Cutoff to eliminate positions with CoM-CoM distance less than CUTD Å.

The OIC_ROTING, OIC_TRAING and OIC_FITING procedures

oic_roting, oic_traing and oic_fiting are the concatenations of OIC with the main molecular replacement programs.

OIC_ROTING

oic_roting computes cross-rotation function.

Calling command: ./e/oic_roting {NOM} {#} {OUT}

equivalent to	./e/oic	{NOM}	{#}	rot
	./e/roting	rot	{#}	{OUT}

Description

./i/{NOM}.i2	=	input to OIC, ROTING mode.
{#}	=	model number.
./i/rot.i1	=	input to ROTING program.
./o/{OUT}.s	=	cross-rotation function output.

OIC_TRAING

oic_traing computes fast translation functions.

Calling command: ./e/oic_traing {NOM} {ANG} {OUT}

equivalent to

```
./e/oic     {NOM}   {ANG}   tra
./e/traing  tra     {OUT}
```

Description

./i/{NOM}.i2 = input to OIC, TRAING mode.

./o/{ANG}.s = filename of ROTING, TRAING or FITING output; orientations to translate (and fixed positions in n-body cases) are read from this file. If the orientations are read from a n-body output, only those corresponding to the last body are taken into account.

./i/tra.i1 = input to TRAING program.

./o/{OUT}.s = translation function output.

Calling command: ./e/oic_traing {NOM} {ANG} {POS} {OUT}

equivalent to

```
./e/oic     {NOM}   {ANG}   {POS}   tra
./e/traing  tra     {OUT}
```

Description

./i/{NOM}.i2 = input to OIC, TRAING mode.

./o/{ANG}.s = filename of ROTING, TRAING or FITING output; orientations to translate (and fixed positions in n-body cases) are read from this file. If the orientations are read from a n-body output, only those corresponding to the last body are taken into account.

./o/{POS}.s = ({POS} $\neq$ {ANG}) filename of TRAING or FITING output (or ROTING for space group P1); fixed positions are read from this file.

./i/tra.i1 = input to TRAING program.

./o/{OUT}.s = translation function output.

OIC_FITING

oic_fiting performs fast rigid-body refinements.

Calling command: ./e/oic_fiting {NOM} {POS} {OUT}

equivalent to

```
./e/oic    {NOM}    {POS}    fit
./e/fiting    fit    {OUT}
```

Description

./i/{NOM}.i2 = input to OIC, FITING mode.

./o/{POS}.s = filename of TRAING or FITING output (or ROTING for space group P1); positions to be refined are read from this file.

./i/fit.i1 = input to FITING program.

./o/{OUT}.s = fast rigid-body refinement output.