

Slide 1

CHARMM Course: Lecture 24

Using Map Objects and Grid Potentials for Protein Docking and Structure Determination

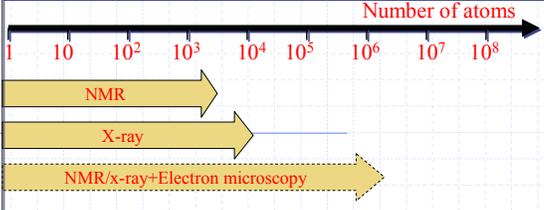
April 4, 2006



The slide features a vertical sidebar on the left containing three logos. From top to bottom: the U.S. Department of Health and Human Services logo, the National Institutes of Health logo, and the National Heart, Lung, and Blood Institute logo. The main content area has a light blue grid background.

Slide 2

Structural Scale of Biomolecular Machineries



Number of atoms

1 10 10² 10³ 10⁴ 10⁵ 10⁶ 10⁷ 10⁸

NMR

X-ray

NMR/x-ray+Electron microscopy

Molecular Modeling

Microscopic understanding of Molecular Systems:
structure, energetics, and dynamics

The diagram shows a horizontal axis labeled 'Number of atoms' with a logarithmic scale from 1 to 10⁸. Three horizontal arrows represent different techniques: NMR (solid arrow, 1 to 10⁴), X-ray (solid arrow, 1 to 10⁵), and NMR/x-ray+Electron microscopy (dashed arrow, 1 to 10⁸). Below the diagram, the text 'Molecular Modeling' is followed by 'Microscopic understanding of Molecular Systems: structure, energetics, and dynamics'.

Slide 3

Why Work with Map Objects?

- Experimental Information: Electron Microscopy maps
- Computational Efficiency: neglect chemical structures of millions of atoms
- Conformational flexibility: low resolution representation

Current Related Fields:

- EM structure determination
- Protein-Protein Interaction
- Biomolecular self-assemblies

Slide 4

Advantages to use map objects

Atomic models	Map objects
Molecular description Large number of atoms: $N \sim 10^6$	Small number of building blocks
Molecular Energetics Pairwise atomic interactions: $N^2 \sim 10^{12}$	Limited block field interactions
Conformational Space Large number of degrees of freedom: M^{3N}	Each block has 6 degrees of freedom

Slide 5

Key Issues in Using Map Objects in Molecular Modeling

Molecular representation: Describe geometric and energetic properties

- VDW core
- charges

Target Functions: Mathematic expression of final states

- Density correlations
- Interaction energies

Conformational search: Process to a final state

- Molecular dynamics
- Monte Carlo
- FFT convolution
- Grid search...

Slide 6

Density Map Objects for EM Fitting

map	x_0, y_0, z_0	starting position
	$\delta x, \delta y, \delta z$	grid sizes
	n_x, n_y, n_z	grid numbers
	* ρ	densities
	* L	Laplacian densities
	* C	core indexes
	* Ia	Atom indexes
	* x_a, y_a, z_a	atom coordinates

Blur to given resolution

Density Map

Slide 7

Target function for EM structure determination

Low resolution: overlap between neighboring proteins
 High noises: ambiguous shape contours

Density correlation: \rightarrow Core-Weighted Density Correlation:

$$DC_{mn} = \frac{\overline{\rho_m \rho_n} - \overline{\rho_m} \overline{\rho_n}}{\delta(\rho_m) \delta(\rho_n)}$$

$$CWDC_{mn} = \frac{(\overline{\rho_m \rho_n})_w - (\overline{\rho_m})_w (\overline{\rho_n})_w}{\delta_w(\rho_m) \delta_w(\rho_n)}$$

Laplacian correlation: \rightarrow Core-Weighted Laplacian Correlation:

$$LC_{mn} = \frac{\overline{\nabla^2 \rho_m \nabla^2 \rho_n} - \overline{\nabla^2 \rho_m} \overline{\nabla^2 \rho_n}}{\delta(\nabla^2 \rho_m) \delta(\nabla^2 \rho_n)}$$

$$CWLC_{mn} = \frac{(\overline{\nabla^2 \rho_m \nabla^2 \rho_n})_w - (\overline{\nabla^2 \rho_m})_w (\overline{\nabla^2 \rho_n})_w}{\delta_w(\nabla^2 \rho_m) \delta_w(\nabla^2 \rho_n)}$$

$\nabla^2 \rho_{ijk} = \rho_{i+1,j,k} + \rho_{i-1,j,k} + \rho_{i,j+1,k} + \rho_{i,j-1,k} + \rho_{i,j,k+1} + \rho_{i,j,k-1} - 6\rho_{ijk}$ Core-weighting factor: \rightarrow

$$\delta_w(X) = \sqrt{(\overline{X^2})_w - (\overline{X})_w^2}$$

$$w_{ln} = \frac{C_1^2}{C_1^2 + C_n^2} \quad (X)_w = \frac{\sum_{i,j,k} w_{mn}(i,j,k) X(i,j,k)}{\sum_{i,j,k} w_{mn}(i,j,k)}$$

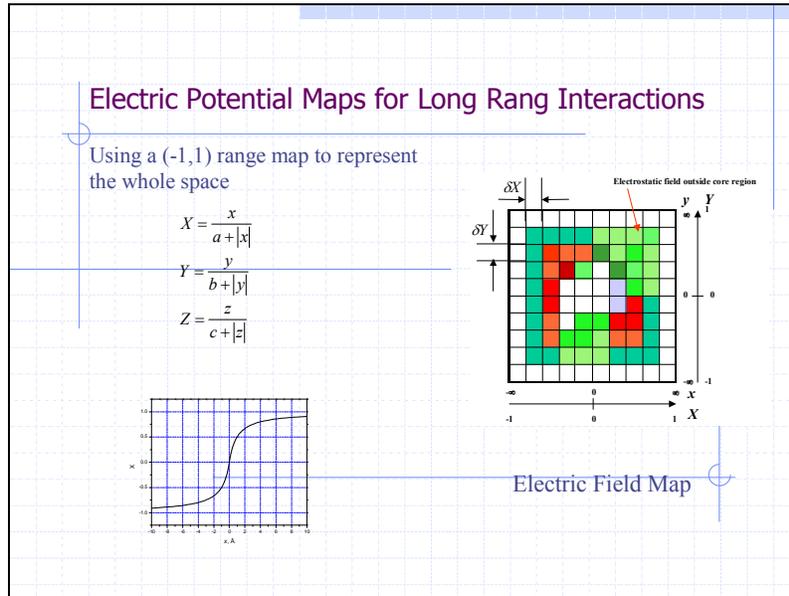
Slide 8

Molecular Map Objects for Energy Calculation

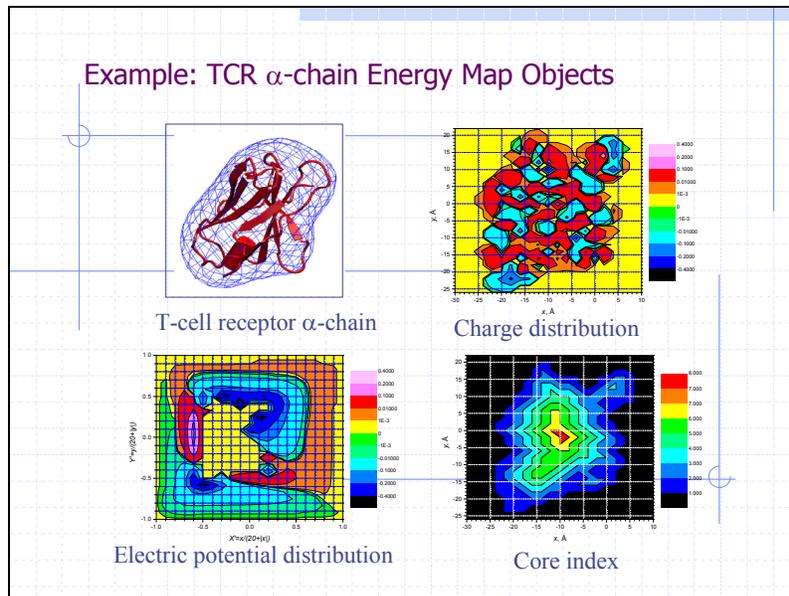
- x_0, y_0, z_0 starting position
- $\delta x, \delta y, \delta z$ grid sizes
- n_x, n_y, n_z grid numbers
- a, b, c coordinate transform factors
- *e charges
- * ϕ electric potential
- *C core indexes
- *Ia Atom indexes
- * $x_a, *y_a, *z_a$ atom coordinates

Molecular Map

Slide 9



Slide 10



Energy function for map object interactions

Energy types: electrostatic, vdw, desolvation, and surface binding.

Interaction style: grid-field interaction ranged up to infinity distances.

Computing cost ~ number of grid points (N)

Function parameters: derived from atomic force field with solvation models.

Energy Calculation for map objects

$$E(M_1, M_2) = E_{12}^{ele} + E_{12}^{desolv} + E_{12}^{binding} + E_{12}^{vdw}$$

$$E_{12}^{ele} = \sum_{MAP1} (e_1 \phi_2)$$

$$\phi_2 = \sum_{MAP1 \& (n_x, n_y, n_z)} \frac{e_2}{r}$$

$$E_{12}^{desolv} = s \delta_1^2 \sum_{MAP1} \left(\frac{C_1 e_1^2}{\delta_1^3 (1 + (C_1/2)^6)(1 + C_2^6)} + \frac{C_2 e_2^2}{\delta_2^3 (1 + C_1^6)(1 + (C_2/2)^6)} \right)$$

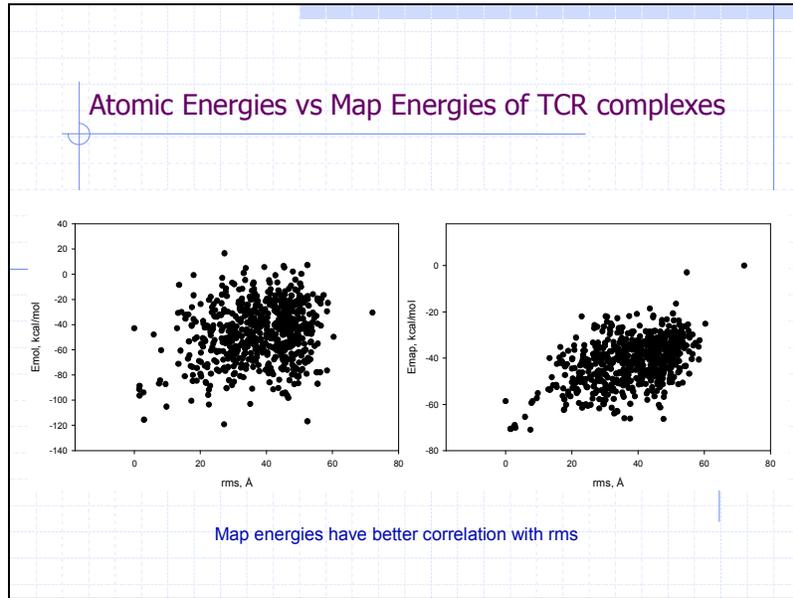
$$E_{12}^{binding} = b \frac{\delta_1^2}{\sqrt{\delta_1^3 \delta_2^3}} \sum_{MAP1} e_1 e_2$$

$$E_{12}^{vdw} = 4\nu \frac{\delta_1^2}{\delta_2^2} \sum_{MAP1} \left(\left(\frac{C_1 C_2}{3} \right)^2 - \frac{C_1 C_2}{3} \right)$$

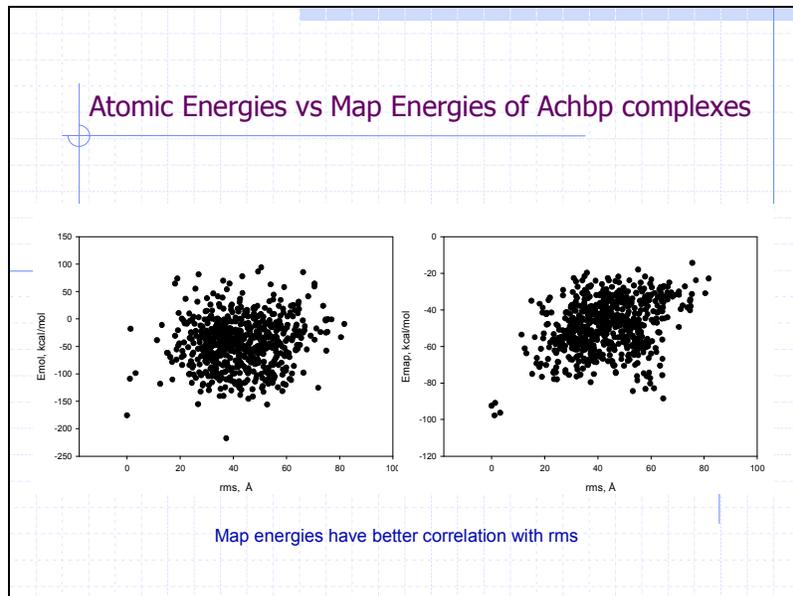
Fit with CHARMM force field (with PB solvation):

$\epsilon=80$
 $s=70 \text{ kcal}/\text{\AA}^2$
 $b=330 \text{ \AA}^2$
 $\nu=0.14 \text{ kcal}/\text{\AA}^2$

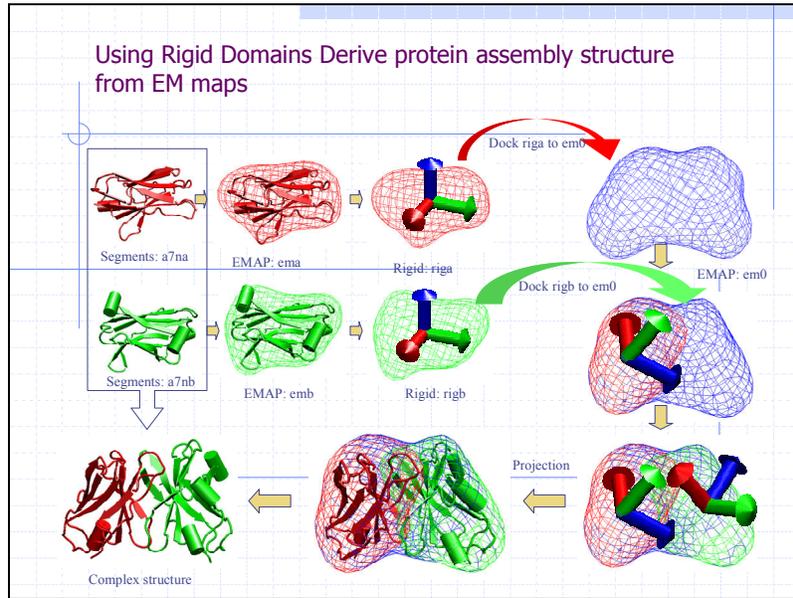
Slide 13



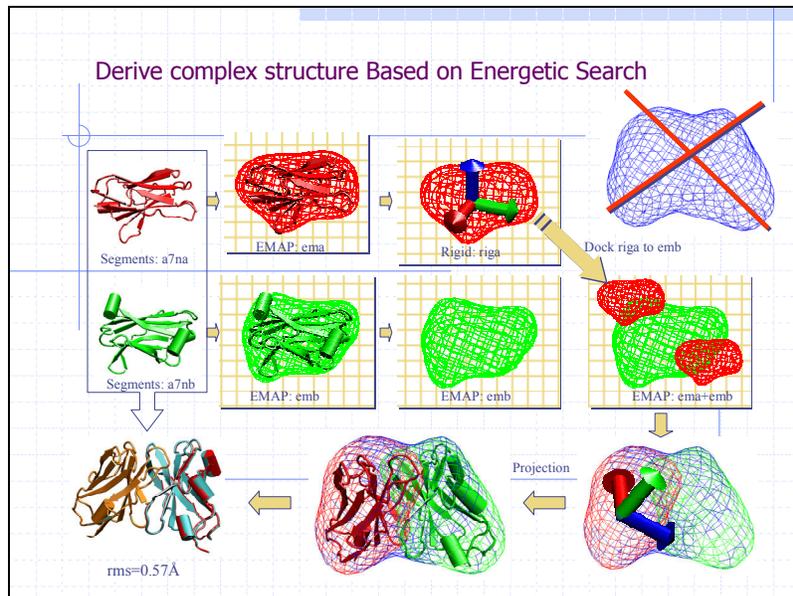
Slide 14



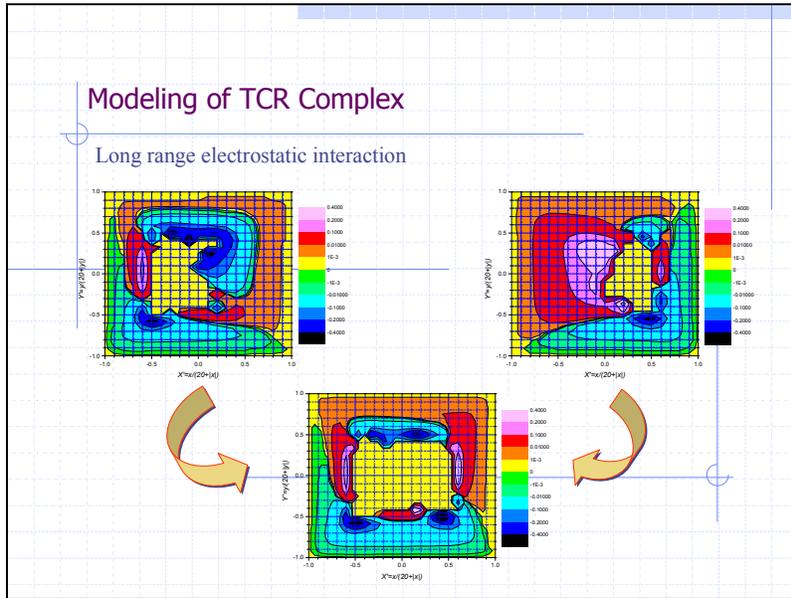
Slide 15



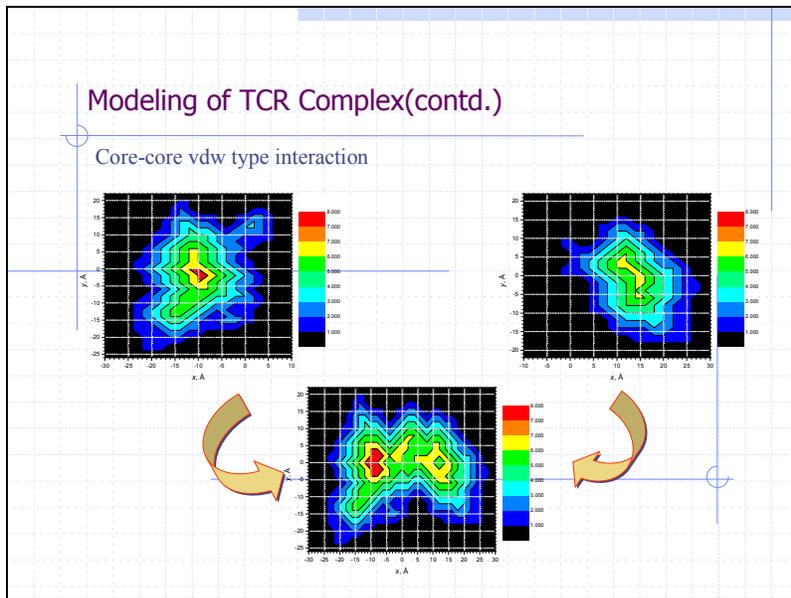
Slide 16



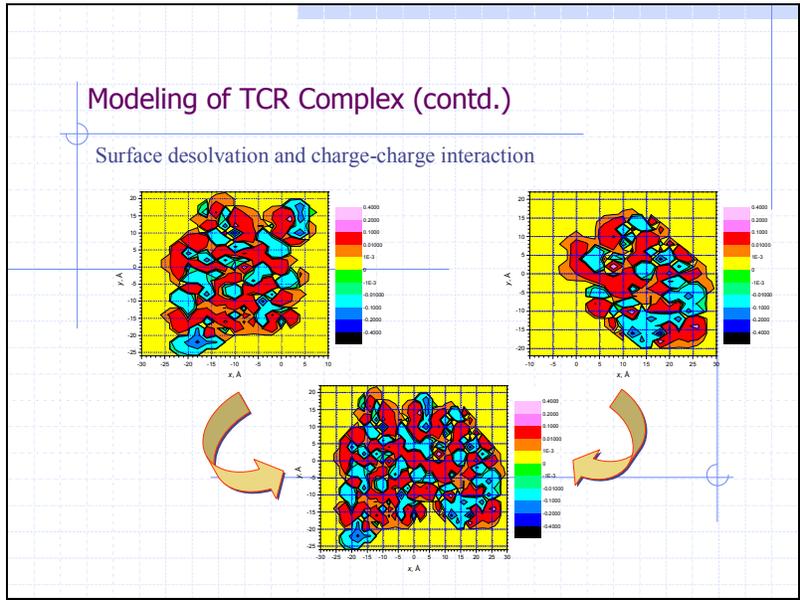
Slide 17



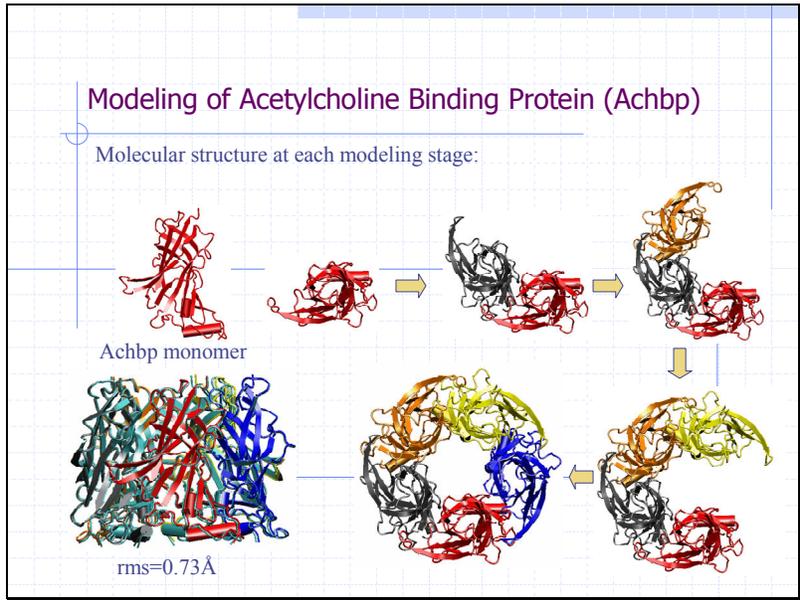
Slide 18

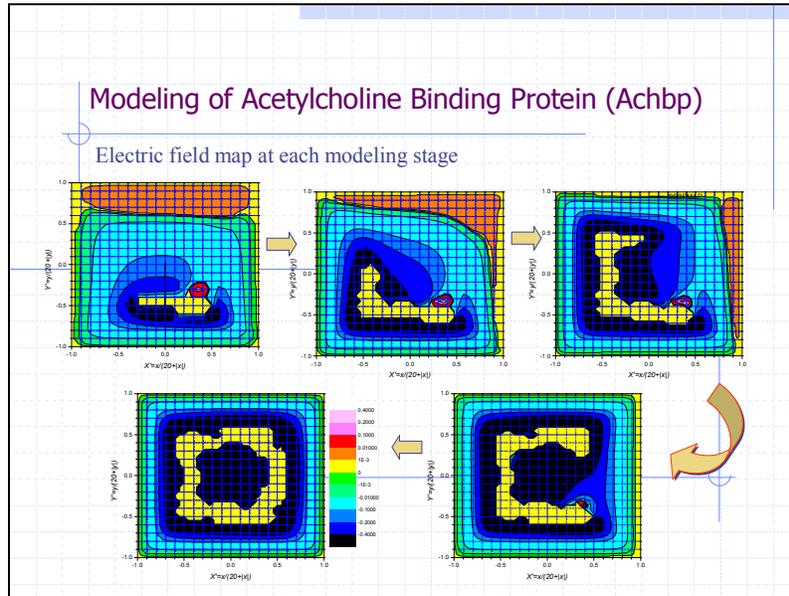


Slide 19



Slide 20





- ### Remarks
- ◆ Map objects provides us an efficient way to study super molecular assemblies.
 - ◆ Potential functions of map objects can be derived from an atomic force field.
 - ◆ Map energies are better correlated to structural accuracy.
 - ◆ CHARMM implementation providing a computational tool for hybrid molecular modeling.

CHARMM c33a2 emap.doc

File: emap]-[**Node:** Top

Up: ([commands.doc](#)) **-- Next:** [Syntax](#)

The EMAP module
Modeling macromolecular assemblies with map objects

by Xiongwu Wu and Bernard R. Brooks
Laboratory of Biophysical Chemistry, NHLBI, NIH

The EMAP module is designed to manipulate map objects as well as interchange between atomic objects and map objects.

A map object is defined as a rectangular space with grid distributions of certain properties. A map object may have its reference atom set which defines the atomic structure used to transfer map to atoms or verse versa.

A rigid domain is defined to represent a map at the position and orientation of an atomic structure. A rigid domain can be moved around as a molecular structure. Many rigid domains can be defined for a map object.

Map objects can be manipulated so as to initialization, resizing, addition, subtraction, reduction, and comparison. With rigid domains, one can perform fitting individual maps to a complex map, constructing complex structure from many components.

Map object manipulation is highly efficient for large system modeling. It is also the necessary approach to derive structure information from electron microscopy experiment.

Field map is defined to describe the electrostatic field generated from a molecule. In addition to the density map used in electron microscopy (EM) image fitting, molecular maps are defined to describe the distribution of atomic properties such as partial charges, vdw cores. Field maps and molecular maps provide a convenient way to evaluate interactions between map objects. Through map energies, the structure of macromolecular assemblies can be modeled through docking or simulation approaches.

- * Menu:
- * [Syntax](#):: Syntax of the EMAP commands
- * [Description](#):: Description of the EMAP functions
- * [Substitution](#):: Description of substitution values
- * [Examples](#):: Usage example of the EMAP commands

Syntax of EMAP Manipulation commands

```
[SYNTAX EMAP manipulation]

! default parameter setting:
EMAP { PARM [RESO real] [RCUT real] -
      [DX real] [DY real] [DZ real] [ICORE int] -
      [AX real] [AY real] [AZ real] [EPS real] -
      [PSOLV real] [PSELE real] [PCORE int] }

! Map file IO:
EMAP { READ mapid NAME filename
      { WRITE mapid NAME filename [DDR|CORE] }

! Map object manipulation
EMAP { CORE mapid [CUT real] [DENSITY|DDR]
      { DDR mapid
      { GENERate mapid [atom-selection] [COMParison-set] [RESO real]
-
      [DX real DY real DZ real] [AS existing_mapid] -
      [FMAP [GRID] [AX real AY real AZ real] }
      { INITIALize emapid [BASE real]
      { REDUce mapid BY mapid [TO mapid]
      { REFERENCE mapid atom-selection
      { RESIZE mapid AS mapid [GRID-only] [BOUNDary-only]
      { SCALE mapid BY real
      { STATistics emapid

! Rigid domain manipulation
EMAP { ASSIgn mapid AS rigid [atom-selection]
      { COMPLEX RIGId rigid [RIGId mapid ...] [APPend] [FIX]}
      { PROJect rigid [ atom-selection ]
      { RESTore rigid
      { ROTate rigid XDIR real YDIR real ZDIR real PHI real
      { SAVE rigid
      { TRANslate rigid XDIR real YDIR real ZDIR real [DIST real]

! Rigid domain IO
EMAP { TRAJectory [OPEN|CLOSE] [READ|WRITE] UNIT integer [NAME
filename| -

! Map object and rigid domain manipulation:
EMAP { ADD rigid TO mapid
      { COPY [MAPId mapid|RIDIId rigid] TO [mapid|rigid]
      { DELEte [MAPId mapid|RIGId rigid]}
```

```

    { DUPLicate [MAPId mapid|RIDId rigid] TO [mapid|rigid] }
    { PSF [MAPId mapid|RIGId rigid] [SKIP int] [RCUT real] }
    { QUERy [MAPId mapid|RIDId rigid] }
    { SUBStract rigid FROM mapid }
    { SUM mapid }

! Conformational evaluation:
EMAP { CORR [MAPId mapid|RIGId rigid] [MAPId mapid|RIGId rigid] [CORE]
[DDR] }
      { INTERaction [MAPId mapid|RIGId rigid] [MAPId mapid|RIGId rigid]
-
          [EPS real] [PSOLV real] [PSELE real] [PCORE int] }

! Conformational optimization:
EMAP { DOCK [FMAP][GTMC [SPACE]] [MANY] [TRAJectory integer] -
      [MAPId mapid] [RIGId rigid [RIGId rigid [...]]] -
      grid-properties MC-parameters Correlation-type -
      Fitting-criteria }

grid-properties::= NTRANslation int NROTation int [SPACE]
MC-parameters::=NCYLc int NSTEp int TEMP real TRAN real ROTA real
Correlation-type::=[CORE [ACORe int] [BCORe real] [CCORe real]] [DDR]
[CORE]
Fitting-criteria::=[LOOP int] [DTCO real] [CFIX real]

! Rigid domain trajectory file IO:
EMAP { TRAJectory [OPEN|CLOSE] UNIT integer [NAME filename] }
      { TRAJectory [READ|WRITE] UNIT integer RIGId integer }

-----

! A shortcut READ command is implmented to simplify the process of
creating
! molecular segments from coordinate files or PDB (default) files.

READ { SEGId segment { PDB [BUILd [SETUp]] }} [UNIT integer]
      { { CARD }}
      { { FREE }}
-----

```

File: [emap](#) |-[Node: **Description**

Up: [Top](#) -[- Previous: [Syntax](#) -[- Next: [Substitution](#)

Descriptions of the map manipulation commands

Map objects are created only by READ, GENERate, or DUPLicate commands. Rigid domains are created only by ASSIgn or DUPLicate commands. All of other commands manipulate existing map objects or rigid domains.

All rigid domains has a storage for backup purpose. Current position and orientation of a rigid domain can be SAVED to the storage and can be RESTored from the storage.

1) The PARM command

The PARM command will set the default values for parameters that would be used in other EMAP commands.

RESO--Map resolution, angstroms;
RCUT--the base density to tell the noise level;
DX, DY, DZ-- Grid interval in map objects;
AX, AY, AZ-- Scaling lengths for reduced coordinates in field maps;
EPS-- Dielectric constant for electrostatic interaction;
PSOLV-- Desolvation energy parameter
PSELE-- Electrostatic solvation parameter
PCORE-- vdw core interaction parameter

2) The READ command

The READ command will create a map object by readin the map information from a map file. Currectly, only CCP4 format is supported.

3) The WRITe command

The WRITe command will write a map object to a map file. Currectly, only CCP4 format is available. Option DDR specify the Laplacian filtered density will be written out, and CORE specify the core indics will be written out.

4) The CORE command

The CORE command will rebuild the core indice of the map object. Two methods, density or Laplacain, can be used for the build up. CUT defines the cutoff density used in the build up.

5) The DDR command

The DDR command will recalculate the Laplacian of the map object.

6) The GENE command

The GENERate command will generate a map object from the coordinates of a selected atom set. The default resolution is 15 angstroms but can be specified for other values. The default map grid properties is DX=DY=DZ=3 angstroms. The grid intervals can be specified by DX, DY, and DZ or taken from other map objects by AS. The generated map object takes the atom set as its reference atom set.

If FMAP is specified, a molecular map will be generated using atomic properties of the molecule. This map contains charge distribution, electrostatic field distribution, and vdw core distribution. GRID option can speed up the calculation in electrostatic field distribution with less accuracy. The field map uses AX, AY, and AZ to define the coordinate scales. Normally, AX, AY, and AZ is set to the gyration radius of the molecule.

7) The INITialize command

The INITialize command set the distribution properties of a map object to be zero, or BASE value, including core indices , throughout its space. The map object should be generated before it can be initialized.

8) The REDUce command

The REDUce command will reduce the first map object by the map object specified after "BY". If a mapid is specified by TO, the result will be put to the mapid. Otherwise, the first map object will be reduced.

9) the REFERENCE command

The REFERENCE command will take the atom-selection as the reference atom set for the map object. ALL rigid domains representing this map object will not change after the reference atom set change.

10) The RESize command

The RESize command will change the map object to have the same grid properties or/and boundary properties as the other map object. Option GRID-only only resizes the grid properties, and BOUNDary-only only resizes boundary properties.

11) The SCALE command

The SCALE command will scale the distribution properties of the map object by the real number specified after "BY".

12) The STATistics command

The STATistics command calculate and print the statistic properties of the distribution properties of the map.

13) The ASSIgn command

The ASSIgn command will create a rigid domain representing the map object. If no atom is selected, a unit vector set at origin will be created for the rigid domain. If atom-selection is given, the relative position and orientation related to the reference atom set will be generated for the rigid domain. The atom-selection should have the same atom number as the reference atom set of the map object. If the map has no reference atom set, a initialized rigid domain will be created and the atom selection is assigned as the reference atom set for the map.

14) The COMPlEx command

The COMPlEx command will define which rigid domains are contained in a complex that will be built with the DOCK command. A COMPlEx command without

APPend option will overwrite previous COMpLex command, while with APPend option the command will add the newly defined rigid domains to the complex. The SEEN option will enable multiple body search during the DOCK procedure, ie., this rigid domain will be seen when docking other rigid domains.

15) The PROJect command

The PROJect command will generate coordinates for the selected atoms based on the reference atom set and the rigid domain. The selected atoms should have the same number of atom as the reference set. coordinates are copied in order of selection and no check is performed.

16) The RESTore command

The RESTore command will copy the stored position and orientation to the rigid domain.

17) The ROTate command

The ROTate command will cause the specified rigid domain to be rotated about the specified axis vector through the map center. The vector need not be normalized, but it must have a non zero length. The PHI value gives the amount of rotation about this axis in degrees.

18) The SAVE command

The SAVE command will copy the position and orientation of the rigid domain to its storage.

19) The TRANslate command

The TRANslate command will cause the position of the rigid domain to be translated. The translation step may be specified by either X,Y, and Z displacements, or by a distance along the specified vector. When no distance

is specified, The XDIR, YDIR, and ZDIR values will be the step vector.
If a distance may be specified, the translation will be along the vector for a distance of DIST.

20) The TRAJjectory command

The TRAJ command perform open, close, read, or write the trajectory file of a rigid domain. The trajectory file stores the translation vector and rotation matrix, configuration energy in a assembly and conformation number and search index.

When OPEN is specified, the file specified by NAME is opened at UNIT channel for accessing by the following EMAP TRAJ command.

When WRITE is specified, the translation vector, rotation matrix, current energy (?EMENG), minimum energy (?EMENGM), and conformational number (?EMNST), and rigid domain index (?EMNSR) will be written to the UNIT channel. These information will be read into the rigid domain if READ is specified and can be shown in CHARMM output by the varable name shown in above parenthese.

CLOSE is used to close the UNIT channel.

21) The ADD command

The ADD command will add the first map object to the map object specified after "TO". The first map object will not change. The second map object will change only its distribution properties, but not its grid and boundary properties.

22) the COPY command

The COPY command will COPY an existing object to another existingone. Only the distribution properties of a map or the position and orientation of a rigid domain will be copied.

23) The DELEte command

The DELEte command will delelte the specified map object or rigid

domain. They can only be deleted in a last in-first out mode by DELETE command.

If the last map object is deleted, all rigid domains representing the map object should be deleted first before the map object can be deleted.

24) the DUPLICATE command

The DUPLICATE command will create an identical map or rigid domain of an existing object .

25) The PSF command

The PSF command will create a segment "EM[nseg]" with atoms "C[0-9]" at grid points. The number [0-9] following C represent the density level at the grid point. SKIP specifies the grid points to be skipped for every representing atom. This command is only for the purpose of viewing the map distribution with a molecular viewer. The segment can be written out in PDB or CHARMM format for displaying.

26) The QUERY command

For map objects, the QUERY command will print out
starting grid numbers (?EMMX,?EMMY,?EMMZ),
grid numbers (?EMLX,?EMLY,?EMLZ),
grid intervals (?EMDX,?EMDY,?EMDZ),
map centers (?EMCX,?EMCY,?EMCZ),
Maximum density(?EMMM),
minimum density (?EMMN),
number of core grids (?EMNC)

For rigid domains, the QUERY command will print out
translation vector (?EMTX,?EMTY,?EMTZ),
rotation matrix (?EMXX,?EMXY,?EMXZ,
?EMYX,?EMYY,?EMYZ,
?EMZX,?EMZY,?EMZZ)

27) The SUBSTRACT command

The SUBSTRACT command will substract the first map object from the map object specified after "FROM". The first map object will not change. The second map object will change only its distribution properties, but not its

grid and boundary properties.

28) The SUM command

The SUM command creates a map object by summing all rigid fragments defined by the COMPLEX command.

29) The CORR command

The CORRELATION command will compute the correlation between the two objects, which can be either map objects or rigid domains or mixed. Option CORE asks for core-weighted correlations, and DDR asks for Laplacian correlations. If both options are specified, the core-weighted Laplacian correlation will be calculated. With the CORE option, the parameters for core-weighting, ACORE, BCORE, and CCORE can be specified. The correlation result can be queried by ?EMCT

30) The INTERaction command

The INTERACTION command will calculate the interaction energy between two rigid domains and/or map objects. These rigid domains or map objects must represent molecular map objects. The interaction energies (?EMENG) consist of VDW core interaction (?EMCORE), electrostatic interaction (?EMELE), desolvation energy (?EMSOLV), and contact energy (?EMCONS). ?EMENGM stores the minimum energy over the history of INTERACTION commands.

31) The DOCK command

The DOCK command will fit the rigid domains defined by COMPLEX command or defined in DOCK command line to a map object or to each other. For single body docking, the grid-threading Monte Carlo (GTMC) is used. If chose MANY option, many-body searching is performed. FMAP will initiate energy-based searching. When FMAP is specified, the grid will be done over protein surface

except SPACE is specified for GTMC. TRAJ can be used to define the writing of rigid domain trajectory. A trajectory file must be opened with EMAP TRAJ OPEN command.

32) The TRAJjectory command

The TRAJ command manipulates the trajectory file of a rigid domain. When OPEN is specified, a file is opened for a channel, while CLOSE will close this channel defined by UNIT. READ or WRITE will read in or write out the translation vector and rotational matrix of given rigid domain.

33) The shortcut READ command

This command provides convenient way to transform a system in PDB file format into new CHARMM segments with given coordinates. When read in segments from a PDB file, one can specify BUILD to generate all atom connectivities and atom types. If there are missing atoms in the PDB file, one can specify SETUP to generate an internal coordinate table of the segments to be used to generate the coordinates of those missing atoms. Each chain in the PDB file will form a new segment named as the given SEGID followed by its segment number. These generated segments are well qualified CHARMM segments and can be used for atom based simulation. This is a very convenient way to generate simulation systems from PDB files. However, It requires that all residue and atom names in the input file are consistent with that in the CHARMM RTF file.

For example:

```
open read unit 10 card name 1b5s.pdb
read segid b5s PDB build setup unit 10
```

This command can also be used to create a new segment from either a PDB file (PDB), a CHARMM coordinate file (CARD), or a free format coordinate file (FREE). BUILD and SETUP option should not be specified. Please be aware of that this segment does not have any atom connectivity information and cannot be used for atom-based molecular simulation. Instead, this segment can be used to generate a map object needed in the EMAP module.

With this command, a map object can be quickly converted from a PDB structure.
(See examples in this document)

File: Emap]-[Node: Substitution

Up: [Top](#) -- Previous: [Description](#) -- Next: [Examples](#)

MAP object Manipulation Values

There are some variables that can be used in titles or CHARMM commands that are set by some of the EMAP manipulation commands. Here is a summary and description of each variable.

'EMCT'

The correlation value calculated by the CORRelation command.

File: Emap]-[Node: Examples

Up: [Top](#) -- Previous: [Substitution](#) -- Next: [Top](#)

Examples to use EMAP module

1. Read in map file and creat a map object
EMAP READ map NAME "a7n.ccp4"

2. Read in PDB files and creat segments

```
OPEN READ UNIT 16 CARD NAME a7na.pdb  
READ SEGId a7na UNIT 16
```

```
OPEN READ UNIT 17 CARD NAME a7nb.pdb  
READ SEGId a7nb UNIT 16
```

3. Generate map objects from structures

```
EMAP GENERate mapa SELEct SEGId a7na END  
EMAP GENERate mapb SELEct SEGId a7nb END
```

4. Assign rigid domains for fitting

```
EMAP ASSIgn mapa AS riga SELE SEGId a7na END  
EMAP ASSIgn mapb AS rigb SELE SEGId a7nb END
```

5. Perform GTMC fitting with default parameters
EMAP DOCK GTMC MAPID map RIGId riga RIGId rigb

6. Perform GTMC fitting with defined parameters
EMAP DOCK GTMC MAPID map RIGId riga RIGId rigb ntran 3 nrot 3 -
ncyc 50 nstep 100 tran 15 rota 30 CORE DDR

7. Perform GTMC fitting with many-body search approach
EMAP DOCK GTMC MAPID map RIGId riga RIGId rigb many ntran 2 nrot 2 -
ncyc 50 nstep 100 tran 15 rota 30 DDR

8. Project rigid domain to obtain fitted coordinates
EMAP PROJ RIGA SELE SEGI A7NA END
EMAP PROJ RIGB SELE SEGI A7NB END

9. Compare the fitting of each rigid domain
EMAP CORR MAPID MAP RIGID RIGA DDR CORE
EMAP CORR MAPID MAP RIGID RIGB DDR CORE

10. Generate the result map: mapn
EMAP DUPLicate MAPID map TO mapn
EMAP INITial mapn
EMAP ADD riga TO mapn
EMAP ADD rigb TO mapn
EMAP SUM mapn

11. Read in a PDB file and create segments with given coordinates
and build missing coordinates
OPEN READ UNIT 10 CARD NAME 1B5S.PDB
READ SEGID B5S PDB BUILD SETUP UNIT 10
IC PARA
IC BUILD

12. Dock a protein represented by rigid domain pep2 into a protein
represented
by a molecular map pep1 based on map interactions
EMAP DOCK FMAP GTMC MAPI PEP1 RIGI PEP2 -
NTRA 3 NROT 3 NCYC 10 NSTEP 1000 TEMP 300

test/c30test/emaptest.inp

```
*  emaptest.inp: Test case for EMAP module.  
*  T-cell Receptor Variable domain(PDB code: 1a7n) a,b chains are used  
*  Xiongwu Wu    July 8, 2002  
*
```

```
stream datadir.def
```

```
if ?emap .ne. 1 then  
  echo "Test NOT performed."  
  stop  
endif
```

```
if ?numnode .ne. 1 then  
  echo "Test NOT performed."  
  stop  
endif
```

```
bomlev -1
```

```
set z a7n
```

```
! Read in a7n molecule  
open read form unit 12 name @0@z.psf  
read psf card unit 12
```

```
open read unit 16 card name @0@z.crd  
read coor unit 16 card
```

```
! Save complex structure  
coor copy comp
```

```
!Generate a map object map1 from the molecule structure  
! with default resolution 15A  
emap gene map1 sele all end
```

```
!Output the map object to a map file in CCP4 format  
emap write map1 name "scratch/a7n.ccp4" format ccp4
```

```
!Create a map object map2 by read in from a map file in CCP4 format  
emap read map2 name "scratch/a7n.ccp4" format ccp4
```

```
!Create a map object by duplication  
emap dup1 mapid map1 to map3
```

```
!Mess up a-chain position  
coor orien sele segid a7na end  
coor tran sele segid a7na end xdir 1 ydir 4 zdir -5  
coor rota sele segid a7na end xdir -1 ydir 3 zdir -5 phi 120
```

```
!Mess up b-chain position
```

```

coor orien sele segid a7nb end
coor tran sele segid a7nb end xdir 1 ydir 4 zdir -5
coor rota sele segid a7nb end xdir -1 ydir 3 zdir -5 phi 120

!Generate a,b-chain map object mapa,mapb
emap gene mapa sele segi a7na end
emap gene mapb sele segi a7nb end

!Restore x-ray coordinates
coor copy

!Define rigid domains for a-chain and b-chain
emap assign mapa as riga sele segid a7na end
emap assign mapb as rigb sele segid a7nb end

!build map2 as riga + rigb
! initialize map2
emap init map2

!Add riga and rigb to map2
emap add riga to map2
emap add rigb to map2

!Update statistics of map2
emap stat map2

!Define complex components as riga and rigb for fitting
emap comp riga
emap comp rigb append

!Generate summation map from all complex components
emap sum map3

! Compare map1 and map2, they are almost identical with certain round
off error
! density correlation
emap corr mapid map1 mapid map2
! Laplacian correlation
emap corr mapid map1 mapid map2 DDR, very sensitive to round off error
! Core-weighted density correlation
emap corr mapid map1 mapid map2 CORE
! Core-weighted Laplacian correlation
emap corr mapid map1 mapid map2 DDR CORE

! Compare map1 and map3, they are almost identical with certain round
off error
emap corr mapid map1 mapid map3
emap corr mapid map1 mapid map3 DDR
emap corr mapid map1 mapid map3 CORE
emap corr mapid map1 mapid map3 DDR CORE

! Compare map1 and map3, they are identical
emap corr mapid map2 mapid map3
emap corr mapid map2 mapid map3 DDR
emap corr mapid map2 mapid map3 CORE
emap corr mapid map2 mapid map3 DDR CORE

```

```

! Save the current position
emap save riga
emap save rigb

!Translate riga and rotate rigb
emap tran riga xdir 1 ydir 4 zdir -5
emap rota riga xdir 1 ydir 3 zdir -5 phi 120
emap rota rigb xdir -1 ydir -4 zdir 5 phi -90

!Project to obtain the new coordinates
emap proj riga sele segi a7na end
emap proj rigb sele segi a7nb end

!Calculate the RMS deviations before fitting
coor rms sele segi a7na end
coor rms sele segi a7nb end
coor rms sele all end

! Compare the map with each rigid domain
emap corr mapid map1 rigid riga DDR CORE
emap corr mapid map1 rigid rigb DDR CORE

!Construct the complex structure according to map1 using single-body
gtmc method
!The components defined as components above will be docked to map1
! Core-weighted Laplacian correlation is recommended for single-body
fitting
emap dock mapid map1 gtmc ntran 2 nrot 2 -
ncyc 10 nstep 100 tran 15 rota 30 CORE DDR

!Generate summation map from all complex components
emap sum map3

! Compare map1 and map3, they are now very similar
emap corr mapid map1 mapid map3
emap corr mapid map1 mapid map3 DDR
emap corr mapid map1 mapid map3 CORE
emap corr mapid map1 mapid map3 DDR CORE

! Compare the fitting of each rigid domain
emap corr mapid map1 rigid riga DDR CORE
emap corr mapid map1 rigid rigb DDR CORE

!Project to obtain fitted coordinates
emap proj riga sele segi a7na end
emap proj rigb sele segi a7nb end

!Calculate the RMS deviation of the fitting result from x-ray structure
coor rms sele segi a7na end
coor rms sele segi a7nb end
coor rms sele all end

```

```

!Refine the complex structure using many-body gtmc method
! Density correlation can be used for many-body fitting
emap dock mapid map1 gtmc many -
LOOP 2 ntran 1 nrot 1 ncyc 10 nstep 100 tran 3.0 rota 5.0

!Generate summation map from all complex components
emap sum map3

! Compare map1 and map3, they are now very similar
emap corr mapid map1 mapid map3

!Project to obtain fitted coordinates
emap proj riga sele segi a7na end
emap proj rigb sele segi a7nb end

!Calculate the RMS deviation of the fitting result from x-ray structure
coor rms sele all end

!Write out the fitting coordinates
open writ form unit 16 name @9@z_fit.pdb
writ coor pdb unit 16 sele segid a7na .or. segid a7nb end

!Transfer the core of a map object to atomic representation
emap psf mapid map1 core skip 10

!Write out the atoms representing the map for display with molecule
viewers
open writ form unit 16 name @9@z_map.pdb
writ coor pdb unit 16 sele segi em* end

stop

```

Output from: test/c30test/emaptest.inp

```

1
          Chemistry at HARvard Macromolecular Mechanics
          (CHARMM) - Developmental Version 33a1      August 15,
2005
          Copyright(c) 1984-2001  President and Fellows of Harvard College
          All Rights Reserved
          Current operating system: IRIX64-6.5(IP35)@Tezrowu.lobos.nih.gov
          Created on 12/30/ 5 at 12: 9:59 by user: xiongwu

          Maximum number of ATOMS:      60120, and RESidues:
20040
          Current HEAP size:      10240000, and STACK size:
2000000

RDTITL> *  EMAPTEST.INP: TEST CASE FOR EMAP MODULE.

```

```
RDTITL> * T-CELL RECEPTOR VARIABLE DOMAIN(PDB CODE: 1A7N) A,B CHAINS  
ARE USED  
RDTITL> * XIONGWU WU JULY 8, 2002  
RDTITL> *
```

```
CHARMM>
```

```
CHARMM> stream datadir.def  
VOPEN> Attempting to open::datadir.def::  
OPNLGU> Unit 99 opened for READONLY access to datadir.def
```

```
INPUT STREAM SWITCHING TO UNIT 99  
RDTITL> * CHARMM TESTCASE DATA DIRECTORY ASSIGNMENT  
RDTITL> *  
Parameter: IN1 <- "" <empty>
```

```
CHARMM> faster on  
MISCOM> FAST option: EXPANDED (limited fast routines)
```

```
CHARMM> set 0 data/ ! input data directory  
Parameter: 0 <- "DATA/"
```

```
CHARMM> set 9 scratch/ ! scratch directory  
Parameter: 9 <- "SCRATCH/"
```

```
CHARMM> return  
VCLOSE: Closing unit 99 with status "KEEP"
```

```
RETURNING TO INPUT STREAM 5
```

```
CHARMM>
```

```
CHARMM> if ?emap .ne. 1 then  
RDCMND substituted energy or value "?EMAP" to "1"  
Comparing "1" and "1".  
IF test evaluated as false. Skip to ELSE or ENDIF
```

```
CHARMM>
```

```
CHARMM> if ?numnode .ne. 1 then  
RDCMND substituted energy or value "?NUMNODE" to "1"  
Comparing "1" and "1".  
IF test evaluated as false. Skip to ELSE or ENDIF
```

```
CHARMM>
```

```
CHARMM> bomlev -1
```

```
CHARMM>
```

```
CHARMM> set z a7n  
Parameter: Z <- "A7N"
```

```
CHARMM>
```

```
CHARMM> ! Read in a7n molecule  
CHARMM> open read form unit 12 name @0@z.psf
```

Parameter: 0 -> "DATA/"
Parameter: Z -> "A7N"
VOPEN> Attempting to open::data/a7n.psf::
OPNLGU> Unit 12 opened for READONLY access to data/a7n.psf

CHARMM> read psf card unit 12
MAINIO> Protein structure file being read from unit 12.
TITLE> * BUILD UP A PSF FROM PDB FILE
TITLE> * DATE: 8/20/ 1 1:59:14 CREATED BY USER: wxw
TITLE> *

Warning from PSFSUM: The sum of charges (-247.090000) is not an integer

***** LEVEL 0 WARNING FROM <PSFSUM> *****
***** Total charge not an integer

BOMLEV (-1) IS NOT REACHED. WRNLEV IS 5

PSFSUM> PSF modified: NONBOND lists and IMAGE atoms cleared.
PSFSUM> Summary of the structure file counters :
223 Number of segments = 2 Number of residues =
1038 Number of atoms = 1706 Number of groups =
0 Number of bonds = 1747 Number of angles =
299 Number of dihedrals = 0 Number of impropers =
0 Number of cross-terms = 0
0 Number of HB acceptors = 338 Number of HB donors =
0 Number of NB exclusions = 0 Total charge = -247.09000

CHARMM>

CHARMM> open read unit 16 card name @0@z.crd
Parameter: 0 -> "DATA/"
Parameter: Z -> "A7N"
VOPEN> Attempting to open::data/a7n.crd::
OPNLGU> Unit 16 opened for READONLY access to data/a7n.crd

CHARMM> read coor unit 16 card
SPATIAL COORDINATES BEING READ FROM UNIT 16
TITLE> * SIMULATION COORDINATE
TITLE> * DATE: 8/20/ 1 1:59:14 CREATED BY USER: WXW
TITLE> *

CHARMM>

CHARMM> ! Save complex structure
CHARMM> coor copy comp
SELECTED COORDINATES COPIED TO THE COMPARISON SET.

CHARMM>

```

CHARMM> !Generate a map object map1 from the molecule structure
CHARMM> ! with default resolution 15A
CHARMM> emap gene map1 sele all end
SELRPN> 1706 atoms have been selected out of 1706
Emap 1 is built from protein structure
MX,MY,MZ= -21 -8 -6
LX,LY,LZ= 39 36 38
DX,DY,DZ= 3.0000 3.0000 3.0000
1425 CORE points are found, RHOCUT= 0.000000

CHARMM>

CHARMM>

CHARMM> !Output the map object to a map file in CCP4 format
CHARMM> emap write map1 name "scratch/a7n.ccp4" format ccp4
emapwrite> NC = 39 (# columns)
emapwrite> NR = 36 (# rows)
emapwrite> NS = 38 (# sections)
emapwrite> MODE = 2 (data type: 2 = stored as floats)
emapwrite> NCSTART = -21 (index of first column)
emapwrite> NRSTART = -8 (index of first row)
emapwrite> NSSTART = -6 (index of first section)
emapwrite> NX = 39 (# of X intervals in unit cell)
emapwrite> NY = 36 (# of Y intervals in unit cell)
emapwrite> NZ = 38 (# of Z intervals in unit cell)
emapwrite> X length = 117.000 (unit cell dimension)
emapwrite> Y length = 108.000 (unit cell dimension)
emapwrite> Z length = 114.000 (unit cell dimension)
emapwrite> Alpha = 90.000 (unit cell angle)
emapwrite> Beta = 90.000 (unit cell angle)
emapwrite> Gamma = 90.000 (unit cell angle)
emapwrite> MAPC = 1 (columns axis: 1=X)
emapwrite> MAPR = 2 (rows axis: 2=Y)
emapwrite> MAPS = 3 (sections axis: 3=Z)
emapwrite> AMIN = 0.000 (minimum density value)
emapwrite> AMAX = 798.029 (maximum density value)
emapwrite> AMEAN = 17.144 (mean density value)
emapwrite> ISPG = 1 (ignored: space group number)
emapwrite> NSYMBT = 0 (ignored: # bytes used for storing
symmetry operators)
emapwrite> LSKFLG = 0 (flag for skew transformation: =0 none)
emapwrite> ARMS = 87.266 (standard deviation of density)
emapwrite> Write header information from CCP4 file scratch/a7n.ccp4

emapwrite> 53352 points are written for image grids

CHARMM>

CHARMM> !Create a map object map2 by read in from a map file in
CCP4 format
CHARMM> emap read map2 name "scratch/a7n.ccp4" format ccp4
emappeek> NC = 39 (# columns)
emappeek> NR = 36 (# rows)
emappeek> NS = 38 (# sections)
emapread> Reading header information from CCP4 file scratch/a7n.ccp4

```

```

emapread>      NC =      39  (# columns)
emapread>      NR =      36  (# rows)
emapread>      NS =      38  (# sections)
emapread>      MODE =       2  (data type: 2 = stored as floats)
emapread>      NCSTART =    -21  (index of first column)
emapread>      NRSTART =     -8  (index of first row)
emapread>      NSSTART =     -6  (index of first section)
emapread>      NX =      39  (# of X intervals in unit cell)
emapread>      NY =      36  (# of Y intervals in unit cell)
emapread>      NZ =      38  (# of Z intervals in unit cell)
emapread>      X length = 117.000  (unit cell dimension)
emapread>      Y length = 108.000  (unit cell dimension)
emapread>      Z length = 114.000  (unit cell dimension)
emapread>      Alpha =   90.000  (unit cell angle)
emapread>      Beta =   90.000  (unit cell angle)
emapread>      Gamma =   90.000  (unit cell angle)
emapread>      MAPC =       1  (columns axis: 1=X)
emapread>      MAPR =       2  (rows axis: 2=Y)
emapread>      MAPS =       3  (sections axis: 3=Z)
emapread>      AMIN =    0.000  (minimum density value)
emapread>      AMAX =  798.029  (maximum density value)
emapread>      AMEAN =   17.144  (mean density value)
emapread>      ISPG =       1  (ignored: space group number)
emapread>      NSYMBT =      0  (ignored: # bytes used for storing
symmetry operators)
emapread>      LSKFLG =      0  (flag for skew transformation: =0 none)
emapread>      ARMS =   87.266  (standard deviation of density)
emapread>      headlen =  256  (headlengthy)

```

```

emapread>      53352 points are read for image grids
      1425 CORE points are found, RHOCUT=  0.000000

```

CHARMM>

```

CHARMM>      !Create a map object by duplication
CHARMM>      emap dupl mapid map1 to map3
<EMAPDUP> Create New MAP ID:MAP3

```

CHARMM>

CHARMM>

```

CHARMM>      !Mess up a-chain position
CHARMM>      coor orien sele segid a7na end
SELRPN>      809 atoms have been selected out of  1706

```

ORIENT THE COORDINATES TO ALIGN WITH AXIS

MOMENTS

```

80809.05463520   2776.30134836 -12063.84491322
                28562.01599332   665.55924885
                30540.84366055

```

Transpose of the rotation matrix

```

0.220625   -0.046415   -0.974254
0.606086    0.789131    0.099656
0.764189   -0.612468    0.202234

```

CENTER OF ATOMS BEFORE TRANSLATION 4.84598 22.52847 41.26830
AXIS OF ROTATION IS 0.358079 0.874145 -0.328099 ANGLE IS 83.92

ALL COORDINATES ORIENTED IN THE MAIN SET BASED ON SELECTED ATOMS.

CHARMM> coor tran sele segid a7na end xdir 1 ydir 4 zdir -5
SELRPN> 809 atoms have been selected out of 1706
TRANSLATION VECTOR 1.000000 4.000000 -5.000000
SELECTED COORDINATES TRANSLATED IN THE MAIN SET.

CHARMM> coor rota sele segid a7na end xdir -1 ydir 3 zdir -5 phi
120
SELRPN> 809 atoms have been selected out of 1706
ROTATION MATRIX
-0.457143 -0.860496 -0.224869
0.603354 -0.114286 -0.789242
0.653441 -0.496472 0.571429

AXIS OF ROTATION IS -0.169031 0.507093 -0.845154 ANGLE IS 120.00

SELECTED COORDINATES ROTATED IN THE MAIN SET.

CHARMM>

CHARMM> !Mess up b-chain position
CHARMM> coor orien sele segid a7nb end
SELRPN> 897 atoms have been selected out of 1706

ORIENT THE COORDINATES TO ALIGN WITH AXIS

MOMENTS
35631.32146070 -6709.80192001 168.74923749
92412.96800683 7118.48701452
36946.08519911

Transpose of the rotation matrix

0.122549 0.986024 -0.112867
-0.867604 0.051214 -0.494612
-0.481918 0.158538 0.861754
CENTER OF ATOMS BEFORE TRANSLATION 8.63814 -8.12544 -18.10111
AXIS OF ROTATION IS -0.326626 -0.184555 0.926960 ANGLE IS 88.98

ALL COORDINATES ORIENTED IN THE MAIN SET BASED ON SELECTED ATOMS.

CHARMM> coor tran sele segid a7nb end xdir 1 ydir 4 zdir -5
SELRPN> 897 atoms have been selected out of 1706
TRANSLATION VECTOR 1.000000 4.000000 -5.000000
SELECTED COORDINATES TRANSLATED IN THE MAIN SET.

CHARMM> coor rota sele segid a7nb end xdir -1 ydir 3 zdir -5 phi
120
SELRPN> 897 atoms have been selected out of 1706

ROTATION MATRIX

```
-0.457143  -0.860496  -0.224869
 0.603354  -0.114286  -0.789242
 0.653441  -0.496472   0.571429
```

AXIS OF ROTATION IS -0.169031 0.507093 -0.845154 ANGLE IS 120.00

SELECTED COORDINATES ROTATED IN THE MAIN SET.

CHARMM>

```
CHARMM> !Generate a,b-chain map object mapa,mapb
CHARMM> emap gene mapa sele segi a7na end
SELRPN> 809 atoms have been selected out of 1706
Emap 4 is built from protein structure
MX,MY,MZ= -13 -14 -11
LX,LY,LZ= 34 32 36
DX,DY,DZ= 3.0000 3.0000 3.0000
738 CORE points are found, RHOCUT= 0.000000
```

```
CHARMM> emap gene mapb sele segi a7nb end
SELRPN> 897 atoms have been selected out of 1706
Emap 5 is built from protein structure
MX,MY,MZ= -18 -16 -19
LX,LY,LZ= 35 36 37
DX,DY,DZ= 3.0000 3.0000 3.0000
817 CORE points are found, RHOCUT= 0.000000
```

CHARMM>

```
CHARMM> !Restore x-ray coordinates
CHARMM> coor copy
SELECTED COORDINATES COPIED TO THE MAIN SET.
```

CHARMM>

```
CHARMM> !Define rigid domains for a-chain and b-chain
CHARMM> emap assign mapa as riga sele segid a7na end
SELRPN> 809 atoms have been selected out of 1706
```

```
CHARMM> emap assign mapb as rigb sele segid a7nb end
SELRPN> 897 atoms have been selected out of 1706
```

CHARMM>

```
CHARMM> !build map2 as riga + rigb
CHARMM> ! initialize map2
CHARMM> emap init map2
```

CHARMM>

```
CHARMM> !Add riga and rigb to map2
CHARMM> emap add riga to map2
```

```
CHARMM> emap add rigb to map2
```

```

CHARMM>

CHARMM>      !Update statistics of map2
CHARMM>      emap stat map2
Statistic properties updated for: MAP2

CHARMM>

CHARMM>      !Define complex components as riga and rigb for fitting
CHARMM>      emap comp riga
<EMAPCOMP> RIGA      added as a unfixed component.  Total:  1

CHARMM>      emap comp rigb append
<EMAPCOMP> RIGB      added as a unfixed component.  Total:  2

CHARMM>

CHARMM>      !Generate summation map from all complex components
CHARMM>      emap sum map3
      1425 CORE points are found, RHOCUT=  0.000000

CHARMM>

CHARMM>

CHARMM>      ! Compare map1 and map2, they are  almost identical with
certain round off error
CHARMM>      ! density correlation
CHARMM>      emap corr mapid map1  mapid map2
Corr: map MAP1      and map MAP2      IS  0.9929 at LDDR LCORE=  F
F

CHARMM>      ! Laplacian correlation
CHARMM>      emap corr mapid map1  mapid map2 DDR, very sensitive to
round off error
Corr: map MAP1      and map MAP2      IS  0.9596 at LDDR LCORE=  T
F
**** Warning ****  The following extraneous characters
were found while command processing in CHARMM
VERY SENSITIVE TO ROUND OFF ERROR

CHARMM>      ! Core-weighed density correlation
CHARMM>      emap corr mapid map1  mapid map2 CORE
Corr: map MAP1      and map MAP2      IS  0.9663 at LDDR LCORE=  F
T

CHARMM>      ! Core-weighed Laplacian correlation
CHARMM>      emap corr mapid map1  mapid map2 DDR CORE
Corr: map MAP1      and map MAP2      IS  0.8875 at LDDR LCORE=  T
T

CHARMM>

CHARMM>      ! Compare map1 and map3, they are  almost identical with
certain round off error
CHARMM>      emap corr mapid map1  mapid map3

```

```

Corr: map MAP1          and map MAP3          IS  0.9929 at LDDR LCORE=  F
F

CHARMM>   emap corr mapid map1  mapid map3 DDR
Corr: map MAP1          and map MAP3          IS  0.9596 at LDDR LCORE=  T
F

CHARMM>   emap corr mapid map1  mapid map3 CORE
Corr: map MAP1          and map MAP3          IS  0.9634 at LDDR LCORE=  F
T

CHARMM>   emap corr mapid map1  mapid map3 DDR CORE
Corr: map MAP1          and map MAP3          IS  0.8895 at LDDR LCORE=  T
T

CHARMM>

CHARMM>   ! Compare map1 and map3, they are  identical
CHARMM>   emap corr mapid map2  mapid map3
Corr: map MAP2          and map MAP3          IS  1.0000 at LDDR LCORE=  F
F

CHARMM>   emap corr mapid map2  mapid map3 DDR
Corr: map MAP2          and map MAP3          IS  1.0000 at LDDR LCORE=  T
F

CHARMM>   emap corr mapid map2  mapid map3 CORE
Corr: map MAP2          and map MAP3          IS  1.0000 at LDDR LCORE=  F
T

CHARMM>   emap corr mapid map2  mapid map3 DDR CORE
Corr: map MAP2          and map MAP3          IS  1.0000 at LDDR LCORE=  T
T

CHARMM>

CHARMM>   ! Save the current position
CHARMM>   emap save riga

CHARMM>   emap save rigb

CHARMM>

CHARMM>   !Translate riga and rotate rigb
CHARMM>   emap tran riga xdir 1 ydir 4 zdir -5

CHARMM>   emap rota riga xdir 1 ydir 3 zdir -5 phi 120

CHARMM>   emap rota rigb xdir -1 ydir -4 zdir 5 phi -90

CHARMM>

CHARMM>   !Project to obtain the new coordinates
CHARMM>   emap proj riga sele segi a7na end
SELRPN>   809 atoms have been selected out of  1706

CHARMM>   emap proj rigb sele segi a7nb end

```

```

SELRPN>      897 atoms have been selected out of   1706

CHARMM>

CHARMM>

CHARMM>      !Calculate the RMS deviations before fitting
CHARMM>      coor rms sele segi a7na end
SELRPN>      809 atoms have been selected out of   1706
TOTAL SQUARE DIFF IS 252356.4756 DENOMINATOR IS   809.0000
      THUS RMS DIFF IS   17.661719
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM>      coor rms sele segi a7nb end
SELRPN>      897 atoms have been selected out of   1706
TOTAL SQUARE DIFF IS 212128.0683 DENOMINATOR IS   897.0000
      THUS RMS DIFF IS   15.378106
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM>      coor rms sele all end
SELRPN>      1706 atoms have been selected out of   1706
TOTAL SQUARE DIFF IS 464484.5439 DENOMINATOR IS  1706.0000
      THUS RMS DIFF IS   16.500463
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM>

CHARMM>      ! Compare the map with each rigid domain
CHARMM>      emap corr mapid map1 rigid riga DDR CORE
Corr: map MAP1          and rigid RIGA          IS  0.3789 at LDDR LCORE=  T
T

CHARMM>      emap corr mapid map1 rigid rigb DDR CORE
Corr: map MAP1          and rigid RIGB          IS  0.2354 at LDDR LCORE=  T
T

CHARMM>

CHARMM>      !Construct the complex structure according to map1 using
single-body gtmc method
CHARMM>      !The components difined as components above will be docked
to map1
CHARMM>      ! Core-weighted Laplacian correlation is recommended for
single-body fitting
CHARMM>      emap dock mapid map1 gtmc ntran 2 nrot 2 -
CHARMM>      ncyc 10 nstep 100 tran 15 rota 30 CORE DDR
TARGET EMAP: 1  MAP1
      2 DOCK DOMAINS
RIGID DOMAIN    1  RIGA          FIXED:  F
RIGID DOMAIN    2  RIGB          FIXED:  F
NTGRID=  2 NRGRID=  2 NCYC=  10
NSTEP= 100 TEMP=  0.0100 RATIOI=  0.50 RATIOI=  0.50
LDDR=  T LCORE=  T
Fiting rigid:  1
TARGET EMAP: 6  SCRATCH
      1 DOCK DOMAINS
NTGRID=  2 NRGRID=  2 NCYC=  10

```

```

NSTEP= 100 TEMP= 0.0100 RATIOI= 0.50 RATIO= 0.50
LDDR= T LCORE= T
MC_1 ICYC CORRMAX CORRT RATT RATR DELTT
DELTR
0 0.005963 0.005963 0.500000 0.500000 15.000000
30.000000
1 0.212717 0.160094 0.130000 0.240000 15.000000
30.000000
2 0.212717 0.212717 0.000000 0.000000 14.250000
28.500000
3 0.212717 0.212717 0.000000 0.000000 13.537500
27.075000
4 0.212717 0.212717 0.000000 0.000000 12.860625
25.721250
5 0.212717 0.212717 0.000000 0.000000 12.217594
24.435187
6 0.212717 0.212717 0.000000 0.000000 11.606714
23.213428
7 0.212717 0.212717 0.000000 0.000000 11.026378
22.052757
8 0.212717 0.212104 0.060000 0.140000 10.475059
20.950119
9 0.224160 0.224160 0.080000 0.190000 9.951306
19.902613
10 0.245535 0.212287 0.030000 0.210000 9.453741
18.907482
GRID 1 1 1 1 1 1 -33.2 26.5 36.3 180.0 180.0 180.0
0.2455 0.3789
MC_1 ICYC CORRMAX CORRT RATT RATR DELTT
DELTR
0 0.128930 0.128930 0.500000 0.500000 15.000000
30.000000
1 0.586768 0.578861 0.030000 0.120000 15.000000
30.000000
2 0.602799 0.602799 0.000000 0.060000 14.250000
28.500000
3 0.608270 0.608270 0.000000 0.030000 13.537500
27.075000
4 0.608270 0.586918 0.000000 0.080000 12.860625
25.721250
5 0.608270 0.601210 0.000000 0.030000 12.217594
24.435187
6 0.608270 0.588947 0.000000 0.040000 11.606714
23.213428
7 0.613351 0.609159 0.000000 0.020000 11.026378
22.052757
8 0.637946 0.637946 0.010000 0.130000 10.475059
20.950119
9 0.648271 0.648271 0.010000 0.040000 9.951306
19.902613
10 0.685942 0.685942 0.010000 0.030000 9.453741
18.907482
GRID 1 1 1 1 1 2 -33.2 26.5 36.3 180.0 180.0 360.0
0.6859 0.6859
MC_1 ICYC CORRMAX CORRT RATT RATR DELTT
DELTR

```



```

          9      0.230928  0.199026  0.100000  0.200000  9.951306
19.902613
          10     0.230928  0.190171  0.130000  0.290000  9.453741
18.907482
  GRID  2  2  2  2  2  1  27.0  62.8  66.4  360.0  360.0  180.0
0.2309 0.8713
  MC_1  ICYC  CORRMAX  CORRT  RATT  RATR  DELTT
DELTR
          0     -2.000000 -2.000000  0.500000  0.500000 15.000000
30.000000
  GRID  2  2  2  2  2  2  27.0  62.8  66.4  360.0  360.0  360.0-
2.0000 0.8713
RIGID  2 was fitted with CORRT=  0.871281
DOWN WITH FITTING
      1437 CORE points are found, RHOCUT=  0.000000
AFTER GTMC FITTING, Final CORRMAX=  0.844446

```

CHARMM>

CHARMM>

```

CHARMM> !Generate summation map from all complex components
CHARMM>  emap sum map3
      1437 CORE points are found, RHOCUT=  0.000000

```

CHARMM>

```

CHARMM> ! Compare map1 and map3, they are now very similar
CHARMM>  emap corr mapid map1 mapid map3
Corr: map MAP1      and map MAP3      IS  0.9898 at LDDR LCORE=  F
F

```

```

CHARMM>  emap corr mapid map1 mapid map3 DDR
Corr: map MAP1      and map MAP3      IS  0.9408 at LDDR LCORE=  T
F

```

```

CHARMM>  emap corr mapid map1 mapid map3 CORE
Corr: map MAP1      and map MAP3      IS  0.9457 at LDDR LCORE=  F
T

```

```

CHARMM>  emap corr mapid map1 mapid map3 DDR CORE
Corr: map MAP1      and map MAP3      IS  0.8444 at LDDR LCORE=  T
T

```

CHARMM>

```

CHARMM> ! Compare the fitting of each rigid domain
CHARMM>  emap corr mapid map1 rigid riga DDR CORE
Corr: map MAP1      and rigid RIGA      IS  0.8507 at LDDR LCORE=  T
T

```

```

CHARMM>  emap corr mapid map1 rigid rigb DDR CORE
Corr: map MAP1      and rigid RIGB      IS  0.8697 at LDDR LCORE=  T
T

```

CHARMM>

```

CHARMM>

CHARMM> !Project to obtain fitted coordinates
CHARMM> emap proj riga sele segi a7na end
SELRPN> 809 atoms have been selected out of 1706

CHARMM> emap proj rigb sele segi a7nb end
SELRPN> 897 atoms have been selected out of 1706

CHARMM>

CHARMM> !Calculate the RMS deviation of the fitting result from x-
ray structure
CHARMM> coor rms sele segi a7na end
SELRPN> 809 atoms have been selected out of 1706
TOTAL SQUARE DIFF IS 1401.4298 DENOMINATOR IS 809.0000
THUS RMS DIFF IS 1.316168
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM> coor rms sele segi a7nb end
SELRPN> 897 atoms have been selected out of 1706
TOTAL SQUARE DIFF IS 352.7923 DENOMINATOR IS 897.0000
THUS RMS DIFF IS 0.627138
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM> coor rms sele all end
SELRPN> 1706 atoms have been selected out of 1706
TOTAL SQUARE DIFF IS 1754.2221 DENOMINATOR IS 1706.0000
THUS RMS DIFF IS 1.014035
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION

CHARMM>

CHARMM>

CHARMM> !Refine the complex structure using many-body gtmc method
CHARMM> ! Density correlation can be used for many-body fitting
CHARMM> emap dock mapid map1 gtmc many -
CHARMM> LOOP 2 ntran 1 nrot 1 ncy 10 nstep 100 tran 3.0 rota 5.0
TARGET EMAP: 1 MAP1
2 DOCK DOMAINS
DOMAIN 1 RIGA FIXED: F
DOMAIN 2 RIGB FIXED: F
NLOOP= 2 NTGRID= 1 NRGRID= 1 NCYC= 10
NSTEP= 100 TEMP= 0.0100 RATIOI= 0.50 RATIOI= 0.50
LDDR= F
Fiting rigid: 1
GRID 0 0 0 0 0 0 5.9 22.1 41.2 0.0 0.0 0.0
0.9898 0.0000
MC_N ICYC CORRMAX CORRT RATT RATR DELTT
DELTR
0 0.989178 0.989178 0.500000 0.500000 3.000000
5.000000
1 0.989897 0.989897 0.500000 0.920000 2.850000
5.250000
2 0.989897 0.989897 0.500000 0.920000 2.707500
5.512500

```

3	0.989897	0.989897	0.470000	0.860000	2.572125
5.788125					
4	0.989897	0.989897	0.470000	0.890000	2.443519
6.077531					
5	0.990866	0.990866	0.600000	0.900000	2.565695
6.381408					
6	0.990866	0.990866	0.490000	0.890000	2.437410
6.700478					
7	0.990866	0.990866	0.520000	0.910000	2.559280
7.035502					
8	0.990866	0.990866	0.530000	0.870000	2.687244
7.387277					
9	0.992158	0.992158	0.450000	0.900000	2.552882
7.756641					
10	0.992158	0.992158	0.410000	0.840000	2.425238
8.144473					

GRID 1 1 1 1 1 1 5.9 22.1 41.2 360.0 360.0 360.0
0.9922 0.9922

1420 CORE points are found, RHOCUT= 0.000000

At ILOOP= 1 RIGID 1 CORRT= 0.990094 0.990094

Fiting rigid: 2

GRID	0	0	0	0	0	-12.0	26.8	28.4	0.0	0.0	0.0
0.9901	0.9922										

MC_N	ICYC	CORRMAX	CORRT	RATT	RATR	DELTT
DELTR						
0		0.990987	0.990987	0.500000	0.500000	3.000000
5.000000						
1		0.990987	0.990987	0.350000	0.930000	2.850000
5.250000						
2		0.990987	0.990987	0.480000	0.860000	2.707500
5.512500						
3		0.991360	0.991360	0.490000	0.920000	2.572125
5.788125						
4		0.991360	0.991360	0.520000	0.860000	2.700731
6.077531						
5		0.991360	0.991360	0.490000	0.920000	2.565695
6.381408						
6		0.991360	0.991360	0.490000	0.910000	2.437410
6.700478						
7		0.991360	0.991360	0.500000	0.820000	2.315539
7.035502						
8		0.991360	0.991360	0.540000	0.890000	2.431316
7.387277						
9		0.991360	0.991360	0.580000	0.910000	2.552882
7.756641						
10		0.991360	0.991360	0.530000	0.870000	2.680526
8.144473						

GRID 1 1 1 1 1 1 -12.0 26.8 28.4 360.0 360.0 360.0
0.9914 0.9914

1419 CORE points are found, RHOCUT= 0.000000

At ILOOP= 1 RIGID 2 CORRT= 0.990163 0.990163

Fiting rigid: 1

GRID	0	0	0	0	0	4.1	22.9	41.2	0.0	0.0	0.0
0.9902	0.9914										

MC_N	ICYC	CORRMAX	CORRT	RATT	RATR	DELTT
DELTR						

```
0      0.991385  0.991385  0.500000  0.500000  3.000000
5.000000
1      0.991385  0.991385  0.420000  0.870000  2.850000
5.250000
```

(Skipped)

```
GRID  1  1  1  1  1  1  -12.0  26.8  28.4  360.0  360.0  360.0
0.9934 0.9934
```

```
1441 CORE points are found, RHOCUT= 0.000000
At ILOOP= 2 RIGID 2 CORRT= 0.991024 0.991024
DOWN WITH FITTING
```

```
1441 CORE points are found, RHOCUT= 0.000000
AFTER GTMC FITTING, CORRMAX= 0.993387
```

CHARMM>

CHARMM>

CHARMM> !Generate summation map from all complex components

CHARMM> emap sum map3

```
1441 CORE points are found, RHOCUT= 0.000000
```

CHARMM>

CHARMM> ! Compare map1 and map3, they are now very similar

CHARMM> emap corr map1 map3

```
Corr: map MAP1      and map MAP3      IS 0.9910 at LDDR LCORE=  F
F
```

CHARMM>

CHARMM> !Project to obtain fitted coordinates

CHARMM> emap proj riga sele segi a7na end

```
SELRPN> 809 atoms have been selected out of 1706
```

CHARMM> emap proj rigb sele segi a7nb end

```
SELRPN> 897 atoms have been selected out of 1706
```

CHARMM>

CHARMM> !Calculate the RMS deviation of the fitting result from x-ray structure

CHARMM> coor rms sele all end

```
SELRPN> 1706 atoms have been selected out of 1706
```

```
TOTAL SQUARE DIFF IS 2346.6423 DENOMINATOR IS 1706.0000
```

```
THUS RMS DIFF IS 1.172827
```

```
RMS FOUND FOR SELECTED COORDINATES WITHOUT ROTATION
```

CHARMM>

CHARMM> !Write out the fitting coordinates

CHARMM> open writ form unit 16 name @9@z_fit.pdb

```
Parameter: 9 -> "SCRATCH/"
```

```

Parameter: Z -> "A7N"
OPNLGU> Unit already open. The old file will be closed first.
VCLOSE: Closing unit 16 with status "KEEP"
VOPEN> Attempting to open::scratch/a7n_fit.pdb::
OPNLGU> Unit 16 opened for WRITE access to scratch/a7n_fit.pdb

CHARMM> writ coor pdb unit 16 sele segid a7na .or. segid a7nb end
RDTITL>
RDTITL> No title read.
SELRPN> 1706 atoms have been selected out of 1706

CHARMM>

CHARMM>

CHARMM> !Transfer the core of a map object to atomic representation
CHARMM> emap psf mapid map1 core skip 10
DMIN, DMAX, DCUT= 0.000000 0.000000 0.000000

Warning from PSFSUM: The sum of charges ( -247.090000) is not an
integer

***** LEVEL 0 WARNING FROM <PSFSUM> *****
***** Total charge not an integer
*****
BOMLEV ( -1) IS NOT REACHED. WRNLEV IS 5

PSFSUM> PSF modified: NONBOND lists and IMAGE atoms cleared.
PSFSUM> Summary of the structure file counters :
Number of segments = 3 Number of residues =
396
Number of atoms = 2407 Number of groups =
1211
Number of bonds = 1747 Number of angles =
0
Number of dihedrals = 0 Number of impropers =
299
Number of cross-terms = 0
Number of HB acceptors = 338 Number of HB donors =
0
Number of NB exclusions = 0 Total charge = -247.09000

CHARMM>

CHARMM> !Write out the atoms representing the map for display with
molecule viewers
CHARMM> open writ form unit 16 name @9@z_map.pdb
Parameter: 9 -> "SCRATCH/"
Parameter: Z -> "A7N"
OPNLGU> Unit already open. The old file will be closed first.
VCLOSE: Closing unit 16 with status "KEEP"
VOPEN> Attempting to open::scratch/a7n_map.pdb::
OPNLGU> Unit 16 opened for WRITE access to scratch/a7n_map.pdb

CHARMM> writ coor pdb unit 16 sele segi em* end
RDTITL>

```

RDTITL> No title read.
SELRPN> 700 atoms have been selected out of 2407
NOTE: A SELECTED SUBSET OF ATOMS WILL BE USED

CHARMM>

CHARMM>

CHARMM> stop

CHARMM c33a2 grid.doc

File: Grid |-[**Node:** Top

Up: ([commands.doc](#)) **-- Next:** [Implementation](#)

Grid: A general facility to implment grid-based potentials for docking

Charles L. Brooks III, TSRI. December 2000.

This document node describes the implementation, commands and syntax associated with an implementation of grid-based potentials to be used in ligand-docking studies, or when an additional set of potentials are to be added to augment. It can be used with dynamics as well as the GA/MC module.

* Menu:
* [Implementation](#):: A brief description of the anatomy of the module
* [Syntax](#):: Syntax of the commands
* [Description](#):: Description of key words and commands usage
* [Restrictions](#):: Restrictions on usage
* [Examples](#):: Supplementary examples of the use of the module

File: Grid |-[**Node:** Implementation

Up: ([commands.doc](#)) **-- Next:** [Syntax](#)

Grid-based potentials: Description and Discussion

CHARMM modules involved: misc/grid.src, fcm/grid.fcm, fcm/energy.fcm
energy/energy.src, energy/eutil.src, energy/intere.src,
energy/printe.src, charmm/iniall.src, charmm/charmm_main.src

This module provides code to 1) generate a set of van der Waals and electrostatic grid-based potentials and to 2) use these potentials in dynamics, minimization and GA/MC-based searching algorithms.

Generation of the grid-based van der Waals potentials is accomplished by establishing a series of vdW radius based potential surfaces over a limited spatial extent specified by the user. This set of potentials is built for radii of a series of test particles of unit epsilon parameter. The general idea is to use radii that span the range of radii used in the force field of interest, either on a discrete grid or at particular values. In utilizing these grids for energy and force calculations, the vdW radius of the atoms in the target molecule are mapped to the nearest probe radius, with a warning being given if the radii differ by more than 0.1 Å, and the overall energy is scaled by the square-root of the specific atoms vdW epsilon. This simplification provides a means to minimize the number of 3D grids that must be generated to represent the potential for a complex system. However, if memory is not an issue, in principle grid-based potentials may be generated for all vdW-based atom radii.

The electrostatic-based potential is the electrostatic potential associated with a test charge throughout the user specified space.

The potential energy is computed as a 3D, 8-point linear interpolation with forces computed from the analytic gradient of this interpolation formula. The potential energy and forces beyond the grid edges is constructed as a quadric potential away from the grid edge.

File: Grid, Node: Syntax, Up: Top, Previous: Top, Next: Description

Syntax for the Grid-based potentials

Generation:

```
grid generate select <atom selection> end -
  [xcen <real>] [ycen <real>] [zcen <real>] -
  [xmax <real>] [ymax <real>] [zmax <real>] -
  [dgrid <real>] [Force <real>] -
  [OutUnit <integer>] [Formatted] [Print]
```

Initialization:

```
grid read select <atom selection> end -
  Unit <integer> [Formatted] [Print]

grid on select <atom selection> end

grid off

grid clear
```

File: Grid |-| **Node:** Description
Previous: [Syntax](#) ==- **Up:** [Top](#) ==- **Next:** [Restrictions](#)

There are two basic parts to utilizing grid-based potentials in CHARMM:

Description of the basic key words for grid-based potentials:

The following is the description of the setup commands for setting up the system

Keyword/Syntax	Default	Purpose
GENERate calculating		Setting up the data structure and the potential grids.
READ potentials		Keyword to read a set of grid-based and set-up grid-based energy calculations.
ON		Keyword to activate grid-based potential calculations.
OFF		Keyword to de-activate grid-based potential calculations.
CLEAR		Keyword to clear all grid-based potential data structures from heap and stack.
XCEN potentials.	0.0 (A)	X-position for center of grid-based
YCEN potentials.	0.0 (A)	Y-position for center of grid-based
ZCEN potentials.	0.0 (A)	Z-position for center of grid-based
XMAX	0.0 (A)	X-direction extent of the potential grid.
YMAX	0.0 (A)	Y-direction extent of the potential grid.
ZMAX	0.0 (A)	Z-direction extent of the potential grid.
DGRId	0.5 (A)	Spacing between consecutive points in potential grids.
FORCe kcal/mol/A ² .	300.0	Force constant for quadratic extension of potential beyond grid edges, in
OUTUnit	Std Out	Unit to write grid-based potential file.
UNIT potential file.	Std Out	Unit from which to read grid-based
FORMatted based	.false.	Logical to set reading/writing of grid-potentials in ascii format.

PRINT potentials .false. Logical to set whether grid-based potentials will be printed to standard out.

File: Grid]-[Node: Restrictions

Up: [Top](#) -- Next: [Examples](#) -- Previous: [Description](#)

This module is in alpha release and subject to change. All aspects should work but this energy term has not been implemented in all other CHARMM modules, e.g., it cannot be used with the free energy modules pert, tsm or block, or with the MC module of Arron Dinner.

File: Grid]-[Node: Examples

Up: [Top](#) -- Next: [Top](#) -- Previous: [Restrictions](#)

Supplementary examples.

Generate and test grid for simple example of test atom.

probes.RTF:

```
* ...
*
  22    0
MASS  301 P1    1.00    P1 !
MASS  302 P2    1.00    P2 !
MASS  303 P3    1.00    P3 !
MASS  304 P4    1.00    P4 !
MASS  305 P5    1.00    P5 !
MASS  306 P6    1.00    P6 !
MASS  307 P7    1.00    P7 !
MASS  308 P8    1.00    P8 !
MASS  309 P9    1.00    P9 !
MASS  310 P10   1.00    P10 !
MASS  311 P11   1.00    P11 !
MASS  312 P12   1.00    P12 !
MASS  313 P13   1.00    P13 !
MASS  314 P14   1.00    P14 !
MASS  315 P15   1.00    P15 !
MASS  316 P16   1.00    P16 !
MASS  317 P17   1.00    P17 !
MASS  318 P18   1.00    P18 !
MASS  319 P19   1.00    P19 !
MASS  320 P20   1.00    P20 !
```

RESI PROB 20.000

```

ATOM P1 P1 1.0 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17
p18 p19 p20
ATOM P2 P2 1.0 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18
p19 p20
ATOM P3 P3 1.0 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18
p19 p20
ATOM P4 P4 1.0 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19
p20
ATOM P5 P5 1.0 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P6 P6 1.0 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P7 P7 1.0 p8 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P8 P8 1.0 p9 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P9 P9 1.0 p10 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P10 P10 1.0 p11 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P11 P11 1.0 p12 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P12 P12 1.0 p13 p14 p15 p16 p17 p18 p19 p20
ATOM P13 P13 1.0 p14 p15 p16 p17 p18 p19 p20
ATOM P14 P14 1.0 p15 p16 p17 p18 p19 p20
ATOM P15 P15 1.0 p16 p17 p18 p19 p20
ATOM P16 P16 1.0 p17 p18 p19 p20
ATOM P17 P17 1.0 p18 p19 p20
ATOM P18 P18 1.0 p19 p20
ATOM P19 P19 1.0 p20
ATOM P20 P20 1.0
END

```

probes.prm:

* Test probes for grid potential set-up

*

```

NBONDED NBXMOD 5 ATOM RDIEL SWITCH VATOM VDISTANCE VSWITCH -
CUTNB 999 CTOFNB 999 CTONNB 999 EPS 3 E14FAC 0.5 WMIN 1.5

```

```

!
!           EMIN           Rmin           These columns used for
!           (kcal/mol)      (A)           1-4 interactions
!
P1           0.00          -1.0000          0.65
P2           0.00          -1.0000          0.75
P3           0.00          -1.0000          0.85
P4           0.00          -1.0000          0.95
P5           0.00          -1.0000          1.05
P6           0.00          -1.0000          1.15
P7           0.00          -1.0000          1.25
P8           0.00          -1.0000          1.35
P9           0.00          -1.0000          1.45
P10          0.00          -1.0000          1.55
P11          0.00          -1.0000          1.75
P12          0.00          -1.0000          1.85
P13          0.00          -1.0000          1.95
P14          0.00          -1.0000          2.05
P15          0.00          -1.0000          2.15
P16          0.00          -1.0000          2.25
P17          0.00          -1.0000          2.35
P18          0.00          -1.0000          2.45
P19          0.00          -1.0000          2.55
P20          0.00          -1.0000          2.65

```

END

```
* GRIDTEST.INP
* This test-case demonstrates features of the grid-based potentials.
* It utilizes the MSI CHARMM (Momany & Rone) force field and the
* trypsin/benzamidine receptor/ligand pair.
* Required files: MASSES.RTF, probes.RTF, AMINO.RTF, PARM.PRM,
probes.prm
* 3ptb_complex.psf, 3ptb_complex.pdb
*
```

```
open unit 1 read card name "MASSES.RTF"
read rtf card unit 1
```

```
open unit 1 read card name "probes.RTF"
read rtf card unit 1 append
```

```
open unit 1 read card name "AMINO.RTF"
read rtf card unit 1 append
```

```
open unit 3 read card name "PARM.PRM"
read param card unit 3
```

```
open unit 1 read card name "probes.prm"
read param card unit 1 append
```

```
open unit 1 read form name "3ptb_complex.psf"
read psf card unit 1
```

```
open unit 1 read form name "3ptb_complex.pdb"
read coor pdb unit 1
```

```
! Find the center of the binding site
coor stat select resname ptb end
set xcen = ?xave
set ycen = ?yave
set zcen = ?zave
```

```
! Remove "real" ligand
delete atom select resname ptb end
```

```
! Generate test probe atoms
read sequ card
* title
*
1
prob
generate prob setup
```

```
! Delete all atoms but single representative for first grid test
delete atom select .not. ( type p15 .or. segid seg1 ) end
```

```
! Set-up position of test atom
scalar x set @xcen select segid prob end
```

```

scalar y set @ycen select segid prob end
scalar z set @zcen select segid prob end

! Fix receptor atoms
cons fix select segid seg1 end
energy

open unit 3 write form name grid.ascii
title
* Test grid for system
*

grid generate xmax 1 ymax 1 zmax 1 xcen @xcen ycen @ycen zcen @zcen -
      force 300 dgrid 0.5 select segid prob end outu 3 formatted print

grid clear

open unit 3 write unform name grid.bin
title
* Test grid for system
*

grid generate xmax 1 ymax 1 zmax 1 xcen @xcen ycen @ycen zcen @zcen -
      force 300 dgrid 0.5 select segid prob end outu 3 print

grid clear

open unit 3 read form name grid.ascii
grid read unit 3 formatted select type p15 end print
close unit 3

grid clear

open unit 3 read unform name grid.bin
grid read unit 3 select type p15 end print
close unit 3

! Generate positions on grid, vdW and elec should match grid terms
energy inbfrq 0
Calc Xmax = @Xcen + .5
Calc Ymax = @ycen + .5
Calc zmax = @zcen + .5
Calc Xmin = @Xcen - .5
Calc Ymin = @ycen - .5
Calc zmin = @zcen - .5

set x = @xmax
label ix
  set y = @ymax
  label iy
    set z = @zmax
    label iz

    scalar x set @x select type p15 end
    scalar y set @y select type p15 end
    scalar z set @z select type p15 end
    energy

```

```

        Calc dvdW = ( ?vdW - ?Grvd ) / ?vdw
        Calc delec = ( ?elec - ?Grel ) / ?elec
write title unit 12
* ?Grvd ?Grel ?vdW ?elec @dvdw @delec
*
        Calc z = @z - 0.5
        if z ge @zmin goto iz
        Calc y = @y - 0.5
        if y ge @ymin goto iy
        Calc x = @x - 0.5
        if x ge @xmin goto ix

! Test on/off components of grid energy terms
grid off
energy

grid on select type p15 end
energy

skipe all excl grvd grel
energy

! Generate energy curve along diagonal of cube to demonstrate
interpolation
! and extrapolation.

label dodiagonal
Calc xlow = @Xmin - 0.5
Calc x = @xmax+0.5
Calc y = @ymax+0.5
Calc z = @zmax+0.5
set cnt = 0
skipe all excl elec vdw grel grvd
label diagonal
    scalar x set @x select type p15 end
    scalar y set @y select type p15 end
    scalar z set @z select type p15 end
    energy
    incr cnt by 1

write title unit 13
* @cnt ?Grvd ?vdW ?Grel ?elec
*
    Calc z = @z - 0.1
    Calc y = @y - 0.1
    Calc x = @x - 0.1

if x ge @xlow goto diagonal

grid clear

stop

Example 2: An exploration of grid-based potential versus full molecular
potential for benzamidine-trypsin pair.

* GRID_2.INP

```

```

* This test-case demonstrates features of the grid-based potentials.
* It utilizes the MSI CHARMM (Momany & Rone) force field and the
* trypsin/benzamidine receptor/ligand pair.
* Required files: MASSES.RTF, probes.RTF, AMINO.RTF, PARM.PRM,
probes.prm
* 3ptb_complex.psf, 3ptb_complex.pdb
*

open unit 1 read card name "MASSES.RTF"
read rtf card unit 1

open unit 1 read card name "probes.RTF"
read rtf card unit 1 append

open unit 1 read card name "AMINO.RTF"
read rtf card unit 1 append

open unit 3 read card name "PARM.PRM"
read param card unit 3

open unit 1 read card name "probes.prm"
read param card unit 1 append

open unit 1 read form name "3ptb_complex.psf"
read psf card unit 1

open unit 1 read form name "3ptb_complex.pdb"
read coor pdb unit 1

! Define dimensions of volume for docking
coor stat select resname ptb end
set xcen = ?xave
set ycen = ?yave
set zcen = ?zave

! Set dimensions of grid as maximum extent of ligand + 4 A
Calc Xmax = ?xmax - ?xmin + 4
Calc Ymax = ?ymax - ?ymin + 4
Calc Zmax = ?zmax - ?zmin + 4
Let Xmax = Max @Xmax @Ymax
Let Xmax = Max @Xmax @Zmax

! If we have already generated the grid potentials go to final part.
! Uncomment after grid generation and run again.
!goto alreadygener

! Remove ligand and generate probe atoms.
delete atom select resname ptb end

read sequ card
* title
*
1
prob
generate prob setup

! Set positions for all probe atoms

```

```

scalar x set @xcen select segid prob end
scalar y set @ycen select segid prob end
scalar z set @zcen select segid prob end

! Fix position of receptor.
cons fix select segid seg1 end
skipe all excl vdw elec

energy

open unit 3 write uniform name grid_3ptb.bin
title
* Test grid for system
*

! Generate grid-based potentials for 20 probe atoms + electrostatic
! using default grid spacing of 0.5 A and default harmonic potential
! beyond grid edges (300 kcal/mol/A^2).
grid generate xmax @xmax ymax @xmax zmax @xmax -
             xcen @xcen ycen @ycen zcen @zcen -
             select segid prob end outu 3

grid clear

stop

! Begin here after grid potentials have been generated
label alreadygener

! Fiex receptor atoms for "rigid"-receptor docking
cons fix select segid seg1 end

! Read grid and set-up for ligand (seg2)
open unit 3 read uniform name grid_3ptb.bin
grid read unit 3 select segid seg2 end
close unit 3

! Randomly rotate ligand about its center and minimize
Calc phi = ?rand * 30

coor rota xdir @xcen ydir @ycen zdir @zcen phi @phi select segid seg2
end

! Turn off grid potential and minimize using "true" receptor.
grid off
energy inbfrq 1
coor copy compare
mini sd nstep 200 inbfrq 0
coor rms select segid seg2 end

! Turn on grid potential, restore coordinates of ligand and remove
receptor
! then minimize using grid-based potential only.
grid on select segid seg2 end
coor swap
coor translate xdir 10000 select segid seg1 end
energy inbfrq 1

```

```
mini sd nstep 200 inbfrq 0
```

```
! Check rmsd between ligand minimized in actual receptor and in grid-  
based
```

```
! receptor.
```

```
coor rms select segid seg2 end
```

```
stop
```



CHARMM c33a2 overlap.doc

File: Olap]-[**Node:** Top

Up: ([charmm.doc](#)) **Next:** [Description](#)

Overlap of Molecular Similarity

This is a maximum overlap method to investigate the structural similarity of flexible molecules. The atoms are described as Gaussians and the interaction energy between different molecules are basically overlap integrals. The Gaussians can represent either volume or charge. Alternatively, the overlap of the electrostatic potential is provided yielding exponential form.

This method supports all CHARMM functionality, because it provides just another energy term and forces for it. Only periodic boundaries and VIBRAN are not supported.

* Menu:

- * [Description](#):: Description of the OVERLAP commands.
- * [Usage](#):: How to use the OVERLAP method.
- * [Implementation](#):: Implementation of the OVERLAP method
- * [Performance](#):: Performance Issues

File: Olap]-[**Node:** Description

Up: [Top](#) **Next:** [Usage](#) **Previous:** [Top](#)

SYNTAX and DESCRIPTION

=====

One command (OLAP) is used in several different forms to specify everything.

To initialize the method use:

OverLAP NUMB <int> WEIGht <real> VOLW <real> CHAW <real> ESPW <real> -
WIDTh <real> GAMMa <real> WEPO <real>

NUMB <int> - how many subsystems do we have

WEIG <real> - weighting factor for the whole overlap term; it also
accounts to bring units to kcal/mol, default = 1.0

VOLW <real> - weighting factor for the volume overlap term,
default = 0.0

CHAW <real> - weighting factor for the partial atomic charge overlap
term, default = 0.0

ESPW <real> - weighting factor for the electrostatic potential
overlap term, default = 0.0

NOTE: Since all these three individual weighting factors default to 0.0,
the user has to specify at least one of them as a non-zero value, or
the
program will bomb out because there is no overlap to calculate!
The overall overlap Hodgkin index is calculated according to the
following
formula:

$$H(\text{total}) = \frac{\text{VOLW} * H(\text{volume}) + \text{CHAW} * H(\text{charge}) + \text{ESPW} * H(\text{e-s.pot})}{\text{VOLW} + \text{CHAW} + \text{ESPW}}$$

This way the overall Hodgkin index will be scaled between -1 and 1, no
matter what are the values of the individual weighting factors.

WIDT <real> - this value is used to scale all the atomic radii when
calculating volume or electrostatic potential overlap,
default = 1.0

GAMM <real> - gamma value for the electrostatic potential, default =
1.0

WEPO <real> - linear factor for the electrostatic potential,
default = 1.0

Before this initial OLAP command is called, WMAIN array should contain
partial atomic charges. In the course of initializing the overlap
subsystem, these charges will be copied from WMAIN to an internal array.
After the initialization, the user should load WMAIN with per-atom
weighting factors for the volume overlap (if the volume overlap is to
be
used at all). The most simple way to do this is via:

SCALAR WMAIN SET 1

which will give equal weighting of 1.0 to all atoms. Be aware of the
commands that could alter WMAIN array so that these weighting factors
are lost before calculating the overlap energy term!

After initialization, subsystems should be defined using the following command:

```
OLAP SYST <int> WEIG <real> SELE <selection factor> END
```

SYST <int> - the number of the subsystem being defined, should be in range from 1 to the number given in the initialization command (NUMB parameter)

WEIG <real> - weighting factor for the system being defined, default = 1.0

SELE ... END - selection of atoms which constitute this system.

The memory usage for these selections of subsystems is specified dynamically so there can be as many as one needs of these lines.

Do not forget to cancel all physical energy terms between subsystems treated with the OLAP! This can be done using BLOCK command. Here is an example for three subsystems:

```
BLOCK 3
CALL 2 SELE ... END
CALL 3 SELE ... END
COEF 1 2 0.0
COEF 1 3 0.0
COEF 2 3 0.0
END
```

[For more than several subsystems, there will be many ``COEF x y 0.0'' lines. This is something which may change, since specifying many block commands may cause users to make errors.

Possible solutions:

1. When generating nonbond list check the following:

```
if ((nolap(i).gt.0).and.(nolap(j).gt.0))then
  if (iolap(nolap(i)).ne.iolap(nolap(j))) then
    these 2 atoms have to be excluded.
  endif
endif
```

2. Or put the above in the exclusion list ??

3. or use block code - this works!

To check which atom is in which subsystem one can use:

```
OLAP PRINT
```

To print out individual forces and separate volume, charge and electrostatic potential Hodgkin indices use:

```
OLAP DEBUg - turn on debugging
```

OLAP NODEbug - turn off debugging

NOTE: This produces huge output! Therefore, it is not recommended to turn debugging on before a minimization or a dynamics run.

Weighting factors for the overlap terms (WEIG, VOLW, CHAW, ESPW) and factors determining the shape of Gaussian and exponential functions (WIDT, GAMMA, WEPO) can be changed via:

```
OLAP REStart WEIG <real> VOLW <real> CHAW <real> ESPW <real> -  
  WIDT <real> GAMMA <real> WEPO <real>
```

For the description of OLAP REST parameters, see above the section on initializing.

NOTE: When utilizing OLAP REST command, default values of all parameters are not the previous ones, but the general defaults (VOLW=0, CHAW=0, ESPW=0, WIDT=1, GAMMA=1, WEPO=1)! Therefore, the user has to specify all the non-default values again.

To turn off the overlap method completely, use:

```
OverLAP OFF
```

NOTE: This command also copies charges back to WMAIN!

File: Olap]-[Node: Usage

Up: [Top](#) -- Next: [Implementation](#) -- Previous: [Description](#)

USAGE
=====

Since everything is flexible, I suggest to start with aligning the systems to themselves first. With this approach one gets the estimate of the weights and radii which can be later used and improved in the alignment process of different species.

It is sometimes useful to exclude certain atoms from the alignment procedure. The obvious procedure to do this is to use SCALAR command and assign the WMAIN array to zero. This can be done both before OLAP initialization (thus setting atomic charges to zero and excluding them from the charge and electrostatic potential overlap) and after it (thus excluding atoms from the volume overlap).

File: Olap]-[Node: Implementation

Up: [Top](#) -- Next: [Performance](#) -- Previous: [Usage](#)

IMPLEMENTATION

=====

This is a new area of research, and the user might want to play with the different ``energy'' terms or formulas. The following is a guideline to do that. Everything CHARMM related is separated from the energy routines, so it should be easy for anyone to adjust the formulas for the systems under investigation.

Because in general we may have one atom in several systems we need to use the following data structure:

NOLAP(i), i=1, NATOM this is a vector of pointers to the IOLAP array.
IOLAP(i), i=1, NOLAP(NATOM) this is a vector which contains the information

to which subsystem each atom belongs to.

Then the loop for the overlap integrals would be like this:

```
do i = 1, natom
  do j = 1, natom
    do k = nolap(i), nolap(i+1)-1
      ix=iolap(k)
      if(ix.gt.0) then
        do l = nolap(j), nolap(j+1)-1
          jx=iolap(l)
          if(jx.gt.0) then
            ipt = (ix-1)*ix/2+jx ! this is not general
case
            s(ipt) = s(ipt) + gauss(i)*gauss(j)
          endif
        enddo
      endif
    enddo
  enddo
enddo
enddo
```

The above is simplified model for illustration purposes only. For details see the actual code. All the code for calculating overlap energies and forces is in energy/eolap.src; command-line analysis is in misc/olap.src. Also see fcm/olap.fcm.

The keyword to compile the method is ##OVERLAP.

File: Olap]-[Node: Performance

Up: [Top](#) -- Next: [Top](#) -- Previous: [Implementation](#)

PERFORMANCE ISSUES

=====

(since the systems are usually small this is not so big issue)

Very probably the method is trivial to parallelize. The following should take care of it:

In OLAPINT()

```
    icalc=0
    do i = 1, natom
      do j = 1, natom
        ....

                icalc=icalc+1
                if(mod(icalc,numnod).eq.mynod) then
                  ...
                  call fmgauss()
                  ...
                endif
        ....
      enddo
    enddo
```

This is a scheme for perfect load balance. However there is some loss in olapd, because it always does it for all atoms (it doesn't scale) This way there is no additional communication involved!!!?
