
ANATRA 1.0 User Guide

Kento Kasahara@Osaka University

2025 年 06 月 15 日

Contents:

第 1 章 ANATRA とは？	1
第 2 章 ANATRA のインストール	4
2.1 ANATRA を利用する上で必要となるプログラム	4
2.2 ソースコードからインストールする	4
第 3 章 anatra の使い方	7
3.1 はじめに	7
3.2 共通オプション	9
3.3 トラジェクトリ変換 (trjconv)	13
3.3.1 オプション一覧	13
3.3.2 解析例	16
3.3.2.1 トラジェクトリから水を除去	16
3.3.2.2 PBC ラップ	16
3.3.2.3 フィッティング	17
3.3.2.4 センタリング	17
3.3.2.5 PBC ラップ&フィッティング	18
3.3.2.6 PBC ラップ&センタリング	19
3.4 距離 (distance, dis)	19
3.4.1 オプション一覧	19
3.4.2 出力ファイルのフォーマット	21
3.4.3 解析例	22
3.4.3.1 通常の原子間距離の計算	22
3.5 重心解析・平均二乗変位 (center_of_mass, com)	22
3.5.1 理論背景	22
3.5.1.1 平均二乗変位	22
3.5.2 オプション一覧	23
3.5.3 出力ファイルのフォーマット	25
3.5.4 解析例	25
3.5.4.1 タンパク質の MSD 計算	25
3.5.4.2 複数個存在する小分子の MSD 計算	26
3.6 動径分布関数 (radial_distr, rdf)	26
3.6.1 理論背景	26
3.6.1.1 同種粒子間の動径分布関数	26
3.6.1.2 異種粒子間の動径分布関数	28
3.6.2 オプション一覧	29
3.6.3 出力ファイルのフォーマット	30
3.6.4 解析例	30

3.6.4.1	水分子の酸素原子間の RDF	30
3.6.4.2	水分子の酸素原子-水素原子間の RDF	31
3.6.5	タンパク質の C α 原子と水分子の酸素原子の間の RDF	31
3.7	空間分布関数 (spatial_distr, sdf)	32
3.7.1	理論背景	32
3.7.2	オプション一覧	33
3.7.3	出力ファイルのフォーマット	35
3.7.4	解析例	36
3.7.4.1	タンパク質周りの水分子の SDF	36
3.8	平均二乗偏差 (rmsd)	36
3.8.1	理論背景	36
3.8.2	オプション一覧	37
3.8.3	出力ファイルのフォーマット	38
3.8.4	解析例	38
3.8.4.1	水中におけるタンパク質の結晶構造からの RMSD	38
3.9	相互作用エネルギー (interaction_energy, ene)	39
3.9.1	オプション一覧	40
3.9.2	出力ファイルのフォーマット	43
3.9.3	解析例	43
3.9.3.1	溶質間の静電相互作用と LJ 相互作用の和	43
3.9.3.2	溶質間の LJ 相互作用	44
3.10	脂質膜秩序 (lipid_order, scd)	45
3.10.1	オプション一覧	45
3.10.2	出力ファイルのフォーマット	46
3.10.3	解析例	46
3.10.3.1	DPPC 脂質膜の S_{CD} 計算	46
3.11	界面法線方向 (z) に沿った分布関数 (z_profile, zprof)	46
3.11.1	オプション一覧	47
3.11.2	出力ファイルのフォーマット	48
3.11.3	解析例	48
3.11.3.1	脂質膜-水系における水分子の数密度分布	48
3.12	特定の方向 z に対する分子の配向 (z_orient, zori)	49
3.12.1	オプション一覧	50
3.12.2	出力ファイルのフォーマット	52
3.12.3	解析例	52
3.12.3.1	脂質二分子膜内における脂質分子の配向分布	52

第1章 ANATRA とは？

ANATRA (*Analyze Trajectories*) は分子動力学 (Molecular Dynamics, MD) シミュレーションのトラジェクトリを解析する Tcl/Fortran90 プログラム群である。

系の特定の原子・分子やタンパク質の特定のアミノ酸残基に注目して解析を行うことが多いが、その特定の部位を抜き出してくる汎用プログラムを書くのは容易ではない。というのも、研究レベルでは『残基番号 11 番目から 50 番目までの部分でかつ残基名が ALA でかつ重原子』のような複数の条件を同時に満たす原子群を取り出す必要があるが、これらの条件により生ずる集合の和や積を解釈するプログラムの作成は相当に大変だからである。一方で、世界中の MD ユーザが利用している VMD (*Visual Molecular Dynamics*) では、特定の部位を選択するための優れた機能 (Atomselection) が提供されている。例えば、上記の例を VMD の Atomselection の文法で書くと、

```
resid 11 to 50 and resname ALA and noh
```

であり、直感的にも理解しやすい。そこで **ANATRA** では、VMD の Atomselection を利用したプログラム構成にすることとした。従って、部位選択に関しては、Atomselection と全く同じ文法を使うことが可能であり、VMD ユーザーにとっては新しい部位選択の文法を覚える必要がない。

ANATRA は解析の種類に応じた多数の Tcl/Fortran プログラムから構成されているが、これらを一元的に管理する "anatra" という名前のプログラムを提供した。ユーザーは

```
$ anatra 解析モード -option1 -option2 ...
```

の形で様々な解析を実行できる。このプログラムでは、解析モードに応じた Tcl スクリプトを VMD 上から実行することで Atomselection 機能を用い、選択した部位の情報を Fortran プログラムに入力として与えることで特定の部位に特化した解析を行っているが、ユーザーはそのことを意識する必要はなく、一連の作業を一気に行える。

同時に、Fortran プログラム側では Atomselection に対応する実装がないために、ソースコードの複雑化を回避できており、ユーザー側が **ANATRA** で提供しているライブラリを元に新たな解析プログラムを作成することも容易である。¹

• Project Leader/Main Developer

- 笠原 健人 (大阪大学基礎工学研究科化学工学領域 松林研究室)

• Contributor

- 昌山 廉 (大阪大学基礎工学研究科化学工学領域 松林研究室)
- 沖田 和也 (大阪大学基礎工学研究科化学工学領域 松林研究室)

¹ 将来的には、開発者向けのマニュアルも整備したいと考えている。

- 松原 優弥 (大阪大学基礎工学研究科化学工学領域 松林研究室)
 - 丸山 優星 (大阪大学基礎工学研究科化学工学領域 松林研究室)
 - 岡部 涼 (大阪大学基礎工学研究科化学工学領域 松林研究室)
 - 山下 湧輝 (大阪大学基礎工学研究科化学工学領域 松林研究室)
 - 松林 伸幸 (大阪大学基礎工学研究科化学工学領域 松林研究室)
-

ANATRA は、**GNU General Public License version 2.0 (GPL v2.0)** の下、配布されている。

ANATRA

Copyright © 2025 Kento Kasahara (Univ. of Osaka)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

また、下記に示す外部ライブラリに依存して開発が進められており、パッケージ内に同梱されている。これらについては、下記に示すライセンスが適用される。

- **NetCDF**

Copyright 2025 Unidata

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

- **xdrfile-1.1.4**

Copyright © 2009-2014, Erik Lindahl & David van der Spoel

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

<https://ftp.gromacs.org/pub/contrib/>

- **xdf.F90**

This routine was originally developed by Wes Barnett and distributed under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

<https://github.com/wesbarnett/libgmxfort>

Modified version of the routine was developed by Kai-Min Tu.

<https://github.com/kmtu/xdrfort>

- **mt19937.f**

This routine was developed by Makoto Matsumoto and Takuji Nishimura and was distributed under the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

<https://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/VERSIONS/FORTRAN/fortran.html>

第2章 ANATRA のインストール

この章では、ANATRA のインストール方法について紹介する。

2.1 ANATRA を利用する上で必要となるプログラム

- VMD 1.7.3 or later
- Intel コンパイラ (Intel OneAPI)
- bash

ログインシェルとして、bash を用いていることを前提としている。¹

VMD については、

```
https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD
```

から無料でダウンロードできる。ただし、アカウント登録が必要である。²

2.2 ソースコードからインストールする

ここでは、自身でソースコードから ANATRA をインストールする方法を紹介する。最新版のソースコードは GitHub 上で管理されている。

```
https://github.com/kenkasa/anatra-release
```

まず、GitHub リポジトリからソースコードをダウンロードする。ANATRA をインストールしたいディレクトリを `/home/USER/software` とすると、まずこのディレクトリに移動ののち、`git clone` を実行する。

```
$ cd /home/USER/software
$ git clone https://github.com/kenkasa/vmd_scripts.git
$ ls
```

(次のページに続く)

¹ ただし、次節で述べるように、zsh などの場合についても、自身で rc ファイル (`~/.zshrc` など) を修正すれば、bash 以外をログインシェルに用いている場合でも利用可能である。

² VMD のインストール方法も簡単に述べておく。web サイトからダウンロードした `vmd-XXX.tar.gz` を解凍すると、`vmd-XXX` というディレクトリが生成されるので、そのディレクトリへ移動する。`configure` ファイルをテキストエディタで開き、ファイル上部にある `install_bin_dir=` に VMD をインストールしたいディレクトリを入力する。その後、Linux マシンの場合は `./configure LINUXAMD64` と実行した上で、`src` ディレクトリに移動し、`make install` を実行すると、`install_bin_dir=` で指定したディレクトリに vmd プログラムが配置されているはずである。このディレクトリに PATH を通しておけば、インストール完了である。

(前のページからの続き)

```
anatra

$ cd anatra
$ ls
README.md
install.sh
bin/
tcl/
f90/
docs/
```

install.sh を実行することで、ANATRA のインストールが開始されるが、過去に ANATRA のインストールを行ったことがある場合は、~/.bashrc に次の 3 行が含まれているので、あらかじめ削除しておく必要がある。ただし、\$ANATRA_PATH が今回インストールする ANATRA と同じディレクトリを指し示している場合は、そのままにしても良い(もし分からない場合は下記 3 行を削除すると良い)。

```
export ANATRA_PATH=...
export PATH=$ANATRA_PATH/bin:$PATH
export LD_LIBRARY_PATH=$ANATRA_PATH/f90/lib/external/netcdf/netcdf/lib:$LD_LIBRARY_PATH
```

bash 以外のログインシェルを利用している場合は、上記に対応するものを自身で rc ファイルに記入すれば良い。例えば、zsh の場合には、

```
export ANATRA_PATH=/path/to/vmd_scripts/anatra
export PATH=$ANATRA_PATH/bin:$PATH
export LD_LIBRARY_PATH=$ANATRA_PATH/f90/lib/external/netcdf/netcdf/lib
```

install.sh を実行する際、引数でコンパイラを指定することが出来る。

```
$ ./install.sh <gcc or intel>
```

デフォルトは intel である。Intel compiler が利用できる環境の場合は intel を指定することを推奨する。いくつかのプログラムは Intel の Math Kernel Library (MKL) に依拠しているため、gcc の場合には、それらのインストールがスキップされる。

Intel compiler を利用する場合は、

```
$ ./install.sh intel
```

のように実行する。スクリプトを実行すると、ビルドの進捗がログとして表示される。ビルド終了直前で下記のメッセージが表示されれば、全ての解析プログラムのビルドが正常終了したと考えて良い。

```
Installation of ANATRA fortran programs have been
succesfully finished!!
```


また、変数として\$ANATRA_PATHが定義されており、anatra コマンドに PATH が通っていれば、インストール成功である。

```
$ echo $ANATRA_PATH
/home/USER/software/anatra

$ which anatra
/home/USER/software/anatra/bin/anatra
```

第3章 anatra の使い方

3.1 はじめに

anatra は VMD と Fortran を用いて解析を行うプログラムである。この章では、**anatra** の使い方を解説する。

先にも述べたが、基本的にこのプログラムを用いる解析では、まず VMD を用いた分子の選択などの処理を通して中間ファイルを生成、それを Fortran にて解析するという二段階の処理を踏んでいる。

利用可能な解析は"anatra -h"を実行することで表示される。

```
$ anatra -h
Available analysis in ANATRA
trjconv          (trj)   : Convert trajectory
distance         (dis)   : Distance Analysis
center_of_mass   (com)   : Center-of-Mass Analysis
radial_distr     (rdf)   : Radial-Distribution Analysis
spatial_distr    (sdf)   : Spatial-Distribution Analysis
rmsd             (rmsd)  : Root-Mean-Square-Deviation Analysis
interaction_energy (ene)  : Interaction Energy Analysis
lipid_order      (scd)   : Lipid S_CD order-parameter Analysis
area_per_lipid   (apl)   : Area Per Lipid Analysis
z_profile        (zprof) : Z-profile Analysis
z_orient         (zori)  : Orientation Analysis along z
species_info     (spec)  : Get Species information
```

trjconv などが解析モードであり、

```
$ anatra trjconv -option1 ...
```

のように実行する。また括弧内に示されているのはそれぞれの解析モード名の省略版であり、例えば trjconv は、

```
$ anatra trj -option1 ...
```

と実行しても良い。

各解析モードでのオプションなどは"anatra 解析モード -h"で簡単な説明とサンプルが表示される。このヘルプメッセージだけで大体の使い方が分かるようになっているが、より詳細な説明が必要な場合にこのマニュアルを参照すると良いだろう。以下では、trjconv の場合を例として示しておく (VMD のデフォルト出力は省略している)。

```
$ anatra trjconv -h
```

```
=====
```

Trajectory Convert

```
=====
```

Usage:

```
anatra trjconv \
  -stype      <structure file type> \
  -sfile      <structure file name> \
  -tintype    <input trajectory file type> \
  -tin        <input trajectory file name> \
  -totype     <output trajectory file type> \
  -to         <output trajectory file name> \
  -beg        <first frame to be read> (default: 1) \
  -end        <last frame to be read> \
              (default: 0, corresponding to last frame)) \
  -selX       <X-th VMD selection> (X=0,1,2...) \
  -fit        <fit is performed or not (true or false)> \
  -centering  <centering is performed or not (true or false)> \
              (default: false) \
  -wrap       <wrap is performed or not (true or false)> \
              (default: false) \
  -wrapcenter <wrapping center (origin or com) (default: fragment)> \
  -wrapcomp   <how to wrap molecules (residue, segid, chain, or fragment)> \
              (residue or segid or chain or fragment or none) \
              (default: fragment) \
  -refpdb     <reference pdb file name> \
              (necessary if fit = true) \
  -outselid   <selection id for output molecules> \
  -fitselid   <selection id for fitting or centering> \
  -refselid   <selection id for reference>
```

Remark:

o If you specify fit = true & wrap = true, wrapcomp is automatically changed to 'com'

Example (fitting):

```
anatra trjconv \
  -stype      parm7 \
  -sfile      str.prmtop \
  -tintype    dcd \
  -tin        inp.dcd \
  -totype     dcd \
```

(次のページに続く)

(前のページからの続き)

```

-to          out.dcd          \
-beg         1                \
-end         150              \
-sel0        not water        \
-sel1        resid 1 to 275 and name CA \
-sel2        resid 1 to 275 and name CA \
-fit         true             \
-centering   false           \
-wrap        true            \
-wrapcenter  origin          \
-wrapcomp    fragment        \
-refpdb      ref.pdb         \
-outselid    0               \
-fitselid    1               \
-refselid    2               \

```

上記の例が示すように、**anatra** では多くのオプションを含んでいるため、**bash** スクリプトなどの中で実行すると良い。以下にスクリプト例を示しておく。

```

#!/bin/bash

anatra trjconv          \
  -stype   parm7        \
  -sfile   str.prmtop    \
  -tintype  dcd          \
  -tin      INP.dcd      \
  -totype   dcd          \
  -to       OUT.dcd      \
  -sel0     resid 1 to 20 \
  -outselid 0

```

3.2 共通オプション

anatra には多くの解析モードが存在するが、解析モード間で多くの共有オプションが存在する。まず、それらを一覧として以下に示しておく。その後に、各オプションの内容について解説する。

オプション名	機能	例
-stype	構造ファイルの種類を指定	-sfile parm7
-sfile	構造ファイル名を指定	-stype A.prmtop
-tintype	入力トラジェクトリの種類を指定	-tintype dcd
-tin	入力トラジェクトリ名を指定	-tin in.dcd
-flist_traj	入力トラジェクトリ一覧を記載したファイルを指定	-flist_traj list.txt
-totype	出力トラジェクトリの種類を指定	-totype dcd
-to	出力トラジェクトリ名を指定	-to out.dcd
-selx ($x = 0, 1, \dots$)	選択する部位を指定 (x を selid と呼ぶ)	-sel0 resid 1 to 20
-mode or -modex ($x = 0, 1, \dots$)	選択した部位をどのように分割するかを指定	-mode residue
-fhead	解析結果の出力ファイル名のヘッダー	-fhead out
-prep_only	解析の前処理のみを行うか指定	-prep_only true

• -stype <構造ファイルの種類>

- Default: N/A
- Example: -stype parm7

構造ファイル (structure file) を指定する。構造ファイルは原子種情報 (や原子間の結合情報) を格納しているファイルのことを指す。多くのタイプのトラジェクトリファイル (dcd や xtc など) では原子の座標情報のみが格納されており、解析を進めるためには構造ファイルが必要となる。対応している構造ファイルは、おおよそ VMD がサポートしている全ての構造ファイルの種類を指定することが可能である。代表的なものを以下に示す。

- **pdb**: PDB ファイル
- **gro**: GRO ファイル
- **psf**: CHARMM PSF ファイル
- **parm7**: Amber パラメータファイル (多くの場合拡張子は .prmtop)

最も完全な情報を含んでいるのは、CHARMM の場合には psf、Amber の場合には parm7 なので、できる限り、これらのファイルを指定しておくが良い。

• -sfile <構造ファイル>

- Default: N/A
- Example: -sfile complex.prmtop

構造ファイルの名前を指定する。

• -tintype <入力トラジェクトリファイルの種類>

- Default: dcd

- Example: `-tintype dcd`

入力トラジェクトリファイルの種類を指定する。現在、`trjconv`を除いて、指定できるのは `dcd` と `xtc` と `netcdf` に限定されている。`trjconv` では、VMD がサポートする全ての種類のトラジェクトリが指定可能である。

• `-tin` <入力トラジェクトリファイル>

- Default: N/A
- Example: `-tin INP.dcd`

入力トラジェクトリのファイル名を指定する。複数のトラジェクトリファイルを読み込ませる場合は、“`-tin run1.dcd run2.dcd`”のように並べて書く。またワイルドカード (*) を用いることも可能であり、“`-tin run*.dcd`”のように書く。次に示す `-flist_traj` を用いて入力のトラジェクトリを指定することも可能である。

• `-flist_traj` <テキストファイル>

- Default: N/A
- Example: `-flist_traj trajlist.txt`

入力トラジェクトリの一覧を記したテキストファイルを指定する。`-tin` と `-flist_traj` の両方が指定された場合には、`-flist_traj` が優先される。入力テキストファイルのフォーマットについては、次に示すように、入力のトラジェクトリ名が一行ずつ記されている必要がある。

```
traj1.dcd
traj2.dcd
traj3.dcd
```

• `-totype` <出力トラジェクトリファイルの種類>

- Default: `dcd`
- Example: `-totype dcd`

出力トラジェクトリファイルの種類を指定する。現在、`trjconv`を除いて、指定できるのは `dcd` と `xtc` に限定されている。`trjconv` では、VMD がサポートする全ての種類のトラジェクトリが指定可能である。

• `-to` <出力トラジェクトリファイル>

- Default: N/A
- Example: `-to OUT.dcd`

出力トラジェクトリのファイル名を指定する。

- **-sel x <部位選択>** ($x = 0, 1, \dots$)

- Default: N/A
- Example: `-sel0 resid 1 to 20 and name CA`

選択する部位を VMD の Atomselection の文法に従って指定する. `-sel x` の x のことを `selid` と呼ぶ. 多くの解析モードでは, `-sel x` で指定された部位の解析を行う. また, 解析モードによっては, どの `selid` を解析に使うかを明示的に指定することもある. (例えば, `trjconv` の `-outselid` など)

- **-mode x <residue or whole or atom>** ($x = 0, 1, \dots$)

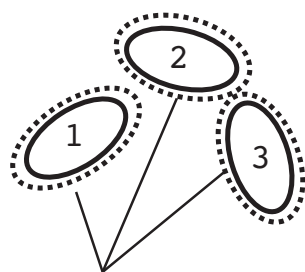
- Default: 解析モードに依存
- Example: `-mode0 residue`

`-sel x` で指定した部位をどのように分割して分子を定義するかを指定する (図 3.1). 解析によっては, `-sel0` の 1 つしか必要がない場合もあり, その解析では `-mode` で指定し, 複数の部位選択が必要な場合には, `-mode0`, `-mode1` のようにそれぞれで対応する分割スキームを指定する. どちらのオプションで指定するかは各解析モードでの解説にて述べる.

- **residue**: 選択部位を残基ごとに分割して分子と定義する.
- **whole**: 選択部位全体を一つの分子と定義する.
- **atom**: 選択部位の構成分子一つ一つを分子と定義する.

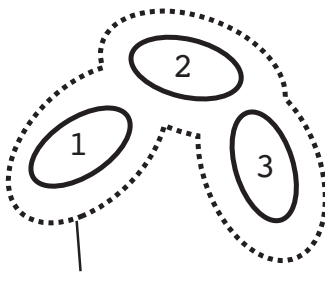
例えば, `-sel0 resname LIG` で複数のリガンド分子が選択されたとして, それぞれのリガンドの重心を知りたい場合には, `-mode residue` と設定する必要がある. 逆に, `-sel0 protein` と選択して, タンパク質一つを丸ごと解析対象としたい場合 (タンパク質の重心の時間変化など), `"-mode whole"` とする必要がある.

-mode residue



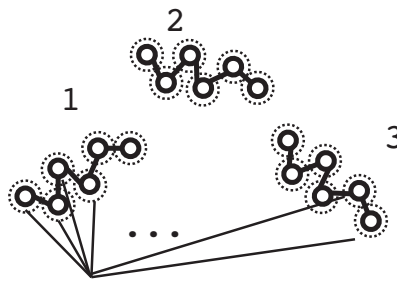
residue 毎に解析

-mode whole



複数の residue を
ひとまとめにして解析

-mode atom



原子毎に解析

図 3.1: `-mode` の使い分け. **residue** では, 選択した部位を残基ごとに分割して分子と定義する. **whole** では, 選択した部位をひとかたまりの分子として定義する. **atom** では, 選択した部位を構成原子ごとに分割して分子と定義する.

- **-fhead <出力ファイル名のヘッダー>**

- Default: 解析モードに依存
- Example: `-fhead out`

解析の出力結果を格納するファイル名のヘッダーを指定する。例えば, `-fhead out` と指定すると, `out.XXX`, `out.YYY` などのファイルが出力される。

- **-prep_only**

- Default: `false`
- Example: `-prep_only true`

実際の解析まで行わず, それに必要な前処理 (パラメータファイルの作成) までで止めるかを指定する。

3.3 トラジェクトリ変換 (trjconv)

`trjconv` は MD 計算で得られたトラジェクトリに対して, 種々の変換を行う解析モードである。主な用途として次のようなものが挙げられる。

- トラジェクトリのファイル形式を変更したい場合
- トラジェクトリ内の特定の部位のみを含むトラジェクトリを作成したい場合
(例. トラジェクトリ内の水を除いたトラジェクトリの作成)
- ラップされていないトラジェクトリを周期境界条件に基づいてラップしたい場合 (PBC ラップ)
- トラジェクトリの特定の部位を参照構造と重なるように系全体を並進回転させたい場合 (フィッティング)
- トラジェクトリの特定の部位の位置が座標原点となるように系全体を並進させたい場合 (センタリング)

3.3.1 オプション一覧

オプション名	備考	オプション名	備考
<code>-stype</code>	Common	<code>-sfile</code>	Common
<code>-tintype</code>	Common	<code>-tin</code>	Common
<code>-totype</code>	Common	<code>-to</code>	Common
<code>-selx</code>	Common	<code>-beg</code>	Mode specific
<code>-end</code>	Mode specific	<code>-fit</code>	Mode specific
<code>-centering</code>	Mode specific	<code>-wrap</code>	Mode specific
<code>-wrapcenter</code>	Mode specific	<code>-wrapcomp</code>	Mode specific
<code>-refpdb</code>	Mode specific	<code>-outselid</code>	Mode specific
<code>-fitselid</code>	Mode specific	<code>-refselid</code>	Mode specific

- **-beg <トラジェクトリの読み込み開始ステップ>**

- Default: 1
- Example: `-beg 10`

トラジェクトリの読み込み開始ステップを指定する。ただし、ステップ数は1始まりでカウントしていることに注意されたい。上記の例は、10 ステップ目からトラジェクトリの読み込みを開始することを意味している。

- **-end <トラジェクトリの読み込み終了ステップ>**

- Default: 0 (トラジェクトリの最終ステップに対応)
- Example: `-end 100`

トラジェクトリの読み込み終了ステップを指定する。ただし、ステップ数は1始まりでカウントしていることに注意されたい。上記の例は、100 ステップ目までトラジェクトリを読み込むことを意味している。

- **-fit <true or false>**

- Default: false
- Example: `-fit true`
- Note: true の場合は、`-fitselid`, `-refpdb`, `-refselid` の指定が必要となる。

フィッティングを行うかどうかを指定する。フィッティングとは、トラジェクトリの特定位点を参照構造に重なるように系全体の並進・回転を行う操作のことを意味する。利用する際、トラジェクトリ内のフィッティング部位に対応する `selid` (`-selx` の x のこと) を指定する `-fitselid`, 参照構造の PDB ファイルを指定する `-refpdb`, 参照構造内のフィッティングに使う部位の `selid` を指定する `-refselid` が必要となる。

- **-wrap <true or false>**

- Default: false
- Example: `-wrap true -wrapcomp fragment -wrapcenter origin`
- Note: ラップを行うときの中心を座標原点から変更したいときには、`-wrapcenter` を利用する。どのようなユニットを1つの分子として扱ってラップするかは `-wrapcomp` で指定する。

アンラップされた状態のトラジェクトリをラップするかどうかを指定する。ラップするときの中心は、デフォルトで座標原点 $(x, y, z) = (0, 0, 0)$ に設定されているが、`-wrapcenter` オプションを利用することで特定部位の重心に変更することが可能である。

- **-centering <true or false>**

- Default: false
- Example: `-centering true -sel1 resid 1 to 200 -fitselid 1`

- Note: 中心に並進移動させる部位の `selid` は `-fitselid` で指定する。

センタリングを行うかどうかを指定する。センタリングとは、トラジェクトリの特定の部位 (の重心) が座標原点に配置されるように系全体を並進移動させる操作のことを指す。どの部分を中心に移動させるかは `-fitselid` で指定する。

- **-refpdb <PDB file>**

- Default: N/A
- Example: `-refpdb A.pdb -sel2 resid 1 to 200 -refselid 2`
- Note: 参照構造内のフィッティングに使う部位の `selid` を `-refselid` で指定する。

フィッティングを行う際の、参照構造の PDB ファイル名を指定する。従って、このオプションは `-fit true` のときに必須となる。参照構造内のフィッティングに使う部位は `-refselid` で指定する。

- **-outselid <selid>**

- Default: N/A
- Example: `-sel0 resid 1 to 200 -outselid 0`

トラジェクトリとして出力する部位に対応する `selid` を指定する。

- **-fitselid <selid>**

- Default: N/A
- Example: `-sel1 resid 1 to 200 -fitselid 1`

フィッティングに使うトラジェクトリ内の部位に対応する `selid` を指定する。

- **-wrapcenter <origin or com>**

- Default: `origin`
- Example: `-wrapcenter origin`
- Note: `com` の場合は中心に据える部位の `selid` を `-fitselid` で指定する。

ラップする際の系の中心をどこに据えるかを指定する。

- **origin:** 系の中心として座標原点を指定する。
- **com:** 系の中心として特定の部位を指定する。中心に据える部位の `selid` を `-fitselid` で指定する。フィッティングまたはセンタリングを同時に行う場合、強制的に `com` が指定される。

- **-wrapcomp <fragment or residue or nocomp>**

- Default: `fragment`

- Example: `-wrapcomp fragment`

ラップを実行する際、どのようなユニットで分子を区切るかを指定する。例えば、シミュレーションボックスをはみ出した原子が存在する場合にその原子のみを PBC に基づいて反対側に移動させるようにするか、もしくは、分子全体がボックスからはみ出した場合にのみ反対側に移動させるかを指定する。その際、ユーザー側はどのような原子団 (ユニット) を分子として定義するかを指定する必要がある。

- **fragment**: 原子 i と j が他の原子との結合を介して繋がっている場合、それらは同じユニットに属していると考え、このような原子のセットを一つの分子としてラップが実行される。
 - **residue**: 一つの残基番号に属する原子群を一つの分子としてラップが実行される。
 - **nocomp**: 原子 1 つ 1 つを分子として扱ってラップが実行される。
-

3.3.2 解析例

ここでは、よく行う解析について、最低限のオプションを使ったスクリプトの例を示しておく。従って、`-beg`、`-end` などのオプションは使用しない。

3.3.2.1 トラジェクトリから水を除去

```
#!/bin/bash

anatra trjconv \
  -stype      parm7 \
  -sfile      complex.prmtop \
  -tintype    dcd \
  -tin        INP.dcd \
  -totype     dcd \
  -to         OUT.dcd \
  -sel0       not water \
  -outselid   0
```

3.3.2.2 PBC ラップ

```
#!/bin/bash

anatra trjconv \
  -stype      parm7 \
  -sfile      complex.prmtop \
```

(次のページに続く)

(前のページからの続き)

```

-tintype    dcd          \
-tin        INP.dcd      \
-totype     dcd          \
-to         OUT.dcd      \
-sel0       all          \
-wrap       true         \
-wrapcenter origin       \
-wrapcomp   fragment     \
-outselid   0

```

3.3.2.3 フィッティング

ここでは、水溶液中にタンパク質 (resid 1 to 192 とする) が一つ存在する系を考える。トラジェクトリ中のタンパク質の C α 原子を参照構造 PDB の C α 原子にフィットさせる場合の例を示す。

```

anatra trjconv          \
  -stype    parm7        \
  -sfile     complex.prmtop  \
  -tintype   dcd          \
  -tin       INP.dcd      \
  -totype    dcd          \
  -to        OUT.dcd      \
  -refpdb    REF.pdb      \
  -sel0      all          \
  -sel1      resid 1 to 192 and name CA \
  -sel2      resid 1 to 192 and name CA \
  -fit       true         \
  -outselid  0            \
  -fitselid  1            \
  -refselid  2

```

3.3.2.4 センタリング

ここでは、水溶液中にタンパク質 (resid 1 to 192 とする) が一つ存在する系を考える。トラジェクトリ中のタンパク質の重心を座標原点に配置させるセンタリングを行う場合の例を示す。

```

anatra trjconv          \
  -stype    parm7        \
  -sfile     complex.prmtop  \
  -tintype   dcd          \

```

(次のページに続く)

(前のページからの続き)

```

-tin      INP.dcd          \
-totype   dcd              \
-to       OUT.dcd         \
-sel0     all              \
-sel1     resid 1 to 192   \
-centering true           \
-outsclid 0                \
-fitsclid 1                \

```

3.3.2.5 PBC ラップ&フィッティング

ここでは、水溶液中にタンパク質 (resid 1 to 192 とする) が一つ存在する系を考える。トラジェクトリ中のタンパク質の C α 原子を参照構造 PDB の C α 原子にフィットさせるのと同時に、PBC ラップを実行する例を示す。この際、ラップ中心は C α 原子の重心である。

```

anatra trjconv          \
  -stype    parm7        \
  -sfile     complex.prmtop \
  -tintype   dcd          \
  -tin       INP.dcd      \
  -totype    dcd          \
  -to        OUT.dcd      \
  -refpdb    REF.pdb      \
  -sel0      all          \
  -sel1      resid 1 to 192 and name CA \
  -sel2      resid 1 to 192 and name CA \
  -fit       true         \
  -wrap      true         \
  -wrapcenter com         \
  -wrapcomp  fragment     \
  -outsclid  0            \
  -fitsclid  1            \
  -refselid  2            \

```

3.3.2.6 PBC ラップ&センタリング

中心に据える部分として、脂質膜 (resid 1 to 200 and segid MEMB とする) を設定した場合の例を示す。

```
anatra trjconv \
  -stype      psf \
  -sfile      complex.psf \
  -tintype    dcd \
  -tin        INP.dcd \
  -totype     dcd \
  -to         OUT.dcd \
  -sel0       all \
  -sel1       resid 1 to 200 and segid MEMB \
  -centering  false \
  -wrap       true \
  -wrapcenter com \
  -wrapcomp   fragment \
  -outselid   0 \
  -fitselid   1
```

3.4 距離 (distance, dis)

distance は選択した部位間の距離を計算する解析モードである。通常の距離計算の場合、選択した部位の分子重心間の距離が計算される。また、部位間の最近接距離 (minimum distance) 計算なども可能である。

3.4.1 オプション一覧

オプション名	備考	オプション名	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel0	Common	-sel1	Common
-mode0	Common	-mode1	Common
-pbc	Mode specific	-dt	Mode specific
-distance_type [†]	Mode specific	-mindist_type0 [†]	Mode specific
-mindist_type1 [†]	Mode specific	-prep_only	Common

通常の距離を計算する上で不要なオプションには[†]を付けているので、通常の距離を求めるだけであれば、それらのオプションは無視して良い。

- **-sel0 <部位選択>**
- **-sel1 <部位選択>**
 - Default: N/A
 - Example: `-sel0 resid 1 to 20 and noh -sel1 resname LIG`

この解析モードでは基本的に選択した二つの部位 (`selid=1, 2`) の間の距離を計算する。その対となる部位を `-sel x` ($x = 1, 2$) で指定する。

- **-mode0 <residue or whole or atom>**
 - **-mode1 <residue or whole or atom>**
 - Default: residue
 - Example: `-mode0 whole -mode1 residue`
- `-sel0`, `-sel1` で選択した部位をどのように分割するかを指定する。
-

- **-pbc <true or false>**
 - Default: false
 - Example: `-pbc true`
- 距離を計算する際に、PBC を考慮するかどうかを指定する。
-

- **-dt <時間刻み>**
 - Default: 1.0
 - Example: `-dt 0.1`
- 時系列データを出力するときの時間軸 (第 1 列) の時間刻みを指定する。
-

- **-distance_type <standard or minimum or intra>[†]**
 - Default: standard
 - Example: `-distance_type standard`
- 距離の種類を指定する。
- **standard**: 通常の距離計算を実行する。
 - **minimum**: 部位間の最近接の原子間距離を計算する。
 - **intra**: 分子内距離を計算する。
-

`intra` は他の 2 つに比べて少し複雑であるので、具体例を交えて使い道を説明する。いま、同種の高分子 (残基名 PLM) が 100 本存在する系を考える。また、両末端の原子の名前を CP1, CP2 とする。このような系で、それぞれの高分子の末端-末端間距離を計算することはよくある。しかし、`-distance_type standard` を選び、`-sel0 name CP1`, `-sel1 CP2` と指定すると、異なる高分子間の CP1-CP2 距離も計算してしまうことになる。`-distance_type intra` では、同一の分子内での距離のみを計算するようになっている。

- `-mindist_type1 <site or com>`[†]

- Default: `site`
- Example: `-mindist_type0 com -mindist_type1 site`

部位内の最近接サイト候補を指定する。

- **site**: 部位を構成する原子を最近接サイト候補に設定する。
- **com**: 部位の重心のみを最近接サイト候補に設定する。

部位 0, 1 がともに `com` の場合、通常の距離計算と等価になる。

3.4.2 出力ファイルのフォーマット

- `*.dis` ファイル

このファイルには、選択部位間の距離または最近接距離が出力される。`-mode0`, `-mode1` の指定により、部位 0, 1 がそれぞれ n_1 個, n_2 個の分子に分割されたとすると、合計で $n_1 n_2$ 個の距離が出力されることになる。そのときの各データは次のように出力される。

- 1 列目: 時間軸
- 2 列目: 分子対 (1, 1) の距離
- 3 列目: 分子対 (1, 2) の距離
- ...
- n_2 列目: 分子対 (1, n_2) の距離
- $n_2 + 1$ 列目: 分子対 (2, 1) の距離
- $n_2 + 2$ 列目: 分子対 (2, 2) の距離
- ...

距離は、Å 単位で出力される。

3.4.3 解析例

3.4.3.1 通常の原子間距離の計算

タンパク質 (resid 1 to 192) とリガンド 10 分子 (resname LIG) の間の距離を計算する場合を示す.

```
#!/bin/bash

anatra distance \
  -stype      parm7 \
  -sfile      complex.prmtop \
  -tintype     dcd \
  -tin        INP.dcd \
  -fhead      out \
  -pbc        true \
  -distance_type standard \
  -dt         0.1 \
  -sel0       resid 1 to 192 \
  -sel1       resname LIG \
  -mode0      whole \
  -mode1      residue
```

3.5 重心解析・平均二乗変位 (center_of_mass, com)

center_of_mass (com) は分子の重心の時間変化や平均二乗変位 (Mean square displacement, MSD) を求めるプログラムである. また, 脂質膜内部での分子の 2 次元拡散 (側方拡散) に対応するため, 2 次元 (xy 平面) での MSD 計算も可能である.

3.5.1 理論背景

3.5.1.1 平均二乗変位

平均二乗変位 (MSD) $\chi(t)$ の定義を導入すると同時に, それを求める意義について, 拡散方程式に基づいて述べる.

分子の時刻 t での位置を $\mathbf{r}(t)$ とすると, MSD は次式で定義される.

$$\chi(t) = \langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle. \quad (3.1)$$

つまり $\chi(t)$ は, 時間が t だけたった後における分子の移動距離の 2 乗平均を表す. MSD が拡散方程式に基づくと, どのように表されるのかを次に考えてみる. 分子の確率密度場を $\rho(\mathbf{r}, t)$, 拡散係数を D とすると, 拡散方程式は次式で表される.

$$\frac{\partial}{\partial t} \rho(\mathbf{r}, t) = D \nabla^2 \rho(\mathbf{r}, t). \quad (3.2)$$

時刻 $t = 0$ で分子は位置 $\mathbf{0}$ に存在していたとすると、初期条件は

$$\rho(\mathbf{r}, t = 0) = \delta(\mathbf{r}), \quad (3.3)$$

となる。この条件の下、拡散方程式を解くと、次式が得られる。

$$\rho(\mathbf{r}, t) = \frac{1}{(4\pi Dt)^{3/2}} \exp\left[-\frac{|\mathbf{r}|^2}{4Dt}\right]. \quad (3.4)$$

この解を用いると、MSD は

$$\chi(t) = \int d\mathbf{r} |\mathbf{r}|^2 \rho(\mathbf{r}, t) \quad (3.5)$$

$$= 6Dt, \quad (3.6)$$

と表される。つまり、MSD は傾き $6D$ で時間に比例して増加していくことになる。ただし、拡散方程式は長時間極限において成り立つ関係式なので、

$$D = \frac{1}{6} \lim_{t \rightarrow \infty} \frac{\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \rangle}{t}, \quad (3.7)$$

とすべきである。上式は Einstein の関係式と呼ばれる。この式は、MSD の長時間領域の傾きから拡散係数が求められることを意味している。

3.5.2 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel0	Common	-mode	Common
-outmsd	Mode specific	-unwarp	Mode specific
-msddim	Mode specific	-dt	Mode specific
-t_sparse	Mode specific	-t_range	Mode specific
-prep_only	Common		

- **-outmsd <true or false>**

- Default: false
- Example: -outmsd true
- Note: true の場合は -msddim と -msdrange も指定する必要がある。

MSD を計算するかどうかを指定する。

- **-unwrap <true or false>**

- Default: `false`
- Example: `-unwrap false`

ラップされているトラジェクトリをアンラップするかどうかを指定する。ラップされているトラジェクトリでは MSD が正しく計算できないため、`true` にする必要がある。ただし、GROMACS のように、分子が PBC によりぶつ切りになっているトラジェクトリには対応できないので、注意されたい。その場合はあらかじめ、GROMACS の `trjconv` にて修正しておく必要がある。

- **-msddim <2 or 3>**

- Default: `3`
- Example: `-msddim 3`

MSD の次元を指定する。通常の拡散では `3` を指定するが、脂質膜系での側方拡散を調べる場合には `2` を指定する。

- **-dt <時間刻み>**

- Default: `0.1`
- Example: `-dt 0.1`

スナップショット間の時間刻みを指定する。

- **-t_range <MSD を計算する時間スケール>**

- Default: `10`
- Example: `-t_range 10`

MSD を計算する時間スケールを指定する。

- **-t_sparse <MSD を出力するときの時間刻み>**

MSD を出力するときの時間刻みを指定する。例えば、`-dt 0.1 (= δt)`, `-t_sparse 1 (= Δt)` のとき、

$$\Delta T = \text{nint} \left(\frac{\Delta t}{\delta t} \right) \delta t \quad (3.8)$$

の時間刻みで MSD (\AA^2) が計算される。

3.5.3 出力ファイルのフォーマット

- *.com

このファイルには、選択部位内の分子それぞれの重心の座標が xyz 形式で出力される。

- *.msd

このファイルには、分子ごとの MSD が出力されている。"-outmsd true"の場合に出力される。

- 1 列目: 時間軸
- 2 列目: 分子 1 の MSD
- 3 列目: 分子 2 の MSD
- ...

- *.msdave

このファイルには、全分子に対して平均をとった MSD が出力されている。"-outmsd true"の場合に出力される。

- 1 列目: 時間軸
- 2 列目: MSD
- 3 列目: MSD の標準偏差

3.5.4 解析例

3.5.4.1 タンパク質の MSD 計算

水溶液中にタンパク質 (resid 1 to 192) が 1 つ溶解している系において、タンパク質の MSD を計算する例を以下に示す。タンパク質は複数のアミノ酸残基から構成されているので、"-mode whole"を指定する必要がある。

```
#!/bin/bash

anatra center_of_mass \
  -stype      parm7      \
  -sfile      complex.prmtop \
  -tintype    dcd        \
  -tin        INP.dcd    \
  -fhead      out        \
  -outmsd     true       \
  -sel0       resid 1 to 192 \
  -mode       whole      \
  -msddim     3          \
  -dt         0.01       \
```

(次のページに続く)

```
-t_range 100 \
-t_sparse 0.1
```

3.5.4.2 複数個存在する小分子の MSD 計算

水溶液中にリガンド小分子 (resname LIG) が 100 分子存在する系を考える。分子ごとの MSD とその平均を計算する例を以下に示す。この場合は残基番号によって分子の切れ目を定めることができるので、"-mode residue" を指定する。

```
#!/bin/bash

anatra center_of_mass \
  -stype parm7 \
  -sfile complex.prmtop \
  -tintype dcd \
  -tin INP.dcd \
  -fhead out \
  -outmsd true \
  -sel0 resname LIG \
  -mode residue \
  -msddim 3 \
  -dt 0.01 \
  -t_range 100 \
  -t_sparse 0.01
```

3.6 動径分布関数 (radial_distr, rdf)

3.6.1 理論背景

ここでは、動径分布関数の定義について解説する。

3.6.1.1 同種粒子間の動径分布関数

粒子 a が N_a 個存在している系 (体積 V ，粒子 a の平均 (バルク) 数密度 $\rho_a = N_a/V$) を考え、局所密度場 $\hat{\rho}_a(\mathbf{r})$ を次式で定義する。

$$\hat{\rho}_a(\mathbf{r}) = \sum_{i=1}^{N_a} \delta(\mathbf{r} - \mathbf{r}_i). \quad (3.9)$$

ここで、 \mathbf{r}_i は i 番目の粒子 a の位置である。もし、系に外場などが存在しない場合、局所密度場のアンサンブル平均は平均数密度 ρ_a に等しくなる。つまり、

$$\langle \hat{\rho}_a(\mathbf{r}) \rangle = \rho_a, \quad (3.10)$$

である。次に、2 体密度相関関数を次式で定義する。

$$\chi(\mathbf{r}, \mathbf{r}') = \langle \hat{\rho}_a(\mathbf{r}) \hat{\rho}_a(\mathbf{r}') \rangle \quad (3.11)$$

$$= \left\langle \sum_{i=1}^{N_a} \sum_{j=1}^{N_a} \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_j) \right\rangle. \quad (3.12)$$

上式において、 $i = j$ の項と $i \neq j$ の分けて書くと、

$$\chi(\mathbf{r}, \mathbf{r}') = \sum_{i=1}^{N_a} \langle \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_i) \rangle + \sum_{i=1}^{N_a} \sum_{j \neq i}^{N_a} \langle \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_j) \rangle, \quad (3.13)$$

第 1 項目は同一粒子が \mathbf{r} と \mathbf{r}' の位置に存在する確率を表しており、これは $\mathbf{r} = \mathbf{r}'$ のとき以外は明らかにゼロとなる。つまり、

$$\sum_{i=1}^{N_a} \langle \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_i) \rangle = \rho_a \delta(\mathbf{r} - \mathbf{r}'), \quad (3.14)$$

である。次に、第 2 項は i 番目の粒子が \mathbf{r} の位置に存在し、 j 番目の粒子が \mathbf{r}' の位置に存在する確率を表している。この項は 2 体密度と呼ばれており、2 粒子の相関関係を表している。2 体密度を $\rho_{aa}^{(2)}(\mathbf{r}, \mathbf{r}')$ と表しておくことにする。

$$\rho_{aa}^{(2)}(\mathbf{r}, \mathbf{r}') = \sum_{i=1}^{N_a} \sum_{j \neq i}^{N_a} \langle \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_j) \rangle. \quad (3.15)$$

もし、 \mathbf{r} と \mathbf{r}' の位置が互いに遠く離れている場合、そのような配置での 2 粒子間の相関は無視できる。

$$\rho_{aa}^{(2)}(\mathbf{r}, \mathbf{r}') \approx \sum_{i=1}^{N_a} \langle \delta(\mathbf{r} - \mathbf{r}_i) \rangle \sum_{j \neq i}^{N_b} \langle \delta(\mathbf{r}' - \mathbf{r}_j) \rangle \quad (3.16)$$

$$= \rho_a^2 \left(1 - \frac{1}{N_a} \right). \quad (3.17)$$

無相関の状態を基準にとって、2 体分布関数を次式で定義することにしておく。

$$g_{aa}^{(2)}(\mathbf{r}, \mathbf{r}') = \frac{1}{\rho_a^2 (1 - 1/N_a)} \rho_{aa}^{(2)}(\mathbf{r}, \mathbf{r}'). \quad (3.18)$$

もし、粒子を十分に多く含んでいる系を取り扱っている場合は、 $1/N_a \simeq 0$ と考えて、

$$g_{aa}^{(2)}(\mathbf{r}, \mathbf{r}') = \frac{1}{\rho_a^2} \rho_{aa}^{(2)}(\mathbf{r}, \mathbf{r}'), \quad (3.19)$$

としても良い。

系が均一の場合 (外場が存在しない場合など)、2 体分布関数 (および 2 体密度) は 2 粒子の絶対的な位置 \mathbf{r} , \mathbf{r}' ではなく、2 粒子の相対的な位置 $\mathbf{r}' - \mathbf{r}$ に依存する。つまり、

$$g_{aa}^{(2)}(\mathbf{r}, \mathbf{r}') = g_{aa}^{(2)}(\mathbf{0}, \mathbf{r}' - \mathbf{r}), \quad (3.20)$$

のような関係が成り立つ。 $\mathbf{r}' - \mathbf{r}$ を改めて \mathbf{r} とおき直しておき、空間分布関数 (spatial distribution function, SDF) を次式で定義する。

$$g_{aa}(\mathbf{r}) = g_{aa}^{(2)}(\mathbf{0}, \mathbf{r}). \quad (3.21)$$

これを角度方向に平均をとり、距離 r の分布関数に変換したものを動径分布関数 (radial distribution function, RDF) と呼ぶ。

$$g_{aa}(r) = \frac{1}{4\pi r^2} \int d\mathbf{r} \delta(|\mathbf{r}| - r) g_{aa}(\mathbf{r}). \quad (3.22)$$

もし、系が等方的である場合、

$$g_{aa}(|\mathbf{r}|) = g_{aa}(\mathbf{r}), \quad (3.23)$$

が成り立つ。SDF を計算することに意義が生じるのは、非等方系の場合や、一方の分子 (形状を持つ粒子) を特定の配向に固定した座標系で他方の粒子の位置を分布を調べるときである。これについては次節で述べる。

RDF は、一方の粒子から眺めて距離 r での他粒子の存在確率がバルクよりもどの程度高いかを表す。つまり、 $g_{aa}(r) > 0$ であれば、その距離で他粒子の存在確率はバルクよりも高く、 $g_{aa}(r) < 0$ の場合はバルクよりも存在確率が低いことを意味する。

3.6.1.2 異種粒子間の動径分布関数

粒子 a が N_a 個、粒子 b が N_b 個存在している系 (体積 V 、平均数密度はそれぞれの粒子で $\rho_a = N_a/V$, $\rho_b = N_b/V$) での、粒子 a と粒子 b の間の RDF を導入する。

$$\rho_{ab}^{(2)}(\mathbf{r}, \mathbf{r}') = \sum_{i=1}^{N_a} \sum_{j=1}^{N_b} \langle \delta(\mathbf{r} - \mathbf{r}_i^a) \delta(\mathbf{r}' - \mathbf{r}_j^b) \rangle. \quad (3.24)$$

ここで、 \mathbf{r}_i^p ($p = a, b$) は粒子 p の i 番目の粒子の位置である。同種粒子間の場合と異なるのは、 $j = i$ の場合を除外しなくても良い点である。これにより、無相関の場合は、

$$\rho_{ab}^{(2)}(\mathbf{r}, \mathbf{r}') \approx \rho_a \rho_b, \quad (3.25)$$

となる。それ以外の点は全く同じなので、同様の手続きを踏んで動径分布関数を導入できる。まず、空間分布関数は、

$$g_{ab}(\mathbf{r}) = \frac{1}{\rho_a \rho_b} \rho_{ab}^{(2)}(\mathbf{0}, \mathbf{r}), \quad (3.26)$$

であり、これを角度方向について平均化することにより、動径分布関数の表式を得る。

$$g_{ab}(r) = \frac{1}{4\pi r^2} \int d\mathbf{r} \delta(|\mathbf{r}| - r) g_{ab}(\mathbf{r}). \quad (3.27)$$

3.6.2 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel0	Common	-sel1	Common
-mode0	Common	-mode1	Common
-dr	Mode specific	-identical	Mode specific
-normalize	Mode specific	-separate_self	Mode specific

- **-dr <距離の刻み (Å)>**

- Default: 0.1

RDF を計算するときの距離の刻みを指定する.

- **-identical <true or false>**

- Default: false

-sel0 と -sel1 が同一の場合に **true** を指定する. 理論の解説で述べたように, RDF の計算は同種粒子間と異種粒子間で扱いが異なるので, ユーザー側でどちらの扱いにするかを指定する必要がある.

- **-normalize <true or false>**

- Default: true

RDF ($g_{12}(r)$) をそのまま出力するか (**true**), -sel2 側の部位の平均数密度 ρ_2 をかけたものを出力するか (**false**) を指定する. **true** の場合は, $\rho_{12}(r) = \rho_2 g_{12}(r)$ が出力される.

- **-separate_self <true or false>**

- Default: false

RDF を分子内と分子間の部分に分けるかどうかを指定する. このオプションが必要となるのは, 同種分子の異なる部位間の RDF を計算するときである. 例えば, 水分子 (原子ラベルを HW1-OW-HW2 とする) の OW 原子と HW1 原子の間の RDF を計算するときに, "-separate_self false" の条件で "-sel0 name OW", "-sel1 name HW1" と指定すると, RDF の中に同一分子内の OW-HW1 の寄与も含まれてしまうが, "-separate_self true" と指定することで, 同一分子内の寄与を分離することが可能である.

3.6.3 出力ファイルのフォーマット

- *.rdf ファイル

このファイルには、RDF が出力される。出力内容は-separate_self によって異なる。

- "-separate_self false"の場合
 - * 1 列目: 距離
 - * 2 列目: RDF
- "-separate_self true"の場合
 - * 1 列目: 距離
 - * 2 列目: 異なる分子間の RDF
 - * 3 列目: 同一分子内の RDF (自己相関関数と呼ばれる)

3.6.4 解析例

3.6.4.1 水分子の酸素原子間の RDF

```
#!/bin/bash
```

```
anatra rdf \
  -stype      parm7 \
  -sfile      complex.prmtop \
  -tintype    dcd \
  -tin        INP.dcd \
  -fhead      out \
  -sel0       resname WAT and name 0 \
  -sel1       resname WAT and name 0 \
  -mode0      residue \
  -mode1      residue \
  -dr         0.1 \
  -identical  true \
  -normalize  true \
  -separate_self false
```

3.6.4.2 水分子の酸素原子-水素原子間の RDF

```
#!/bin/bash

anatra rdf \
  -stype      parm7 \
  -sfile      complex.prmtop \
  -tintype    dcd \
  -tin        INP.dcd \
  -fhead      out \
  -sel0       resname WAT and name O \
  -sel1       resname WAT and name H1 \
  -mode0      residue \
  -mode1      residue \
  -dr         0.1 \
  -identical  false \
  -normalize  true \
  -separate_self true
```

3.6.5 タンパク質の C α 原子と水分子の酸素原子の間の RDF

タンパク質 (resid 1 to 192) の各 C α 原子と水分子の酸素原子の間の RDF を計算し、各 RDF を平均化したものが出力される。

```
anatra rdf \
  -stype      parm7 \
  -sfile      complex.prmtop \
  -tintype    dcd \
  -tin        INP.dcd \
  -fhead      out \
  -sel0       resid 1 to 192 and name CA \
  -sel1       resname WAT and name O \
  -mode0      atom \
  -mode1      residue \
  -dr         0.1 \
  -identical  false \
  -normalize  true \
  -separate_self false
```

3.7 空間分布関数 (spatial_distr, sdf)

現バージョンでは gcc コンパイラを使った際に spatial_distr は使用できないので注意

3.7.1 理論背景

粒子 a が N_a 個存在している系 (体積 V , 数密度 $\rho_a = N_a/V$) を考える. ただし, 系の中心に分子が一つ配向も含めてピン止めされている状況を考える. この場合, ピン止めされた分子は外場の働きをする. このような状況は通常の溶液系のシミュレーションではありえないが, この節の最後で述べるようにトラジェクトリのフィッティングを行うことで, 仮想的に同様の状況を作り出すことができる.

動径分布関数の場合と同様に, 粒子 a の局所密度場を次式で定義する.

$$\hat{\rho}_a(\mathbf{r}) = \sum_{i=1}^{N_a} \delta(\mathbf{r} - \mathbf{r}_i). \quad (3.28)$$

このアンサンブル平均は 1 体密度と呼ばれる.

$$\rho_a(\mathbf{r}) = \langle \hat{\rho}_a(\mathbf{r}) \rangle_{\text{ext}} \quad (3.29)$$

$$= \left\langle \sum_{i=1}^{N_a} \delta(\mathbf{r} - \mathbf{r}_i) \right\rangle_{\text{ext}}. \quad (3.30)$$

ここで, $\langle \dots \rangle_{\text{ext}}$ は外場存在下でのアンサンブル平均を表す. 外場が存在しないとき $\rho_a(\mathbf{r}) = \rho_a$ となる. また, 中心の分子 (外場) から遠く離れているところでも, $\rho_a(\mathbf{r}) = \rho_a$ となる. 空間分布関数 (spatial distribution function, SDF) を次式で定義する.

$$g_a(\mathbf{r}) = \frac{\rho_a(\mathbf{r})}{\rho_a}. \quad (3.31)$$

$g_a(\mathbf{r}) > 0$ はその位置での粒子 a の存在確率がバルクよりも高く, $g_a(\mathbf{r}) < 0$ は存在確率がバルクよりも低いことを意味する.

SDF を計算する際には, まず MD トラジェクトリ内に含まれている外場として扱う分子を参照構造に重なるように並進・回転させる (フィッティング). また, アンラップされているトラジェクトリについては, あらかじめ PBC ラップを行っておく必要がある (図 3.2). spatial_distr では, これら一連の操作を一括して行うことが可能である. 変換を行った後は, 注目している分子は特定の位置・配向に固定された状態になるため, 固定された外場として扱うことができる. 周囲の分子の位置から SDF の計算が可能である. しかし, 外場として扱う分子が, 特定の安定構造を持たない場合は, フィッティングが適切に行えないため, SDF の計算は困難となる. 一方で, 天然構造を持つタンパク質については, シミュレーションの間に大きな構造変化が起きていなければ, 多くの場合 SDF 計算は可能である.

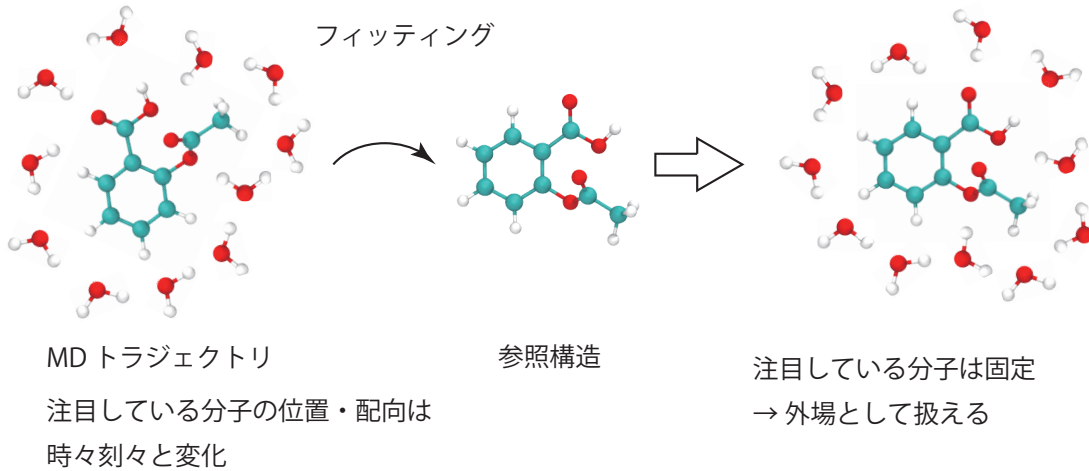


図 3.2: トラジェクトリ内の注目している分子を参照構造に重なるように系を並進・回転させることで (フィッティング), 特定の配向の分子周囲の他分子の 3 次元的位置を調べることが可能である. `spatial_distr` では, これら一連の操作を一括して行うことが可能である.

3.7.2 オプション一覧

オプション	備考	オプション	備考
<code>-stype</code>	Common	<code>-sfile</code>	Common
<code>-tintype</code>	Common	<code>-tin</code>	Common
<code>-flist_traj</code>	Common	<code>-fhead</code>	Common
<code>-selX</code>	Common	<code>-mode</code>	Common
<code>-ng3</code>	Mode specific	<code>-del</code>	Mode specific
<code>-origin</code>	Mode specific	<code>-fit</code>	Mode specific
<code>-refpdb</code>	Mode specific	<code>-refselid</code>	Mode specific
<code>-fitselid</code>	Mode specific	<code>-use_spline</code>	Mode specific
<code>-spline_resolution</code>	Mode specific	<code>-prep_only</code>	Common

上記以外にも, 条件付き空間分布など, より発展的な解析に対応したオプションも実装されているが, それらは次のアップデートの際に公式サポート予定である.

- `-ng3` $\langle x, y, z$ 軸のグリッド数 \rangle

- Default: 50 50 50
- Example: `-ng3 50 50 50`

x, y, z 軸方向のグリッド数 n_x, n_y, n_z を指定する. 従って, 合計グリッド数は $n_x n_y n_z$ である.

- `-del` $\langle x, y, z$ 軸のグリッド幅 (Å) \rangle

- Default: 0.5 0.5 0.5
- Example: `-del 0.5 0.5 0.5`

x, y, z 軸のグリッド幅 $\Delta x, \Delta y, \Delta z$ (angstrom) を指定する。従って、グリッド点一つの体積要素 $\Delta V = \delta x \delta y \delta z$ である。

- **-origin** <原点の (x, y, z) 座標 (Å)>

- Default: 0.0 0.0 0.0
- Example: `-origin 0.0 0.0 0.0`

グリッドを張るときの原点を指定する。原点の座標を (x_0, y_0, z_0) とすると、グリッド点 i_x, i_y, i_z の座標 (x_i, y_i, z_i) は次式で与えられる。

$$x_i = x_0 + \Delta x (i_x - 1) \quad (3.32)$$

$$y_i = y_0 + \Delta y (i_y - 1) \quad (3.33)$$

$$z_i = z_0 + \Delta z (i_z - 1) \quad (3.34)$$

- **-fit** <true or false>

- Default: false
- Example: `-fit true`

外場として扱う分子 (溶質) 周囲を中心として、PBC ラップを行った後に、溶質分子が参照構造に重なるようにフィッティングする。-fit true の場合、次以降に示す `-refpdb`, `-refselid`, `-fitselid` を指定すると同時に、

`-sel0` <SDF を計算したい分子の selection>

`-sel1` <参照構造内のフィッティングに使う部位>

`-sel2` <トラジェクトリ内のフィッティングに使う部位>

のように 3 つの selection を指定する必要がある。入力のトラジェクトリがフィッティング後の場合、必ず **-fit false** に設定しなければならない。フィッティング後のトラジェクトリに対して PBC ラップを行うと、不適切な分子配置になってしまうためである。

- **-refselid**

- Default: N/A
- Example: `-refselid 1`

参照構造内のフィッティングに使う部位の selid を指定する。-fit で例のように `-selX` ($x=0, 1, 2$) を指定した場合には、`-refselid 1` とする。

- **-fitselid** <フィッティングに使うトラジェクトリ内の部位に対応する selid>

- Default: N/A
- Example: `-fitselid 2`

フィッティングに使うトラジェクトリ内の部位の `selid` を指定する。 `-fit` での例のように `-selX` ($x=0, 1, 2$) を指定した場合には、 `-fitselid 2` とする。

- **-use_spline <true or false>**

- Default: `false`
- Example: `-use_spline true`
- Note: スプラインによるグリッド細分化度は `-spline_resolution` で指定する。

SDF 計算の際、スプライン補間を行う。このプログラムでは Akima 補間を用いる。

- **-spline_resolution <スプラインによるグリッド細分化度>**

- Default: 4
- Example: `-spline_resolution 4`
- Note: "`-use_spline true`"のときに有効となる。

スプラインによるグリッド細分化の度合いを整数値で指定する。この値を n_r とし、スプライン実行前の SDF のグリッド刻みを $\Delta\alpha$ ($\alpha = x, y, z$) とすると、スプラインを施した SDF のグリッド刻みは

$$\Delta\alpha' = \Delta\alpha/n_r, \quad (3.35)$$

となる。また、スプライン実行前の合計グリッド数を N 、実行後のグリッド数を N' とすると、

$$N' = n_r^3 N, \quad (3.36)$$

となる。

3.7.3 出力ファイルのフォーマット

- ***_out_g_original.dx**

SDF が dx 形式で出力される。このファイルに格納されている SDF はスプラインが施されていない。

- ***_out_g_spline.dx**

SDF が dx 形式で出力される。このファイルに格納されている SDF はスプラインが施されている。"`-use_spline true`"のときのみこのファイルが出力される。

3.7.4 解析例

3.7.4.1 タンパク質周りの水分子の SDF

タンパク質 (resid 1 to 192) 周りの水分子 (重心) の SDF を計算する.

```
#!/bin/bash

anatra spatial_distr \
  -stype          parm7 \
  -sfile          complex.prmtop \
  -tintype        dcd \
  -tin            INP.dcd \
  -fhead          out \
  -sel0           water \
  -sel1           resid 1 to 192 and noh \
  -sel2           resid 1 to 192 and noh \
  -mode           residue \
  -ng3            50 50 50 \
  -del            0.4 0.4 0.4 \
  -origin         -10.0 -10.0 -10.0 \
  -fit            true \
  -refpdb         ref.pdb \
  -refselid       1 \
  -fitselid       2 \
  -use_spline     true \
  -spline_resolution 4
```

3.8 平均二乗偏差 (rmsd)

3.8.1 理論背景

ここでは、平均二乗偏差 (Root-mean-square deviation, RMSD) の定義について解説する. RMSD は注目する分子やその特定部位について、MD 計算中のある時刻での構造が、参照とする構造 (参照構造) からどの程度ずれているかを定量的に評価するものである. 例えばあるタンパク質について、結晶構造を参照構造として、平衡化過程で参照構造からどの程度構造のズレが生じ、また構造が緩和しているかを判別する際によく用いられる.

原子 N_{sel} 個からなる分子について、 \mathbf{r}_i を i 番目の原子の座標、 $\mathbf{r}_i^{\text{ref}}$ を参照構造中の原子の座標、 $\mathbf{r}(t)$ を時刻 t での座標とする. この時、平均二乗偏差 $\chi(t)$ は以下の式で定義される.

$$\chi(t) = \sqrt{\frac{1}{N_{\text{sel}}} \sum_{i=1}^{N_{\text{sel}}} |\mathbf{r}_i(t) - \mathbf{r}_i^{\text{ref}}|^2}. \quad (3.37)$$

3.8.2 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-fout	Common	-sel0	Common
-sel1	Common	-sel2	Common
-sel3	Common	-fit	Mode specific
-refpdb	Common	-fitselid	Mode specific
-refselid	Mode specific	-rmsdselid	Mode specific
-rmsdrefselid	Mode specific		

- **-fout <出力ファイル>**

- Default: out.rmsd
- Example: -fout out.rmsd

計算結果を出力するファイルを指定する.

- **-refpdb <参照構造の PDB ファイル>**

- Default: N/A
- Example: -refpdb ref.pdb

参照構造の PDB ファイルを指定する. この場合は RMSD 計算における参照構造である.

- **-fit <true or false>**

- Default: false
- Example: -fit true
- Note: true の場合, -refselid, -fitselid の指定が必要となる.

RMSD 計算の前にフィッティングを行うかどうかを指定する.

- **-fitselid <selid>**

- Default: N/A
- Example: -fitselid 1

- **-refselid <selid>**

- Default: N/A
- Example: `-fit selid 2`

フィッティングを行う際, その対象部位に対応する `selid` を指定する. 従って, このオプションは "`-fit true`" のときに必須となる. 参照構造内のフィッティングに使う部位は "`-refselid`" で指定する.

- **`-rmsdselid <selid>`**

- Default: N/A
- Example: `-rmsdselid 2`

RMSD 計算を行う分子や部位に対応する `selid` を指定する.

- **`-rmsdrefselid <selid>`**

- Default: N/A
- Example: `-rmsdrefselid 3`

RMSD 計算を行う際の参照構造に対応する `selid` を指定する.

3.8.3 出力ファイルのフォーマット

- **`*.rmsd` ファイル**

このファイルには, RMSD が出力される.

- 1 列目: 時間
- 2 列目: RMSD

3.8.4 解析例

3.8.4.1 水中におけるタンパク質の結晶構造からの RMSD

タンパク質 (resid 1 to 161) の $C\alpha$ 原子に対する RMSD を計算する.

```
anatra rmsd \
-stype      parm7 \
-sfile      complex.prmtop \
-tintype    dcd \
-tin        INP.dcd \
-fout       out.rmsd \
-sel0       resid 1 to 161 and name CA \
```

(次のページに続く)

(前のページからの続き)

```

-sel1      resid 1 to 161 and name CA  \
-sel2      resid 1 to 161 and name CA  \
-sel3      resid 1 to 161 and name CA  \
-fit       true                        \
-refpdb    ref.pdb                    \
-fitselid  0                          \
-refselid  1                          \
-rmsdselid 2                          \
-rmsdrefselid 3

```

3.9 相互作用エネルギー (interaction_energy, ene)

現バージョンでは gcc コンパイラを使った際に **interaction_energy** は使用できないので注意

interaction_energy (ene) では選択した分子やその集団、もしくは特定部位等の間に働く相互作用エネルギーを計算できる。

現在 **anatra** では以下の三種類の相互作用が計算可能である。

- 静電相互作用

分子 A と分子 B の間に働く静電相互作用 U_{elec} について、2通りのスキームを実装している。一つは Bare 相互作用であり、 i と j はそれぞれ分子 A と B の原子インデックス番号とすると A-B 間の相互作用は

$$U_{\text{elec}} = \sum_{i \in A} \sum_{j \in B} \frac{q_i q_j}{r_{ij}}, \quad (3.38)$$

で表される。もう一つは、Smooth particle mesh Ewald (SPME) 法による静電相互作用の計算である。実際の MD 計算は長距離的な静電相互作用を考慮するために SPME 法を用いるのが現在の標準であり、MD 計算と正しく対応する静電相互作用の計算が必要な場合は SPME 法を用いた計算が必要となる。

- Lennard-Jones (LJ) 相互作用

van der Waals 相互作用を表現するための経験的なモデルである Lennard-Jones 相互作用は、以下の式で定義される。

$$U_{\text{LJ}} = \sum_{i \in A} \sum_{j \in B} 4\epsilon_{ij} \left\{ \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right\}. \quad (3.39)$$

この式中の ϵ_{ij} と σ_{ij} は分子 A 中の原子 i と分子 B 中の原子 j との間の LJ パラメータである。

- Attractive LJ 相互作用 Attractive LJ 相互作用は、以下で定義される。

$$U_{\text{attr}} = \sum_{i \in A} \sum_{j \in B} u_{\text{attr},ij}, \quad (3.40)$$

$$u_{\text{attr},ij} = \begin{cases} 4\epsilon_{ij} \left\{ \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right\} & r_{ij} \geq 2^{1/6}\sigma \\ -\epsilon_{ij} & r_{ij} < 2^{1/6}\sigma. \end{cases} \quad (3.41)$$

これは LJ 相互作用の極小部分を引き伸ばした形の関数であり (図 3.3 参照)、LJ 相互作用の関数で生じていた、エネルギーに対する距離の多価性を取り除いたものである。分子会合過程等の化学過程について PMF

作成を行う際、Attractive LJ 相互作用を反応座標として用いれば、LJ 相互作用に比べて明確に解離や会合などの状態区別が可能となる。

以上の 3 種類の相互作用計算に加え、静電相互作用と LJ 相互作用、もしくは静電相互作用と attractive LJ 相互作用の和を計算することも可能である。

3.9.1 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel0	Common	-sel1	Common
-mode0	Common	-mode1	Common
-parmformat	Mode specific	-fanaparm	Mode specific
-pbc	Mode specific	-calc_vdw	Mode specific
-calc_elec	Mode specific	-vdw_type	Mode specific
-elec_type	Mode specific	-dt	Mode specific
-rvdwcut	Mode specific	-relcut	Mode specific
-pme_alpha	Mode specific	-pme_grids	Mode specific
-pme_rigid	Mode specific	-pme_dual	Mode specific
-pme_fprm	Mode specific	-prep_only	Common

- **-parmformat <prmtop or anaparm>**

- Default: prmtop
- Example: -parmformat anaparm
- Note : anaparm を指定した場合、-fanaparm の指定が必要となる。

電荷情報を参照するパラメータファイルを指定する。prmtop を指定する場合は-sfile にて指定した prmtop ファイルを自動で参照する。

- **-fanaparm <anaparm ファイル名>**

- Default: N/A
- Example: -fanaparm complex.anaparm

LJ パラメータや電荷情報を参照する anaparm ファイル名を指定する。このオプションは-parmformat anaparm のときに必要となる。

- **-pbc <true or false>**

- Default: false
- Example: -pbc true

周期境界を考慮して相互作用計算を行うかを指定する。つまり box の上下左右に同様の box が存在するとしてその中のミラー分子からの寄与も計算するかを決定する。

- **-calc_vdw <true or false>**

- Default: true
- Example: -calc_vdw false

van der Waals (LJ) 相互作用を計算するかを指定する。

- **-calc_ele <true or false>**

- Default: false
- Example: -calc_ele true

静電相互作用を計算するかどうかを指定する。

- **-vdw_type <standard or attractive>**

- Default: standard
- Example: -vdw attractive

計算する van der Waals 相互作用の関数を選択する。standard は通常の LJ 相互作用 (式 (3.39)) を計算し, attractive は U_{attr} (式 (3.40)) を計算する

- **-elec_type <bare or pme>**

- Default: bare
- Example: -elec bare

静電相互作用の計算方法を指定する。bare の場合には、式 (3.38) に基づいて計算が行われる。pme を指定した場合、Smooth particle mesh Ewald (PME) 法に基づいて計算が行われる。

- **-dt <時間刻み>**

- Default: 0.1
- Example: -dt 1.0

時系列データ出力の際の時間軸（第 1 列）の時間刻みを指定する.

- **-rvdwcut** <カットオフ距離 (Å)>

- Default: 12.0
- Example: `-rvdwcut 20.0`

vdW 相互作用計算の際のカットオフ距離を指定する.

- **-relcut** <カットオフ距離 (Å)>

- Default: 1.0e10
- Example: `-relcut 1.0e10`

静電相互作用計算の際のカットオフ距離を指定する.

- **-pme_alpha** <スクリーニングパラメータ (\AA^{-1})>

- Default: 0.35e0
- Example: `-pme_alpha 0.35e0`

PME 法におけるスクリーニングパラメータの値を指定する. `-elec pme` の時のみ参照される値である.

- **-pme_grids** < x, y, z 軸のグリッド数>

- Default: 64 64 64
- Example: `-pme_grids 64 64 64`

x, y, z 軸方向のグリッド数 n_x, n_y, n_z を指定する. 従って, 合計グリッド数は $n_x n_y n_z$ である.

- **-pme_rigid** <true or false>

- Default: false
- Example: `-pme_rigid false`

SPME 計算を行う際, `-sel0` で指定した部位が空間上に固定されている場合に, `-pme_rigid true` と設定することで, 計算負荷を削減することができる.

3.9.2 出力ファイルのフォーマット

- *.uij ファイル

選択部位間の静電相互作用が出力される。-mode 0, -mode1 の指定により、部位 0,1 がそれぞれ n_1 個, n_2 個の分子に分割されたとすると、合計で $n_1 n_2$ 通りの相互作用が出力されることとなる。この時、各データは以下のように出力される。

- 1 列目: 時間軸
- 2 列目: 分子対 (1, 1) の相互作用
- 3 列目: 分子対 (1, 2) の相互作用…
- n_2 列目: 分子対 (1, n_2) の相互作用
- $n_2 + 1$ 列目: 分子対 (2, 1) の相互作用
- $n_2 + 2$ 列目: 分子対 (2, 2) の相互作用…

3.9.3 解析例

ここではよく行なう解析について最低限のオプションを使ったスクリプトの例を示しておく。

3.9.3.1 溶質間の静電相互作用と LJ 相互作用の和

タンパク質 (resid 1 to 161) とリガンド分子 (resname LIG) の間の静電相互作用+LJ 相互作用の和を計算する場合を示す。ここでは、静電相互作用の計算に SPME 法を用いる。

```
#!/bin/bash

anatra energy
  -stype      parm7                \
  -sfile      complex.prmtop       \
  -tintype     dcd                  \
  -tin        INP.dcd              \
  -fhead      out                   \
  -parmformat  prmtop               \
  -pbc        true                  \
  -calc_vdw   true                  \
  -calc_elec   true                  \
  -vdw_type    standard             \
  -elec_type   pme                  \
  -mode0       whole                \
  -mode1       residue              \
```

(次のページに続く)

(前のページからの続き)

```

-dt          0.1          \
-rvdwcut     12.0         \
-relcut      12.0         \
-pme_alpha   0.35         \
-pme_grids   64 64 64    \
-pme_rigid   false       \
-sel0        resid 1 to 161 \
-sel1        resname LIG  \
-prep_only   false

```

3.9.3.2 溶質間の LJ 相互作用

タンパク質 (resid 1 to 161) とリガンド分子 (resname LIG) の間の LJ 相互作用を計算する場合を示す.

```

#!/bin/bash

anatra energy
  -stype      parm7          \
  -sfile      complex.prmtop \
  -tintype    dcd            \
  -tin        INP.dcd        \
  -fhead      out            \
  -parmformat prmtop         \
  -pbc        true           \
  -calc_vdw   true           \
  -calc_elec   false         \
  -vdw_type    standard      \
  -mode0       whole         \
  -mode1       residue       \
  -dt          0.1           \
  -rvdwcute    12.0          \
  -relcut      1.0e10        \
  -sel0        resid 1 to 161 \
  -sel1        resname LIG    \
  -prep_only   false

```

3.10 脂質膜秩序 (lipid_order, scd)

lipid_order (scd) ではリン脂質のアシル鎖に関する配向秩序パラメータ S_{CD} の計算が可能である。脂質膜の法線方向を z として、特定の炭素原子と隣接する水素原子のなす角を θ と定義する。 S_{CD} は θ を用いて式で定義される。

$$S_{CD} = \frac{1}{2} \langle 3 \cos^2 \theta - 1 \rangle. \quad (3.42)$$

S_{CD} の値域は $0 \leq S_{CD} \leq 1$ であり、その値が大きいほど秩序性が高く、0 は無秩序であることを意味する。 S_{CD} は炭素ごとに定義されるので、炭素原子の位置に応じた膜秩序の解析が可能である。

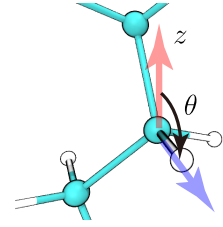


図 3.3: z 軸と特定の炭素原子と水素原子を結ぶベクトルのなす角を θ と定義する。 θ は脂質の局所的な配向を表す。

3.10.1 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-selX	Common	-mode	Common
-cindex	Mode specific	-prep_only	Common

• -selX <部位選択>

- Default: N/A
- Sample: -sel0 name C32 H2X H2Y

炭素原子および隣接する水素原子を指定する。 S_{CD} を計算したい炭素原子ごとに -selX で指定する。

• -cindex < Carbon index >

- Default: N/A
- Sample: -cindex 1 2 3 4 5

-sel0, -sel1, … で指定した炭素原子が、アシル鎖の付け根から数えて何番目か (Carbon index) をこのオプションで指定する。したがって、-selX で定義した selid の数分だけ番号を並べる必要がある。このオプションで指定した Carbon index は、出力ファイル*.scd の x 軸 (1 列目) として用いられる。

3.10.2 出力ファイルのフォーマット

- *.scd

S_{CD} を -sel0, -sel1, ... の順番で指定した炭素原子の順番で出力する.

- 1 列目: Carbon index (-cindex で指定)
- 2 列目: S_{CD}

3.10.3 解析例

3.10.3.1 DPPC 脂質膜の S_{CD} 計算

```
anatra lipid_order \
  -stype      psf \
  -sfile      complex.psf \
  -tintype     netcdf \
  -flist_traj flist \
  -fhead      run_nc \
  -dt         0.1 \
  -cindex     1 2 3 4 5 \
  -sel0       name C23 H3R H3S and segid MEMB \
  -sel1       name C24 H4R H4S and segid MEMB \
  -sel2       name C25 H5X H5Y and segid MEMB \
  -sel3       name C26 H6X H6Y and segid MEMB \
  -sel4       name C27 H7X H7Y and segid MEMB \
  -prep_only  false
```

3.11 界面法線方向 (z) に沿った分布関数 (z_profile, zprof)

z_profile (zprof) は, 膜界面の法線方向 (z) に沿った分布を計算する. 分子 a の z 軸に沿った局所密度場を $\hat{\rho}_a(z)$ と定義する.

$$\hat{\rho}_a(z) = \frac{1}{A_{xy}} \sum_{i=1}^{N_a} \delta(z - z_i). \quad (3.43)$$

ここで, A_{xy} はシミュレーションセルの xy 面の面積, z_i は i 番目の分子の z 座標である. これの平均をとったものが, 分布関数である.

$$\rho_a(z) = \langle \hat{\rho}(z) \rangle. \quad (3.44)$$

実験データと比較するために, 電子密度分布 $\rho_a^e(z)$ を計算することも多い. その場合の局所密度場は

$$\hat{\rho}_a^e(z) = \sum_{i=1}^{N_a} \sum_{\lambda=1}^{N_s} q_{\lambda} \delta(z - z_{i,\lambda}), \quad (3.45)$$

で与えられる。ただし、 q_λ は分子 a 内の λ 番目の原子が持つ電子数、 $z_{i,\lambda}$ は i 番目の分子 a 内にある λ 番目の原子の z 座標である。電子密度分布は $\rho_a^e(z) = \langle \rho_a^e(z) \rangle$ と定義される。

3.11.1 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel0	Common	-sel1	Common
-mode0	Common	-mode1	Common
-target_selid	Mode specific	-center_selid	Mode specific
-denstype	Mode specific	-symmetrize	Mode specific
-prep_only	Common	-	-

- **-selid0 <selid>**

- **-selid1 <selid>**

- Default: N/A
- Example: -selid0 resname ETOH -selid1 segid MEMB

分布関数を計算したい分子を選択する。また、 $z = 0$ を膜中心に設定したい場合は、膜分子も追加で選択する。

- **-target_selid <selid>**

- Default: N/A
- Example: -target_selid 0

-sel x のうち、分布関数を計算したい分子に対応する selid を指定する。

- **-center_selid <selid>**

- Default: false
- Example: -target_selid 1

-sel x のうち、膜分子に対応する selid を指定する。この selid で指定された膜分子の重心が $z = 0$ に設定される。

- **-symmetrize <true or false>**

- Default: false
- Example: `-symmetrize false`

$z < 0$ と $z > 0$ の分布を平均して対称化するかどうかを指定する.

- **-denstype <number or electron>**

- Default: number
- Example: `-denstype number`

数密度分布 (number) か電子密度分布 (electron) のどちらを計算するかを指定する.

3.11.2 出力ファイルのフォーマット

- *.zprof

-target_selid で指定した selid の分子の分布関数が出力される.

- 1 列目: z 座標
- 2 列目: 分布関数

3.11.3 解析例

3.11.3.1 脂質膜-水系における水分子の数密度分布

```
#!/usr/bin/env bash
```

```
anatra z_profile \
  -stype      psf \
  -sfile      complex.psf \
  -tintype    netcdf \
  -flist_traj flist \
  -fhead      run \
  -sel0       water \
  -sel1       segid MEMB and noh \
  -mode0      residue \
  -mode1      whole \
  -target_selid 0 \
  -center_selid 1 \
  -denstype   number \
  -prep_only  false
```

3.12 特定の方向 z に対する分子の配向 (z_orient, zori)

z_orient (zori) は、特定の方向 (z) を基準として、分子の配向分布を計算する。分子内にある 2 つの部位 **bottom**, **head** を定義し、それぞれの重心を $\mathbf{R}_B, \mathbf{R}_H$ とおく。そして、分子軸 \mathbf{u} を

$$\mathbf{u} = \mathbf{R}_H - \mathbf{R}_B, \quad (3.46)$$

と定義する。 z 軸 (単位ベクトル \mathbf{e}_z) と \mathbf{u} (単位ベクトル \mathbf{e}_u) のなす角を θ とすると、

$$\cos \theta = \mathbf{e}_z \cdot \mathbf{e}_u, \quad (3.47)$$

である。 $\cos \theta$ に対する分布関数を

$$P(\chi) = C \langle \delta(\chi - \cos \theta) \rangle, \quad (3.48)$$

とする。ただし、 C は規格化定数であり、

$$\int_0^\pi d\theta \sin \theta P(\chi) = 1, \quad (3.49)$$

を満たすように設定する。上式は、 θ から $\chi = \cos \theta$ への変数変換を考えることで、次式のように書き直せる。

$$\int_0^1 d\chi P(\chi) = 1. \quad (3.50)$$

z_orient (zori) は、上式を満たす $P(\chi) = P(\cos \theta)$ を計算する。

さらに、実験室座標系での z 軸に加えて、特定の分子の向きによって定義されるベクトルを z 方向に設定することも可能である。その分子の 4 つの部位の重心を $\mathbf{x}_{B_0}, \mathbf{x}_{H_0}, \mathbf{x}_{B_1}, \mathbf{x}_{H_1}$ として、ベクトル

$$\mathbf{v}_0 = \mathbf{x}_{H_0} - \mathbf{x}_{B_0} \quad (3.51)$$

$$\mathbf{v}_1 = \mathbf{x}_{H_1} - \mathbf{x}_{B_1} \quad (3.52)$$

を用意することで、 z 方向の単位ベクトル

$$\mathbf{e}_z = \frac{\mathbf{v}_1 \times \mathbf{v}_2}{|\mathbf{v}_1 \times \mathbf{v}_2|} \quad (3.53)$$

を定義する。このようにして定義した \mathbf{e}_z と注目している分子の分子軸ベクトル \mathbf{u} のなす角によって配向を定義することで、配向分布 $P(\chi)$ の定義・計算が可能である。

3.12.1 オプション一覧

オプション	備考	オプション	備考
-stype	Common	-sfile	Common
-tintype	Common	-tin	Common
-flist_traj	Common	-fhead	Common
-sel x	Common	-mode x	Common
-zdef_type	Mode specific	-vecX_bottom_selid	Mode specific
-vecX_head_selid	Mode specific	-bottom_selid	Mode specific
-head_selid	Mode specific	-judgeup	Mode specific
-nmolup	Mode specific	-center_selid	Mode specific
-dx	Mode specific	-dt	Mode specific
-prep_only	Common	-	-

- **-sel x <selection>** (最大 7 個)

- Default: N/A
- Example: -sel0 name P -sel1 name N

配向を定義するための部位 bottom, head を指定する。実験室座標系ではなく、特定の分子の配向によって決まる方向を z 方向に用いるときは (-zdef_type を参照), ベクトル $\mathbf{v}_1, \mathbf{v}_2$ (式 (3.51)), (3.52)) を定義するための部位選択を行う必要がある。また、実験室座標系の z 軸を用いる場合において、 \mathbf{e}_z の向きを特定分子の重心の z 座標を基準にして反転させるときには (-judgeup を参照), その分子の選択も必要となる。

- **-head_selid <selid>**

- **-bottom_selid <selid>**

- Default: N/A
- Example: -head_selid 0 -bottom_selid 1

分子軸 \mathbf{u} (式 (3.46)) を定義するための部位 bottom, head に対応する部位をそれぞれ -bottom_selid, -head_selid で指定する。

- **-zdef_type <coord or outerprod>**

- Default: N/A
- Example: -zdef_type coord

- **-vec x _bottom_selid <selid>** ($x = 0, 1$)

- **-vecx_head_selid <selid>** ($x = 0, 1$)

z 方向の定義を指定する. 実験室座標系の z 軸を z 方向に設定する場合は `coord` を, 特定の分子の配向によって z 方向を定義する場合は `outerprod` を指定する. `outerprod` の場合は, `-vecx_bottom_selid`, `-vecx_head_selid` を設定する必要がある. これらのオプションは, 式 (3.51), (3.52) の

- x_{B_0} : `-vec0_bottom_selid`
- x_{H_0} : `-vec0_head_selid`
- x_{B_1} : `-vec1_bottom_selid`
- x_{H_1} : `-vec1_head_selid`

に対応している.

- **-judgeup <nmolup coord or none>**

- Default: `coord`
- Example: `-judgeup coord`

- **-nmolup <# of target molecules in upper leaflet>**

- Default: 0
- Example: `-nmolup 100`

- **-center_selid <selid>**

- Default: N/A
- Example: `-center_selid 2`

実験室座標系の z 軸を基準にとったときの, \mathbf{e}_z の向きをこのオプションで指定する. z 軸の正の方向を向いた単位ベクトルを \mathbf{e}'_z とする.

`-judgeup nmolup` の場合: オプション `-nmolup` で指定した数を N_{up} とすると, 最初の N_{mol} 分子は $\mathbf{e}_z = \mathbf{e}'_z$ として $\cos \theta$ を計算し, それ以外の分子は $\mathbf{e}_z = -\mathbf{e}'_z$ として $\cos \theta$ を計算する.

`-judgeup coord` の場合: `-center_selid` で指定した分子の重心の z 座標を z_c , `-bottom_selid` と `-head_selid` で指定された部位の重心の z 座標をそれぞれ z_B, z_H とすると,

$$\mathbf{e}_z = \begin{cases} \mathbf{e}'_z & \frac{z_H + z_B}{2} > 0 \\ -\mathbf{e}'_z & \frac{z_H + z_B}{2} \leq 0 \end{cases} \quad (3.54)$$

のように, 注目している分子の位置に応じて, \mathbf{e}_z の向きを設定する.

- **-dx <グリッド幅 (Å)>**

- Default: 0.1

- Example: `-dx 0.1`

$P(\chi)$ を計算するときのグリッド幅 ($\Delta\chi$) を指定する.

- **-dt <時間刻み>**

- Default: 1
- Example: `-dt 1`

時系列データ出力の際の時間軸 (第 1 列) の時間刻みを指定する.

3.12.2 出力ファイルのフォーマット

- ***.costheta**

$\cos \theta$ の時系列データが出力される.

- 1 列目: 時間
- 2 列目: 1 番目の分子の $\cos \theta$
- 3 列目: 2 番目の分子の $\cos \theta$...

- ***.distr**

$P(\chi) = P(\cos \theta)$ が出力される.

- 1 列目: x 軸 ($\cos \theta$)
- 2 列目: $P(\cos \theta)$

3.12.3 解析例

3.12.3.1 脂質二分子膜内における脂質分子の配向分布

```
#!/usr/bin/env bash
```

```
anatra z_orient \
  -stype      psf \
  -sfile      complex.psf \
  -tintype     nc \
  -flist_traj flist \
  -fhead      run_nc \
  -judgeup    coord \
  -dx         0.1 \
```

(次のページに続く)

(前のページからの続き)

```
-dt          0.1          \  
-sel0        name P       \  
-sel1        name N       \  
-sel2        segid MEMB and noh \  
-mode0       residue      \  
-mode1       residue      \  
-mode2       whole        \  
-bottom_selid 0           \  
-head_selid  1           \  
-center_selid 2          \  
-prep_only   false
```