gmx_clusterByFeatures Documentation

Release 0.1.17

Rajendra Kumar

Jul 01, 2019

Contents

gmx_clusterByFeatures can be used to cluster the conformations of a molecule in a molecular dynamics trajectory using collection of features. The features could be any quantity as a function of time such as Projections of eigenvector from PCA or dihedral-PCA, distances, angles, channel radius etc.

CHAPTER 1

Installation on Linux and MacOS

```
sudo pip3 install gmx-clusterByFeatures
```

No dependency on GROMACS. Just install it and use it.

CHAPTER 2

Contents

2.1 gmx_clusterByFeatures

During the Molecular Dynamics Simulations, molecule conformations changes considerably and identifying the conformations is very important to study the biomolecular dynamics. Conformational clustering can be performed to identify different conformations sampled during the simulations.

Most widely approach for conformational clustering is to calculate Root Mean Square Deviations between all conformations and cluster them according to these deviations. However, for large MD trajectories, this RMSD matrix could be huge and takes very long time to calculate. Therefore, an alternative method such as features based clustering can be used to identify the cluster of conformations.

gmx_clusterByFeatures can be used to cluster the conformations of a molecule in a molecular dynamics trajectory using collection of features. The features could be any quantity as a function of time such as Projections of egienvector from PCA or dihedral-PCA, distances, angles, channel radius etc.

Note: It is developed for **GROMACS MD trajectory**. However, it can be used with any other trajectory format after converting it to GROMACS format trajectory.

When Projections of egienvector from PCA or dihedral-PCA is used as features, it yields clusters depending on the largest conformational changes during the simulations. Depending on the Clustering metrics, a cluster may contain small conformational fluctuations around the respective central structure.

When other features such as distances, angles, channel radius etc are used as the features, the obtained clusters of conformations depends on these features. It can be used to study the specific conformations given the features while ignoring all other conformational fluctuations.

2.1.1 Clustering methods

Presently three methods are implemented:

-

-
-

2.1.2 Clustering metrics

To determine the number of clustering, following metrics are implemented:

- RMSD : Root Mean Square deviation between central structures of clusters.
- SSR/SST ratio ( ) : Relative change in SSR/SST ratio in percentage.
-
-

2.2 Download and Installation

2.2.1 Quick Installation using `pip`

It is **recommended** method to install gmx_clusterByFeatrues.

Not require to install GROMACS

Only available on Linux, MacOS-10.12 (Sierra), MacOS-10.13 (High Sierra) and MacOS-10.14 (Mojave)

On Linux

1. Python3 is available through package managers such as **yum** (Fedora, CentOs), **YaST** (OpenSuse) and **apt-get** (Ubuntu, Linux Mint). For example on ubuntu: run `sudo apt-get install python3` command to install Python3.
2. Install **gmx_clusterByFeatrues** by `sudo pip3 install gmx-clusterByFeatures` command.

On MacOS

1. Python3 is available through package manager. After installing Homebrew, run `brew install python3` command to install Python3.
2. Install **gmx_clusterByFeatrues** by `pip3 install gmx-clusterByFeatures` command.

Note: Presently, installation with pip on MacOS is restricted to **10.12 (Sierra)**, **10.13 (High Sierra)** and 10.14 (Mojave) versions. For other MacOS versions, install gmx_clusterByFeatrues from source as described further below.

Updating gmx_clusterByFeatrues

To update the gmx_clusterByFeatrues package use following command:

```
pip3 install --upgrade --no-deps gmx-clusterByFeatures
```

`--upgrade` flag is used to update the package and `--no-deps` prevents update of dependent packages like numpy, scipy, matplotlib etc.

2.2.2 Installation from source-code

Requirements

It depends on following two packages:

- **GROMACS** : 2016 and above version
- **Python** : 3.4 and above version

GROMACS

A standard installation of GROMACS is sufficient. GROMACS library (`libgromacs.a` or `libgromacs.so`) and header files are required for compilation.

If GROMACS is not installed at standard location, define `GMX_PATH` environment variable as follows:

```
export GMX_PATH=/path/to/installed/gromacs
```

Python3

To compile gmx_clusterByFeatures, Python3 development files should be installed previously.

On Debian like distribution (Debian, Ubuntu, Linux Mint etc.), which uses apt as package manager, python3-development files can be installed as follows:

```
sudo apt-get install python3 python3-dev
```

On OS such as fedora/centos/RHEL, which uses yum as package manager, python3- development files can be installed as follows:

```
sudo yum install python3 python3-devel
```

Four other Python packages , , , and are required that can be installed as follows:

```
sudo pip3 install numpy scipy sklearn matplotlib
```

Downloading source-code

It can be downloaded using git as follows

```
git clone -b master https://github.com/rjdkmr/gmx_clusterByFeatures
```

It can be also downloaded as zip file.

Compilation and Installation using python

Clone the repository from github as directed above then follow these steps.

```
cd gmx_clusterByFeatures # or gmx_clusterByFeatures-master (zip file download)
export GMX_PATH=/path/to/installed/gromacs
sudo GMX_PATH=$GMX_PATH python3 setup.py install
```

Now, gmx_clusterByFeatures command will be accessible in terminal.

Compilation and Installation using cmake for C++ IDEs

This method can be used for development purpose using C++ IDE like QT creator and KDevelop etc.

To install and use gmx_clusterByFeatrues from source location:

```
cd gmx_clusterByFeatures # or gmx_clusterByFeatures-master (zip file download)
mkdir build
cd build
cmake -DGMX_PATH=/path/to/installed/gromacs -DINPLACE=ON ..
make
sudo make install  # Only needed for first time install
```

In this installation, only `gmx_clusterByFeatures` executable file is installed at default location (mostly `/usr/local/bin`) while whole package remains at the source location.

This method is extremely useful for development because `make install` is only required for first time to install executable file. During subsequent development, only command `make` need to be repeated. In IDEs `make` command is executed by `build`. In IDEs project build setting, cmake arguments `-DGMX_PATH=/path/to/installed/gromacs -DINPLACE=ON` needs to be added manually.

2.3 How to use gmx_clusterByFeatures?

It contains several sub-commands for different purposes.

Other tools are presently in development.

Table 1: List of sub-commands available in gmx_clusterByFeatures

Command	Function
cluster	Main module to perform clustering
featuresplot	Feature vs Feature plot to check quality of clustering
distmat	Distance-matrix related calculations
matplot	To visulaize/plot matrix obtained from `distmat`
hole	To calculate cavity/channel radius using HOLE program
holeplot	To calculate average and plot hole output radius file
holefeatures	To write radius as a features for clustering
holeclustersplot	To plot or write radius for clusters separately

2.3.1 sub-commands

cluster

It is the main tool for clustering. It takes at least three input files and perform clustering according to the given option. It also generate a log file containing the information related to clustering.

- `gmx_clusterByFeatures cluster` can be used with trajectory and tpr file generated by GROMACS.
- In case of other versions or other programs such as NAMD and AMBER, PDB file can be used in place of tpr file.
- Trajectories from NAMD and AMBER should be converted to GROMACS compatible formats such as trr, xtc, pdb etc.

Execute following command to get full help

```
gmx_clusterByFeatures cluster -h
```

Warning: Only PBC corrected trajectory and tpr files should be used as inputs. PBC corrected PDB/GRO file can be used in place of tpr file.

Command summary

```
gmx_clusterByFeatures cluster [-f [<.xtc/.trr/...>]] [-s [<.tpr/.gro/...>]]
             [-feat [<.xvg>]] [-n [<.ndx>]] [-clid [<.xvg>]] [-g [<.log>]]
             [-fout [<.xtc/.trr/...>]] [-cpdb [<.pdb>]] [-rmsd [<.xvg>]]
             [-b <time>] [-e <time>] [-dt <time>] [-tu <enum>] [-xvg <enum>]
             [-method <enum>] [-nfeature <int>] [-cmetric <enum>]
             [-ncluster <int>] [-crmsthres <real>] [-ssrchange <real>]
             [-db_eps <real>] [-db_min_samples <int>] [-sil_ssize <real>]
             [-nminfr <int>] [-[no]fit] [-[no]fit2central] [-outframe <int>]
             [-sort <enum>] [-plot <string>] [-fsize <int>] [-pltw <real>]
             [-plth <real>]
```

Options summary

Table 2: Options to specify input files to cluster

Option	Default	File type
-f [<.xtc/.trr/...>]	traj.xtc	Trajectory: xtc trr cpt gro g96 pdb tng
-s [<.tpr/.gro/...>]	topol.tpr	Structure+mass(db): tpr gro g96 pdb brk ent
-n [<.ndx>]	index.ndx	Index file
-feat [<.xvg>]	feature.xvg	xvgr/xmgr file

Table 3: Options to specify output files to cluster

Option	Default	File type
-clid [<.xvg>]	clid.xvg	xvgr/xmgr file (Can be used as both input and output)
-g [<.log>]	cluster.log	Log file
-fout [<.xtc/.trr/. . . >]	trajout.xtc	Trajectory: xtc trr cpt gro g96 pdb tng
-cpdb [<.pdb>]	central.pdb	Protein data bank file
-rmsd [<.xvg>]	rmsd.xvg	xvgr/xmgr file

Table 4: Other options to cluster

Option	Default	Description
-b <real>	0	First frame (ps) to read from trajectory
-e <real>	0	Last frame (ps) to read from trajectory
-dt <real>	0	Only use frame when t MOD dt = first time (ps)
-xvg <keyword>	xmgrace	xvg plot formatting: xmgrace, xmgr, none
-method <keyword>	kmeans	Clustering methods. Accepted methods are:: kmeans, dbscan, gmixture
-nfeature <int>	10	Number of features to use for clustering
-cmetric <keyword>	prior	Cluster metrics: Method to determine cluster number. Accepted methods are: prior, rmsd, ssr-sst, pFS, DBI
-ncluster <int>	5	Number of clusters to generate for prior method. Maximum number of cluster for ctrmsd method.
-crmsthres <real>	0.1	RMSD (nm) threshold between central structures for RMSD cluster metric method.
-ssrchange <real>	2	Thershold relative change % in SSR/SST ratio for ssr-sst cluster metric method.
-sil_ssize <real>	20	Percentage of number of frames to be considered as sample size for silhouette score calculation.
-db_eps <real>	0.5	The maximum distance between two samples for them to be considered as in the same neighborhood.
-db_min_samples <int>	20	The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself.
-nminfr <int>	20	Number of minimum frames in a cluster to output it as trajectory
-[no]fit	Enable	Enable fitting and superimposition of the atoms groups different from RMSD/clustering group before RMSD calculation.
-[no]fit2central	Disable	Enable/Disable trajectory superimposition or fitting to central structure in the output trajectory
-outframe <int>	-1	Number of maximum frames in the output trajectories.
-sort <keyword>	none	Sort trajectory according to these values. Accepted methods are: none, rmsd, rmsdist, features, user
-plot <string>	pca_cluster.png	To plot features with clusters in this file.
-fsize <int>	14	Font size in plot.
-pltw <real>	12	Width (inch) of the plot.
-plth <real>	20	Height (inch) of the plot.

Options to specify input files

-f traj.xtc

Input trajectory file of `xtc` `trr` `cpt` `gro` `g96` `pdb` or `tng` format.

Note: If this file is not provided, only clustering will be performed. No operations will be performed that require trajectory such as RMSD calculation, central structure calculations, clustered trajectories etc.

Note: In case of **XTC** and **TNG** formats, writing central structures and clustered trajectories are relatively fast.

-s topol.tpr

An input structure file of `tpr` `gro` `g96` or `pdb` format. It is **required** if trajectory is given as input.

-n index.ndx

If given, index groups from this file will be prompted for selection. Otherwise, default index groups will be prompted for selection.

This file will be ignored when no trajectory file will be provided.

Users will be prompted for three index group

- **Choose a group for the output:** Select a index group to output it as central structure and clustered trajectory. It can be whole system or any part of the system.
- **Choose a group for clustering/RMSD calculation:** The actual atom groups for which clustering has to be done and RMSD has to be calculated.

  Note: If you are doing PCA based clustering, it should be same second index group as selected in `gmx covar` and `gmx anaeig`.

- **Choose a group for fitting or superposition:** The atom groups used for fitting or superposition before RMSD calculation.

  Note: This input will be only prompted when `-fit` or `-fit2central` option is given. Otherwise, group selected above will be used for fitting.

  Note: If you are doing PCA based clustering, it should be same as first index group selected in `gmx covar` and `gmx anaeig`.

-feat features.xvg

It accepts a file containing features of trajectory as a function of time. Its format is similar to the projections file generated by `gmx anaeig`. Therefore, in case of PCA data, output (`-proj`) of `gmx anaeig` can be directly used as input for gmx_clusterByFeatures.

In this file, two columns should be present. First column is time and second column is feature values. Each time-feature columns should be separated by "&".

The format is as following:

```
# FEATURE - 1
# Time    values
0.0     123.12
10.0    123.12
20.0    123.12
.
.
.

&

# FEATURE - 2
0.0     123.12
10.0    123.12
20.0    123.12
.
.
.

&

# FEATURE - 3
0.0     123.12
10.0    123.12
20.0    123.12
.
.
.

&
```

Note: If this file is not provided, `-clid [<.xvg>]` is the required option.

Options to specify output files

-clid clid.xvg

It can be both **input** and **output** file. It contains two columns, first column is time and second column is cluster label/id.

In default case when clustering has to be done, it is generated after clustering is finished and contains information about cluster id of each frame.

However, it can be also given as input to obtain clustered trajectories. For example, if clustering was performed with "gmx cluster", the obtained `-clid [<.xvg>]` file can be used here to extract clustered trajectory.

Note: To treat this as an input file, do not use `-feat [<.xvg>]` option.

`-g cluster.log`

It is output log file and contains several information about clustering methods and obtained results.

`-fout trajout.xtc`

Output clustered trajectories. Separate trajectory of clusters is written for convenience. These separate trajectories can be used for further analysis.

Each trajectory file name is suffixed by its respective cluster-id.

`-cpdb central.pdb`

Output separate pdb files for central structures of each cluster.

Each pdb file name is suffixed by its respective cluster-id.

`-rmsd rmsd.xvg`

RMSD of clustering atom groups with respect to central structure.

Each RMSD file name is suffixed by its respective cluster-id.

In case of `-sort rmsdist` option, RMSD in distance-matrix is calculated.

Other options

`-xvg xmgrace`

It directs the formatting of all output <.xvg> files. By default, <.xvg> files are in `xmgrace` format, which can be plotted using (`xmgrace` command).

To plot with any other program, use `-xvg none` then a plain text file is obtained.

Three keywords are accepted:

- xmgrace

- xmgr
- none

-method kmeans

Method to use for clustering. All the methods used here are used from Python library.

An overview on clustering method are presented .

Presently following methods are implemented:

1. `-method kmeans`

   - It needs cluster number as input (`-ncluster <int>`). Therefore, one should know beforehand how many cluster is there in data. To automatically determine the cluster number, *-cmetric* For more details about k-means method, see .

2. `-method dbscan`

   - It does not require cluster number beforehand. The clusters are controlled by two other input options: *-db_eps* and *-db_min_samples*. For more details about DBSCAN method, see .

3. `-method gmixture`

   - It also needs cluster number as input (`-ncluster <int>`). Therefore, one should know beforehand how many cluster is there in data. To automatically determine the cluster number, see *-cmetric* For more details about k-means method, see .

-nfeature 10

Number of features to be read from *-feat* file.

If file contains less than requested number of features, all features will be read.

-cmetric prior

Cluster metric to determine the total number of cluster automatically, particularly for k-means and Gaussian-mixture model.

Note: All the cluster metrics are only applicable when `-method kmeans` or `-method gmixture` is used.

Presently following cluster metrics are implemented:

1. `-cmetric prior`

   If clusters count is known beforehand, use this with `-ncluster <int>`. Here, `-ncluster` takes input as the clusters count.

2. `-cmetric rmsd`

   Root Mean Square deviation between central structures of clusters. It uses *-crmsthres* option for RMSD threshold/cutoff.

   Note: It requires trajectory file as input. Otherwise, `-cmetric ssr-sst` will be used for cluster metric with default *-ssrchange* value.

3. `-cmetric ssr-sst`

   It is SSR/SST ratio and used for . It is the threshold in relative change in SSR/SST ratio in percentage.

4. `-cmetric silhouette`

   Silhouette score. From wikipedia: First encountered clusters count with highest Silhouette score value is considered as final cluster number.

   To calculate score, either entire data will be considered with option `-sil_ssize -1`, which could be time expensive or percentage of data by random sampling will be taken with option `-sil_ssize`. Because of the random sampling, this score might not be precisely reproduced in successive calculation.

5. `-cmetric DBI`

   . Lowest value is considered.

`-ncluster 5`

It takes the number of clusters. Its usage depends on *-cmetric*.

Note: It is only applicable when `-method kmeans` or `-method gmixture` is used.

Conditions:

1. For `-cmetric prior`, it is considered as the number clusters to be generated.
2. For `-cmetric rmsd`, it is considered as largest number of clusters to be generated and iteratively number of clusters are reduced to check whether RMSD between central structures are **not** below RMSD threshold (`-crmsthres <real>`).
3. For `-cmetric ssr-sst`, `-cmetric pFS` and `-cmetric DBI`, it is considered as maximum number of clusters to generated. At first, two clusters are generated and iteratively number of clusters are increased by one. When maximum number of clusters is reached, these three cluster-metrics are calculated and finally, number of clusters is selected.

`-crmsthres 0.1`

RMSD (nm) threshold between central structures for RMSD cluster metric method.

It is used with `-cmetric rmsd`. In each iteration, RMSD between all central structures are calculated. If any RMSD value is within the input RMSD (nm) threshold, number of clusters is decreased by one in next iteration.

It is assumed that when RMSD between two central structures are within the threshold, central structures are similar enough to merge the two clusters as a single cluster. However, it is **not** necessary that these two clusters will merge in next iteration.

-ssrchange 2.0

Threshold relative percentage change in SSR/SST ratio to choose number of clusters automatically. This threshold gives potential position of Elbow in .

Note: This option is only used when `-cmetric ssr-sst` is provided as input.

-sil_ssize 20

Percentage of number of frames to be considered as sample size for silhouette score calculation. If its value is `-1`, sampling is not considered.

-db_eps 0.5

The maximum distance between two samples for them to be considered as in the same neighborhood.

See also:

scikit-learn DBSCAN class

-db_min_samples 20

The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself.

See also:

scikit-learn DBSCAN class

-nminfr 20

Number of minimum frames in a cluster to output it as trajectory. If number of frames is less than this number, the cluster will be ignored.

-fit/-nofit

Enable fitting and superimposition of the atoms groups different from RMSD/clustering group before RMSD calculation. If Enabled, index group for fitting will be prompted. Otherwise, fitting will be performed with RMSD/clustering group.

-fit2central/-nofit2central

Enable/Disable trajectory superimposition or fitting to central structure in the output trajectory. Atoms group used for fitting depends on `-[no]fit` option. If `-nofit`, second input index group (RMSD/clustering group) will be used for fitting otherwise third index group will be used for fitting.

-outframe -1

Number of maximum frames in the output trajectories. It can be helpful to get output trajectory with only structures around the central structure.

-sort none

Sort trajectory according to these values.

Accepted methods are:

- `-sort none` : Ouput trajectory will not be sorted
- `-sort rmsd`

  Sort trajectory according to RMSD with respect to central structure. Therefore, obtained trajectory's first frame will be central structure and RMSD will increase gradually after first frame.

- `-sort rmsdist`

  Sort trajectory according to distance-matrix RMSD with respect to central structure. Therefore, obtained trajectory's first frame will be central structure and distance-matrix RMSD will increase gradually after first frame.

- `-sort features`

  Sort trajectory according to features sub-space. Distance of each conformation to respective central structure is calculated in feature-space and Trajectory is written from lowest to highest distance. In this trajectory, first frame will be central structure.

  This option is very useful when features are other than PCA's projections of eigenvector.

- `-sort user`

  Sort trajectory using values supplied by user. Not yet implemented.

-plot pca_cluster.png

To plot features vs featrues with clusters in this file.

Plot is generated where feature-vs-feature are depicted with different clusters as colors. It is helpful in checking whether number of clusters is enough.

See also:

Similar types of plots can be obtained using featuresplot sub-command.

featuresplot

Description

Features vs Features plot

This can be used to generate plots for features vs features data. These type of plots are useful to check quality of clustering.

`gmx_clusterByFeatures cluster` with `-plot` option also produces features vs features plot. However, the obtained plot is fixed and cannot be changed. Therefore, this sub-command can be used to obtained plots for desired features with several different options to customize the plot.

Command summary

```
gmx_clusterByFeatures featuresplot  [-h] [-i radius.dat]
                                    [-feat features.xvg]
                                    [-clid clid.xvg] [-o output.png]
                                    [-b 0] [-e -1] [-tmargin 0.1]
                                    [-lcols 5] [-fs 18] [-wd 8] [-ht 10]
                                    [-dpi 300]
```

Options

-i input.txt, --input input.txt

Name of input text file. It should contains two features and their respective labels in each row. All these values should be separated by comma. Each row in file should be in following format:

```
[feature no. at X-axis], [feature no. at Y-axis],[X-Label],[Y-Label]
```

For example, following input will result in four plots:

```
1,2,PC-1,PC-2
2,3,PC-2,PC-3
1,3,PC-1,PC-3
1,4,PC-1,PC-4
```

-feat features.xvg, --features features.xvg

Input features file. This file should be same as supplied to `gmx_clusterByFeatures cluster` with `-feat` option.

-clid clid.xvg, --cluster-id clid.xvg

Input file containing cluster-id as a function of time. The number of frames in this file should be same as in features file.

-o output.png, --output output.png

Name of the output plot file. The extension will be used to determine the output format.

Following output formats (system dependent) might be available:

- ps : Postscript
- eps : Encapsulated Postscript
- pdf : Portable Document Format
- pgf : PGF code for LaTeX
- png : Portable Network Graphics
- raw : Raw RGBA bitmap
- rgba : Raw RGBA bitmap
- svg : Scalable Vector Graphics
- svgz : Scalable Vector Graphics
- jpg : Joint Photographic Experts Group
- jpeg : Joint Photographic Experts Group
- tif : Tagged Image File Format
- tiff : Tagged Image File Format

Note: To list the output formats, use `gmx_clusterByFeatures holeplot -h`.

-b 0, --begin 0

First frame in time to read from the input file

-e -1, --end -1

Last frame in time to read from the input file. By default (`-e -1`), all frames till the end will be read.

-tmargin 0.1, --top-margin 0.1

Margin at top side of the plot. If legends overflow into the plot area, margin can be increased to fit the legend.

-lcols 5, --legend-cols 5

Number of legend columns. If legend overflow the plot area, legends can be made of more than one rows by limiting number of columns to accommodate all legends.

-fs 14, --font-size 14

Font-size of all texts in plot

-wd 8, --width 8

Width of plot in inch

-ht 8, --height 8

Height of plot in inch

-dpi 300, --dpi 300

Resolution of plot

distmat

Description

This tool can be used to calculate:

Average distance matrix: It can be used to calculate average minimum-distance matrix of residues between two atom-groups.

MSF/RMSF in distance-matrix: It can be used to calculate either variance (representing MSF) or standard-deviation (representing RMSF) of distance-matrices.

Contact map: It can be used to calculate contact-frequency map over the trajectory for the residues that are within a minimum distance given by `-ct` option value.

Fluctuation in second trajectory with reference to average of first trajectory: To calculate fluctuations (MSF - variance or RMSF - std. deviation in distance-matrix) in a trajectory with respect to average distances from another trajectory, use `-f traj_for_average.xtc` and `-f2 traj_for_rmsf.xtc`. The averages will be calculated from first trajectory `traj_for_average.xtc`. Subsequently, variances and deviation will be calculated for `traj_for_variance.xtc` with respect to previously calculated averages.

Trajectory and pdb for distance-matrix PCA: To speed up the calculation, it uses all available cores of the CPU using multi-threading. Number of threads/cores could be change by "-nt" option.

Command summary

```
gmx_clusterByFeatures distmat [-f [<.xtc/.trr/...>]] [-s [<.tpr/.gro/...>]]
                              [-n [<.ndx>]] [-f2 [<.xtc/.trr/...>]] [-mean [<.dat>]]
                              [-var [<.dat>]] [-std [<.dat>]] [-cmap [<.dat>]] [-pca [
↪<.xtc>]]
                              [-b <time>] [-e <time>] [-dt <time>] [-ct <real>] [-nt
↪<int>]
                              [-gx <int>] [-gy <int>] [-power <real>]
```

Table 5: Options to specify input files to distmat

Option	Default	File type
-f [<.xtc/.trr/...>]	traj.xtc	Trajectory: xtc trr cpt gro g96 pdb tng
-s [<.tpr/.gro/...>]	topol.tpr	Structure+mass(db): tpr gro g96 pdb brk ent
-n [<.ndx>]	index.ndx	Index file
-f2 [<.xtc/.trr/...>]	traj.xtc	Trajectory: xtc trr cpt gro g96 pdb tng

Table 6: Options to specify output files to distmat

Option	Default	File type
-mean [<.dat>]	average.dat	Generic data file containing matrix
-var [<.dat>]	variance.dat	Generic data file containing matrix
-std [<.dat>]	stdeviation.dat	Generic data file containing matrix
-cmap [<.dat>]	contact_map.dat	Generic data file containing matrix
-pca [<.xtc>]	pca.xtc	Trajectory format file containing distance-matrix of each frame

Table 7: Other options to distmat

Option	Default	Description
-b <real>	0	First frame (ps) to read from trajectory
-e <real>	0	Last frame (ps) to read from trajectory
-dt <real>	0	Only use frame when t MOD dt = first time (ps)
-ct <real>	0.4	cut-off distance (nm) for contact map
-nt <int>	All CPU cores	number of threads for multi-threading
-gx <int>	5	Gap between residues along X-axis in distance-matrix for PCA
-gy <int>	1	Gap between residues along Y-axis in distance-matrix for PCA
-power <real>	1	Distances will be raised by this power and then dumped in xtc file

Options to specify input files

-f traj.xtc

Input trajectory file of `xtc` `trr` `cpt` `gro` `g96` `pdb` or `tng` format.

-s topol.tpr

An input structure file of `tpr` `gro` `g96` or `pdb` format. It is **required** if trajectory is given as input.

-n index.ndx

Two index groups from this file will be prompted for selection. Otherwise, default index groups will be prompted for selection.

Minimum-distance matrix will be calculated between the two selected atom-groups.

-f2 traj.xtc

Input trajectory file of `xtc` `trr` `cpt` `gro` `g96` `pdb` or `tng` format.

Second input trajectory. If this trajectory is provided, fluctuations in this trajectory will be calculated with reference to average-distance matrix of first trajectory.

Options to specify output files

-mean average.dat

Output file containing average of minimum-distance matrix.

-var variance.dat

Output file containing variance of minimum-distance matrix over entire trajectory.

-std stdeviation.dat

Output file containing standard-deviation or RMSF of minimum-distance matrix over entire trajectory.

-cmap contact_map.dat

Output file containing contact map over entire trajectory. The contact is determined using the threshold distance given by `-ct` option;

-pca pca.xtc

Output file containing distance-matrices for each snapshot of the trajectory. This file can be used as input to `gmx covar` and `gmx anaeig` for distance matrix PCA.

A dummy pdb file is also dumped to use with `gmx covar` and `gmx anaeig` for distance matrix PCA.

> **Warning:** These two outputs are not real trajectory and pdb file. These two files are dumped as a data-container to use with `gmx covar` and `gmx anaeig`. For more details, see examples.

Other options

-ct 0.4

cut-off distance (nm) for contact map. Minimum distance below this thershold will be considered to be in contact with each other.

-nt 4

Number of parallel threads for distance-matrix computation.

-gx 5

Gap between residues in distance-matrix along **X-axis** dumped with option -pca for further PCA. This gap reduces the distance-matrix size and subsequently speed-up the PCA performance.

Note: This option **ONLY** affect output from `-pca` option.

-gy 1

Gap between residues in distance-matrix along **Y-axis** dumped with option -pca for further PCA. This gap reduces the distance-matrix size and subsequently speed-up the PCA performance.

Note: This option **ONLY** affect output from `-pca` option.

-power 1

Distances will be raised by this power and then dumped in xtc file.

matplot

Description

`distmat` produces several output files containing matrix data. `matplot` can be used to visualize these data as a 2D map plot.

Command summary

```
gmx_clusterByFeatures matplot [-h] [-i distmat.dat] [-o output.png]
                              [-xs 1] [-ys 1] [-xl Residue]
                              [-yl Residue] [-cbl nm] [-a auto]
                              [-cmap binary] [-vmin VMIN] [-vmax VMAX]
                              [-fs 14] [-cbor vertical] [-wd 8] [-ht 8]
                              [-dpi 300]
```

Options

-i distmat.dat, --input distmat.dat

Input file containing matrix-data. This file is obtained as a output from `distmat`.

-o output.png, --output output.png

Name of the output matrix-plot file. The extension will be used to determine the output format.

Following output formats (system dependent) might be available:

- ps : Postscript
- eps : Encapsulated Postscript
- pdf : Portable Document Format
- pgf : PGF code for LaTeX
- png : Portable Network Graphics
- raw : Raw RGBA bitmap
- rgba : Raw RGBA bitmap
- svg : Scalable Vector Graphics
- svgz : Scalable Vector Graphics
- jpg : Joint Photographic Experts Group
- jpeg : Joint Photographic Experts Group
- tif : Tagged Image File Format
- tiff : Tagged Image File Format

Note: To list the output formats, use `gmx_clusterByFeatures matplot -h`.

-xs 1, --x-start 1

First residue number along X-axis. Input file does not contain information about residue number. Therefore, this option can be used to set the number of residues along X-axis.

-ys 1, --y-start 1

First residue number along Y-axis. Input file does not contain information about residue number. Therefore, this option can be used to set the number of residues along Y-axis.

-xl Residue, --x-label Residue

X-axis label

-yl Residue, --y-label Residue

Y-axis label

-cbl (nm), --colorbar-label (nm)

Label for color bar

-a auto, --image-aspect auto

Controls the aspect ratio of the axes. The aspect is of particular relevance for images since it may distort the image, i.e. pixel will not be square.

Following two options are available:

- `-a equal` : Ensures an aspect ratio of 1. Pixels will be square.
- **`-a auto`** [The axes is kept fixed and the aspect is adjusted so] that the data fit in the axes. In general, this will result in non-square pixels.

-cmap binary, --colormap binary

Name of colormap by which matrix image will be colored. To preview the available colormaps, visit .

Following colormaps might be available:

Accent	Blues	BrBG	BuGn	
BuPu	CMRmap	Dark2	GnBu	Greens
Greys	OrRd	Oranges	PRGn	Paired
Pastel1	Pastel2	PiYG	PuBu	PuBuGn
PuOr	PuRd	Purples	RdBu	RdGy
RdPu	RdYlBu	RdYlGn	Reds	Set1
Set2	Set3	Spectral	Wistia	YlGn
YlGnBu	YlOrBr	YlOrRd	afmhot	autumn
binary	bone	brg	bwr	cividis
cool	coolwarm	copper	cubehelix	flag
gist_earth	gist_gray	gist_heat	gist_ncar	gist_rainbow
gist_stern	gist_yarg	gnuplot	gnuplot2	gray
hot	hsv	inferno	jet	magma
nipy_spectral	ocean	pink	plasma	prism
rainbow	seismic	spring	summer	tab10
tab20	tab20b	tab20c	terrain	viridis
winter				

Reverse of the available colormaps are also available with same name suffixed by "_r". For example, reverse of binary colormap is binary_r, reverse of gist_earth is gist_earth_r etc.

Note: To list all available colormaps, use `gmx_clusterByFeatures matplot -h`.

-vmin VMIN, --min-value VMIN

Minimum value to begin color-mapping. If not provided, minimum value of whole matrix will be considered.

-vmax VMAX, --max-value VMAX

Maximum value to end color-mapping. If not provided, maximum value of whole matrix will be considered.

-fs 14, --font-size 14

Font-size of all texts in plot

-cbor vertical, --colorbar-orientation vertical

Orientation of color bar Following keywords are available: * `-cbor vertical` - vertical colorbar at right side * `-cbor horizontal` - horizontal colorbar at top

---

-wd 8, --width 8

Width of plot in inch

-ht 8, --height 8

Height of plot in inch

-dpi 300, --dpi 300

Resolution of plot

hole

Description

It can be used to calculate radius of protein channel/cavity for GROMACS MD trajectory. It uses program to calculate radius of cavity/channel and dumps the output to a text file as a function of time. It also extract channel's outlining residues and dumps to same output file. This output file can be further read to perform final statistical operations and plotting.

Please cite the original publication of hole: O.S. Smart, J.M. Goodfellow and B.A. Wallace (1993). The Pore Dimensions of Gramicidin A. Biophysical Journal 65:2455-2460.

Command summary

```
gmx_clusterByFeatures hole  [-f [<.xtc/.trr/...>]] [-s [<.tpr/.gro/...>]]
                            [-n [<.ndx>]] [-o [<.dat>]] [-pdb [<.pdb>]] [-b <time>]
                            [-e <time>] [-dt <time>] [-tu <enum>] [-[no]fit]
                            [-endrad <real>] [-sample <real>] [-cvect <vector>]
                            [-cpoint <vector>] [-catmid <int>] [-rad <enum>]
```

Table 8: Options to specify input files to hole

Option	Default	File type
-f [<.xtc/.trr/...>]	traj.xtc	Trajectory: xtc trr cpt gro g96 pdb tng
-s [<.tpr/.gro/...>]	topol.tpr	Structure+mass(db): tpr gro g96 pdb brk ent
-n [<.ndx>]	index.ndx	Index file

Table 9: Options to specify output files to hole

Option	Default	File type
-o [<.dat>]	radius.dat	Generic data file containing matrix
-pdb [<.pdb>]	sphpdb.pdb	PDB file for scpheres along the radius

Table 10: Other options to hole

Option	Default	Description
-b <real>	0	First frame (ps) to read from trajectory
-e <real>	0	Last frame (ps) to read from trajectory
-dt <real>	0	Only use frame when t MOD dt = first time (ps)
-tu <keyword>	0	Unit for time values: fs, ps, ns, us, ms, s
-[no]fit	Enable	Enable fitting and superimposition of the atoms groups.
-endrad <real>	5	Radius value (A) after which calculation is stopped.
-sample <real>	0.5	The distance between the planes for the sphere centers.
-cvect <vector>	0 0 1	Vector along the channel
-cpoint <vector>	999 999 999	Coordinate within a channel as seed for channel/cavity.
-catmid <int>	-1	Serial number of atom, whoose coordinate acts as seed for channel/cavity.
-rad <keyword>	bondi	Radius type for atoms.

Options to specify input files

-f traj.xtc

Input trajectory file of `xtc` `trr` `cpt` `gro` `g96` `pdb` or `tng` format.

-s topol.tpr

An input structure file of `tpr` `gro` `g96` or `pdb` format.

-n index.ndx

Two index groups from this file will be prompted for selection. Otherwise, default index groups will be prompted for selection.

Minimum-distance matrix will be calculated between the two selected atom-groups.

Options to specify output files

-o radius.dat

Output file containing channel/cavity axis, radius and outlining residues as a function of time.

-pdb sphpdb.pdb

Output PDB file containing coordinates of sphere inside the channel/Cavity. Radius of these sphere is channel/cavity radius. This file can be used to visualize whether obtained radii are from inside the intended channel/cavity.

Other options

-fit/-nofit

Weather to superimpose structures on reference structure using least-square fitting.

-endrad 5

Radius (A) above which the program regards a result as an indication that the end of the pore has been reached

-sample 0.5

The distance (A) between the planes for the sphere centers

-cvect 0 0 1

This specified a vector which lies in the direction of the channel/cavity.

-cpoint 999 999 999

A point which lies within the channel. If not given, center of mass will be used. This point will be used a seed to start calcualtion for channel/cavity radius.

Note: Due to this option, superimposition of structures on reference structure is necessary.

Note: Conformations changtes during the simulations, therefore, this coordinate may not be inside the cavity. To dynamically select seed coordinate, use `-catmid` option.

-catmid -1

Serial number of atom, which lies within the channel and acts as a seed for channel/cavity. If not given, center of mass will be used. It can be used to assign seed-coordinate dynamically.

-rad bondi

Radius of atoms considered during channel/cavity calculation.

Accepted categories of radii are:

- `bondi` - For all-atom force-fields, this category can be used.
- `amberuni` - For united-atom force-fields, this category can be used.
- `downscaled`
- `hardcore`
- `simple` - For united-atom force-fields, this category can be used.
- `xplor`

These radii are taken from original HOLE implementation. For values of these radii, please follow this .

holeplot

Description

It can be used to generate plots for outputs generated from hole. It generates plot of average radius with standard deviation as a function of axis-points. It also shows the distribution of residues that outlines the channel/cavity.

Command summary

```
gmx_clusterByFeatures holeplot  [-h] [-i radius.dat] [-resplot]
                                [-o output.png] [-vplot]
                                [-csv output.csv] [-xmin XMIN]
                                [-xmax XMAX] [-endrad ENDRAD] [-ax Z]
                                [-gap 1] [-b 0] [-e -1] [-do 90]
                                [-rfreq 50] [-ymin YMIN] [-ymax YMAX]
```

(continues on next page)

(continued from previous page)

```
                                    [-fs 18] [-rlsize 10] [-wd 6] [-ht 6]
                                    [-dpi 300]
```

Options

-i radius.dat, --input radius.dat

Name of input radius file. Radius file shoudl be obtained from `hole` as an output file.

-resplot, --residues-plot

Plot distributions of outlining residues to cavity/channel. By default, these distributions are not plotted. This option enables the plotting of distributions.

-o output.png, --output output.png

Name of the output plot file. The extension will be used to determine the output format.

Following output formats (system dependent) might be available:

- ps : Postscript
- eps : Encapsulated Postscript
- pdf : Portable Document Format
- pgf : PGF code for LaTeX
- png : Portable Network Graphics
- raw : Raw RGBA bitmap
- rgba : Raw RGBA bitmap
- svg : Scalable Vector Graphics
- svgz : Scalable Vector Graphics
- jpg : Joint Photographic Experts Group
- jpeg : Joint Photographic Experts Group
- tif : Tagged Image File Format
- tiff : Tagged Image File Format

Note: To list the output formats, use `gmx_clusterByFeatures holeplot -h`.

-csv output.csv, --out-csv output.csv

Output csv file. The radius as a function of axis-points in csv formatted file. This file can be read in external data-plotting program.

-vplot, --violinplot

In place of normal line-plot, it plots radius distribution as violins. It is useful because this plot gives distribution of radius values over entire trajectory for each axis-points

---

-xmin XMIN, --axis-min XMIN

Minimum value of axis point after which radius value will be considered for plot.

If not supplied, minimum axis value will be extracted from input radius file.

-xmax XMAX, --axis-max XMAX

Maximum value of axis point after which radius value will be discarded from plot.

If not supplied, maximum axis value will be extracted from input radius file.

-endrad ENDRAD, --end-radius ENDRAD

End/Opening radius. If radius is larger than this value, radius will not considered for average calculation and features output. This option value might be equal or less than `-endrad` value supplied with `hole` sub-command.

-ax Z, --axis Z

Principal axis parallel to the channel or cavity.

-gap 1, --gap 1

Gap between axis-points in Angstroms It should be either equal to or larger than `-sample` value supplied with `hole` sub-command.

-b 0, --begin 0

First frame in time to read from the input file

-e -1, --end -1

Last frame in time to read from the input file. By default (`-e -1`), all frames till the end will be read.

-do 90, --data-occupancy 90

Percentage of radius-data occupancy for axis-points. If an axis-point has radius-data less than this percentage of frames, the axis-point will not be considered for average calculation and features output.

This is critical for axis-points, which are at the opening of channel/cavity. In several frames, radius-value could be missing and therefore, `dataOccupancy` threshold could be used to discard those axis points with lots of missing radius values over the trajectories.

-rfreq 50, --residue-frequency 50

Frequency percentage of residue occurrence during the simulations at a given axis points. If frequency is less than this threshold, it will not considered for plotting.

-ymin YMIN, --y-axis-min YMIN

Minimum value at Y-axis. If not supplied minimum value from data will be used. It can be useful to minimum and maximum values of Y-axis when several plots are compared together.

-ymax YMAX, --y-axis-max YMAX

Maximum value at Y-axis. If not supplied maximum value from data will be used. It can be useful to minimum and maximum values of Y-axis when several plots are compared together.

-rlsize 10, --rlabel-size 10

Fontsize of residue label along Y-axis

-fs 14, --font-size 14

Font-size of all texts in plot

-wd 8, --width 8

Width of plot in inch

-ht 8, --height 8

Height of plot in inch

-dpi 300, --dpi 300

Resolution of plot

holeplot

Description

Write channel/cavity radius as features for clustering.

The output file can be used as input features for clustering of channel/cavity shape in cluster.

Command summary

```
gmx_clusterByFeatures holefeatures [-h] [-i radius.dat] [-o output.xvg]
                                   [-pca 5] [-xmin XMIN] [-xmax XMAX]
                                   [-endrad ENDRAD] [-ax Z] [-gap 1]
                                   [-b 0] [-e -1] [-do 90]
```

Options

-i radius.dat, --input radius.dat

Name of input radius file. Radius file should be obtained from `hole` as an output file.

-o output.xvg, --output output.xvg

Name of output file containing radius as function of time at each axis points. This file can be used as features file for clustering. This file can be also used to plot radius vs time with external plotting program.

The file name should end with xvg extension, which is recognized by "cluster" command.

-pca 5, --pca-pcs 5

Number of eigenvectors to be considered for the features. In place for taking radius as features, this option enable PCA of radii and the resultant projections on eigenvectors can be used as features.

-xmin XMIN, --axis-min XMIN

Minimum value of axis point after which radius value will be considered for plot.

If not supplied, minimum axis value will be extracted from input radius file.

-xmax XMAX, --axis-max XMAX

Maximum value of axis point after which radius value will be discarded from plot.

If not supplied, maximum axis value will be extracted from input radius file.

-endrad ENDRAD, --end-radius ENDRAD

End/Opening radius. If radius is larger than this value, radius will not considered for average calculation and features output. This option value might be equal or less than `-endrad` value supplied with `hole` sub-command.

-ax Z, --axis Z

Principal axis parallel to the channel or cavity.

-gap 1, --gap 1

Gap between axis-points in Angstroms It should be either equal to or larger than `-sample` value supplied with `hole` sub-command.

-b 0, --begin 0

First frame in time to read from the input file

-e -1, --end -1

Last frame in time to read from the input file. By default (`-e -1`), all frames till the end will be read.

-do 90, --data-occupancy 90

Percentage of radius-data occupancy for axis-points. If an axis-point has radius-data less than this percentage of frames, the axis-point will not be considered for average calculation and features output.

This is critical for axis-points, which are at the opening of channel/cavity. In several frames, radius-value could be missing and therefore, `dataOccupancy` threshold could be used to discard those axis points with lots of missing radius values over the trajectories.

holeclustersplot

Description

It can be used to plot radius of cavity/channel for clusters separately. It reads radius file from hole and cluster-id file from cluster, and extract radius of each cluster separately and plot them in one plot. This plot could be extremely useful to compare radius along the channel/cavity in all clusters.

Command summary

```
gmx_clusterByFeatures holeclustersplot  [-h] [-i radius.dat]
                                        [-clid clid.xvg] [-o output.png]
                                        [-csv output.csv] [-xmin XMIN]
                                        [-xmax XMAX] [-endrad ENDRAD]
                                        [-ax Z] [-gap 1] [-b 0] [-e -1]
                                        [-do 90] [-stdbar] [-dl 0]
                                        [-ymin YMIN] [-ymax YMAX]
                                        [-rmargin 0.15] [-lcols 1]
                                        [-fs 18] [-wd 6] [-ht 6]
                                        [-dpi 300]
```

Options

-i radius.dat, --input radius.dat

Name of input radius file. Radius file shoudl be obtained from `hole` as an output file.

-clid clid.xvg, --clid clid.xvg

Input file containing cluster-id as a function of time. The number of frames in this file should be same as in input radius file.

-o output.png, --output output.png

Name of the output plot file. The extension will be used to determine the output format.

Following output formats (system dependent) might be available:

- ps : Postscript
- eps : Encapsulated Postscript
- pdf : Portable Document Format
- pgf : PGF code for LaTeX
- png : Portable Network Graphics
- raw : Raw RGBA bitmap
- rgba : Raw RGBA bitmap
- svg : Scalable Vector Graphics
- svgz : Scalable Vector Graphics
- jpg : Joint Photographic Experts Group
- jpeg : Joint Photographic Experts Group
- tif : Tagged Image File Format
- tiff : Tagged Image File Format

Note: To list the output formats, use `gmx_clusterByFeatures holeclustersplot -h`.

`-csv output.csv`, `--out-csv output.csv` Output csv file. The radius as a function of axis-points in csv formatted file. This file can be read in external data-plotting program.

-xmin XMIN, --axis-min XMIN

Minimum value of axis point after which radius value will be considered for plot.

If not supplied, minimum axis value will be extracted from input radius file.

-xmax XMAX, --axis-max XMAX

Maximum value of axis point after which radius value will be discarded from plot.

If not supplied, maximum axis value will be extracted from input radius file.

-endrad ENDRAD, --end-radius ENDRAD

End/Opening radius. If radius is larger than this value, radius will not considered for average calculation and features output. This option value might be equal or less than `-endrad` value supplied with `hole` sub-command.

-ax Z, --axis Z

Principal axis parallel to the channel or cavity.

-gap 1, --gap 1

Gap between axis-points in Angstroms It should be either equal to or larger than `-sample` value supplied with `hole` sub-command.

-b 0, --begin 0

First frame in time to read from the input file

-e -1, --end -1

Last frame in time to read from the input file. By default (`-e -1`), all frames till the end will be read.

-do 90, --data-occupancy 90

Percentage of radius-data occupancy for axis-points. If an axis-point has radius-data less than this percentage of frames, the axis-point will not be considered for average calculation and features output.

This is critical for axis-points, which are at the opening of channel/cavity. In several frames, radius-value could be missing and therefore, `dataOccupancy` threshold could be used to discard those axis points with lots of missing radius values over the trajectories.

-stdbar, --stddev-bar

To show standard deviation as error-bar If it is supplied, standard deviation will be shown as an error-bar in the plot.

-dl 0, --discard-lasts 0

Number of smallest clusters to discard from the plotting. It can be useful to filter out few smallest clusters because these may contain small number of frames.

-ymin YMIN, --y-axis-min YMIN

Minimum value at Y-axis. If not supplied minimum value from data will be used. It can be useful to minimum and maximum values of Y-axis when several plots are compared together.

-ymax YMAX, --y-axis-max YMAX

Maximum value at Y-axis. If not supplied maximum value from data will be used. It can be useful to minimum and maximum values of Y-axis when several plots are compared together.

-rmargin 0.15, --right-margin 0.15

Margin at right side of the plots. If legends overflow into the plot area, margin can be increased to fit the legend.

-lcols 1, --legend-cols 1

Number of legend columns If legend overflow into the plot area, legends can be made of more than one column to accommodate all legends.

-fs 14, --font-size 14

Font-size of all texts in plot

-wd 8, --width 8

Width of plot in inch

-ht 8, --height 8

Height of plot in inch

-dpi 300, --dpi 300

Resolution of plot

2.4 Examples

2.4.1 Clustering ligand conformations using cartesian PCA

In this example, conformation of ligands were clustered with respect to receptor.

At first, PCA was performed using atom-coordinates (cPCA), and subsequently, projections on eigenvectors were used as the features

Atom-coordinates PCA

1. **Covariance, eigenvector and eigenvalue caculcations**

```
echo 13 14 | gmx covar -s input-files/input.tpr -f input-files/trajectory.xtc -n␣
↪input-files/input.ndx
```

Here, `13` is index group of receptor atoms, which were used for superposition by least-square fitting. `14` is index group of ligand without any hydrogen atoms. Above command generated `eigenvec.trr` and `eigenval.xvg` files. `eigenvec.trr` is necessary in next command as input.

2. **Projections on eigenvectors**

```
echo 13 14 | gmx anaeig -s input-files/input.tpr -f input-files/trajectory.xtc -n␣
↪input-files/input.ndx -proj -first 1 -last 20
```

In the above command, `-v eigenvec.trr` was used by default and eigenvectors were read from this file. A new output file `proj.xvg` is generated containing projections on first 20 eigenvectors. This file is used as an input file in `gmx_clusterByFeatures`.

Clustering

```
echo 0 14 13 | gmx_clusterByFeatures cluster -s input-files/input.tpr -f input-files/
↪trajectory.xtc -n input-files/input.ndx \
                                            -feat proj.xvg -method kmeans -nfeature␣
↪20 -cmetric ssr-sst -ncluster 15 \
                                            -fit2central -sort features -cpdb␣
↪clustered-trajs/central.pdb \
                                            -fout clustered-trajs/cluster.xtc -plot␣
↪pca_cluster.png\
```

K-means clustering was used with maximum number of 15 clusters (`-ncluster 15`). It means, clustering were performed 15 times, and in each iteration, starting from two, one more cluster was generated. Subsequently, 9 clusters were accepted as final clusters using change in SSR/SST ratio (`-cmetric ssr-sst` and `-ssrchange 2`)

Note: Check carefully order of index groups selected in the above command.

a. First index group - output in central structures and clustered trajectories

b. Second index group - clustering group, here it is ligand without hydrogen atoms

c. Third group - Used for superposition by least-square fitting.

Outputs

Central structures of each cluster:

Cluster-ID	Central Frame	Total Frames
1	45447	19639
2	51211	15441
3	36523	10488
4	63595	9101
5	70685	6909
6	41378	6157
7	3166	5891
8	21937	4756
9	7755	2166

RMSD (nm) between central structures:

c1	c2	c3	c4	c5	c6	c7	c8	c9
0.000	0.292	0.701	0.444	0.484	0.498	1.076	0.411	0.883
0.292	0.000	0.684	0.428	0.418	0.552	1.063	0.439	0.844
0.701	0.684	0.000	0.834	0.574	0.360	0.860	0.588	0.812
0.444	0.428	0.834	0.000	0.571	0.705	0.940	0.733	0.763
0.484	0.418	0.574	0.571	0.000	0.351	0.947	0.670	0.961
0.498	0.552	0.360	0.705	0.351	0.000	0.959	0.548	0.967
1.076	1.063	0.860	0.940	0.947	0.959	0.000	1.165	0.614
0.411	0.439	0.588	0.733	0.670	0.548	1.165	0.000	0.890
0.883	0.844	0.812	0.763	0.961	0.967	0.614	0.890	0.000

Output files generated:

a. `-g cluster.log` : log output containing information about the clusters.

b. `-clid clid.xvg` : Cluster-id as a function of time.

c. `-fout clustered-trajs/cluster.xtc` : 9 clustered trajectories were extracted with name cluster_c{ID}.xtc

d. `-cpdb clustered-trajs/central.pdb` : 9 central structures PDB files were extracted with name central_c{ID}.pdb

e. `-plot pca_cluster.png` : Plots of feature-vs-feature with different colors as clusters and central structure. This plot can be used for visual inspection of clustering.

2.4.2 Clustering conformations using distances between atoms

In this example, conformations of G-Quadruplex DNA is clustered according to distances between three atom-pairs. These atom-pairs form hydrogen bonds during the simulations. However, either only one atom-pair among them could form hydrogen bond at a time or neither can form hydrogen bond. The formation of hydrogen bonds are extremely fluctuating between these atom-pairs, and therefore, clustering will filter conformations based on these hydrogen bonds and distances between atom-pairs.

Calculation of distances

At first distances between atom-pairs are calculated using `gmx pairdist` tool as follows.

```
gmx pairdist -s input.tpr -f input_traj.xtc -ref "resid 1 and atomname N7"  -sel
↪"resid 17 and atomname H62" -o r1N7-r17H62
gmx pairdist -s input.tpr -f input_traj.xtc -ref "resid 1 and atomname H62" -sel
↪"resid 17 and atomname N3"  -o r1H62-r17N3
gmx pairdist -s input.tpr -f input_traj.xtc -ref "resid 1 and atomname H61" -sel
↪"resid 17 and atomname N1"  -o r1H61-r17N1
```

In next step, all above files are merged to a single file to use as a feature input file.

```
cat r1N7-r17H62.xvg  > distances.xvg
printf "\n& \n\n"   >> distances.xvg
cat r1H62-r17N3.xvg >> distances.xvg
printf "\n& \n\n"   >> distances.xvg
cat r1H61-r17N1.xvg >> distances.xvg
printf "\n& \n\n"   >> distances.xvg
```

Clustering

```
echo 0 1 7 | gmx_clusterByFeatures cluster -s input.tpr -f input_traj.xtc -n input.
↪ndx -feat distances.xvg \
                                          -method kmeans -nfeature 3 -cmetric ssr-
↪sst -ncluster 10 -fit2central \
                                          -sort features -ssrchange 2 -cpdb␣
↪clustered-trajs/central.pdb \
                                          -fout clustered-trajs/cluster.xtc -plot␣
↪features_cluster.png \
```

K-means clustering was used with maximum number of 10 clusters (`-ncluster 10`). It means, clustering will be performed 10 times, and in each iteration, starting from two, one more cluster was generated. Subsequently, 5 clusters were accepted as final clusters using change in SSR/SST ratio (`-cmetric ssr-sst` and `-ssrchange 2`)

Note: Check carefully order of index groups selected in the above command.

a. First index group - output in central structures and clustered trajectories

b. Second index group - RMSD group, here it is whole G-Quadruplex DNA.

c. Third group - Used for superposition by least-square fitting, here it is four tetrads of G-Quadruplex DNA.

Outputs

Central structures of each cluster:

```
Cluster-ID  Central Frame   Total Frames
1           27707           15640
2           19260           8435
3           30851           6338
4           24630           5332
5           39369           4894
```

RMSD (nm) between central structures:

```
c1      c2      c3      c4      c5
0.000   0.512   0.274   0.266   0.401
0.512   0.000   0.483   0.443   0.397
0.274   0.483   0.000   0.259   0.484
0.266   0.443   0.259   0.000   0.385
0.401   0.397   0.484   0.385   0.000
```

Output files generated:

a. `-g cluster.log` : log output containing information about the clusters.

b. `-clid clid.xvg` : Cluster-id as a function of time.

c. `-fout clustered-trajs/cluster.xtc` : 5 clustered trajectories were extracted with name cluster_c{ID}.xtc

d. `-cpdb clustered-trajs/central.pdb` : 5 central structures PDB files were extracted with name central_c{ID}.pdb

e. `-plot features_cluster.png` : Plots of feature-vs-feature with different colors as clusters and central structure. This plot can be used for visual inspection of clustering.

Overall Results:

- Cluster-1: conformations with Hydrogen bonds between A1.N7 and A17.H62 atoms
- Cluster-2: conformations where distance between all these atom-pairs are between 0.5 to 1.5 nm
- Cluster-3: conformations with Hydrogen bonds between A1.H61 and A17.N1 atoms
- Cluster-4: conformations with Hydrogen bonds between A1.H62 and A17.N3 atoms
- Cluster-5: conformations where distance between all these atom-pairs are between 1.5 to 2.5 nm

These results demonstrate that clustering is able to filter out the conformations based on these distances.

2.4.3 Clustering conformations using distance-matrix PCA

In this example, conformational clustering of a flexible protein will be performed using the distance-matrix PCA (dmPCA). This protein is extremely flexible and, superposition of conformations are not accurate that is required during the conventional PCA. Therefore, to avoid the superposition step, distance-matrix can be used in place of atom-coordinates for PCA.

Calculation of distance-matrix

At first, distance-matrix over the trajectory can be calculated using distmat command.

```
echo 3 3 | gmx_clusterByFeatures distmat -f input_traj.xtc -s input.tpr -n input.ndx -
↪pca -gx 5
```

Above command produces two outputs:

- `pca.xtc`: This file is a container for distance-matrices over the entire trajectory in xtc file format. This is **not a real** trajectory file.
- `pca_dummy.pdb`: This is a *dummy* pdb file containing same number of entries as obtained in above xtc file.

Note: `-gx 5` is used to reduce the size of distance-matrix. It means that there is a gap of 4 residues along X-axis in distance-matrix. For example, if a protein contains 100 residues, distance-matrix size is 100x100. If `-gx 5` is used, new size is 20x100.

Note: `-gx` and `-gy` options **ONLY** affect output produced with `-pca` option of `distmat`.

Distance-matrix PCA

The `distmat` produces files `pca.xtc` and `pca_dummy.pdb` in the above command. These two files are compatible to use with GROMACS PCA tools. Following steps are used to perform dmPCA.

1. **Covariance, eigenvector and eigenvalue caculcations**

```
echo 0 0 | gmx covar -f pca.xtc -s pca_dummy.pdb -nofit -nomwa -nopbc
```

Above command generated `eigenvec.trr` and `eigenval.xvg` files. `eigenvec.trr` is neccessary in next command as input.

2. **Projections on eigenvectors**

```
echo 0 0 | gmx anaeig -f pca.xtc -s pca_dummy.pdb -first 1 -last 10 -proj
```

In this command, `-v eigenvec.trr` was used by default and eigenvectors were read from this file. A new output file `proj.xvg` is generated containing projections on first 10 eigenvectors. This file is used as an input file in `gmx_clusterByFeatures`.

Clustering

```
echo 0 3 3 | gmx_clusterByFeatures cluster -s input.tpr -f input_traj.xtc -n input.
↪ndx -feat proj.xvg -method kmeans \
                                          -nfeature 5 -cmetric ssr-sst -ncluster 20 -
↪fit2central -sort rmsdist \
                                          -ssrchange 2 -cpdb clustered-trajs/central.
↪pdb -fout clustered-trajs/cluster.xtc \
                                          -plot pca_cluster.png -rmsd rmsdist/raw.xvg
```

K-means clustering was used with maximum number of 20 clusters (`-ncluster 20`). It means, clustering were performed 20 times, and in each iteration, starting from two, one more cluster was generated. Subsequently, 8 clusters were accepted as final clusters using change in SSR/SST ratio (`-cmetric ssr-sst` and `-ssrchange 2`). RMSD in distance-matrix is used for sorting (`-sort rmsdist`) frames in clustered trajectory to avoid the superposition of structure.

Note: Check carefully order of index groups selected in the above command.

a. First index group - output in central structures and clustered trajectories

b. Second index group - rmsd group, here it is C-alpha atoms of protein. ONLY used in in calculation of RMSD matrix, which is dumped in log file with `-g` option.

c. Third group - Used for superposition by least-square fitting. ONLY used in clustered trajectories to align with central structure.

Note: In the clustered trajectories, conformations are sorted on the basis of `rmsdist` (RMSD in distance-matrix). With both `-sort rmsdist` and `-rmsd` options, rmsdist is calculated for each clustered trajectory.

Outputs

Central structures of each cluster:

Cluster-ID	Central Frame	Total Frames
1	20876	6715
2	4902	5803
3	22646	4958
4	7717	4721
5	8287	3137
6	13989	2791
7	24749	2090
8	13740	1801

Output files generated:

a. `-g cluster.log` : log output containing information about the clusters.

b. `-clid clid.xvg` : Cluster-id as a function of time.

c. `-fout clustered-trajs/cluster.xtc` : 8 clustered trajectories were extracted with name cluster_c{ID}.xtc

d. `-cpdb clustered-trajs/central.pdb` : 8 central structures PDB files were extracted with name central_c{ID}.pdb

e. `-plot pca_cluster.png` : Plots of feature-vs-feature with different colors as clusters and central structure. This plot can be used for visual inspection of clustering.