G4Analysis Documentation

Release 0.1.0

Zhu H.

February 24, 2012

CONTENTS

The MDPackage is a program package write in python mostly base on the MDAnalysis Pacakge. The aim of this package is for editing the input and output files of MD software like Gromacs, Amber, and NAMD. It also used for reading and analyzing the MD trajectory, which mostly in binary format and cannot read directly.

CHAPTER

ONE

PARALLEL_ANALYSIS

1.1 Introduction

Parallel analysis.py is used for calculate the distance and angle between two bases groups. usually a group contain 1, 2 or 4 bases in a plane. The angle is useful to analysis the base stack. Two stack bases usually have a small angle and uctuation. If the opition "–rmsd" used, only one bases group will be selected and the RMSD in z-axis for this group will be calculated.

1.2 Usage

Files

Option	Type	Filename	Description
-p	Input	coor_file	Structure fie: gro pdb etc.
-f	Input	traj_file	Trajectory: xtc trr.
-o	Input	output_file	xvgr/xmgr file.
-i	Input	para_an.in	input parmarter file.

Other options

Option	Type	Value	Description
–rmsd	bool	False	skip Calculate the RMSD of DNA bases groups.
–skip	int	1	Get frames when frame MOD skip = 0
-h	bool	yes	Print help info and quit

1.3 Some details of the Algorithm

CHAPTER

TWO

LEAST SQUARES FITTING PROCEDURES

2.1 Algorithm

Define **S** matrix and **E** matrix which are N*3 matrix, and the N is the number of coordinates. **s_ave** and **e_ave** are two vector which are the average of the cloumn of **S** matrix and **E** matrix. This process fitting the **S** coordinates to **E** coorinates.

First, construct a 3*3 covariance matrix **C** between **S** matrix and **E** matrix using the following formula:

$$C = \frac{1}{N-1}(S^T E - \frac{1}{N} S^T I I^T E)$$

Here **I** is an N*1 column vector consisting of only ones.

From the nine elements of **C**, we subsequently generate the 4*4 real symmetric matrix **M** using the expression:

$$M = \begin{vmatrix} c_{11}+c_{22}+c_{33} & c_{23}-c_{32} & c_{31}-c_{13} & c_{12}-c_{21} \\ c_{23}-c_{32} & c_{11}-c_{22}-c_{33} & c_{12}+c_{21} & c_{31}+c_{13} \\ c_{31}-c_{13} & c_{12}+c_{21} & -c_{11}+c_{22}-c_{33} & c_{23}+c_{32} \\ c_{12}-c_{21} & c_{31}+c_{13} & c_{23}+c_{32} & -c_{11}-c_{22}+c_{33} \end{vmatrix}$$

The (q_1, q_2, q_3, q_4) is the eigenvector corresponding to the largest eigenvalue of matrix **M**. Using the element of the largest eigenvector, the orientation matrix **R** can be established by equation below:

$$R = \begin{vmatrix} q_0q_0+q_1q_1-q_2q_2-q_3q_3 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_2q_1+q_0q_3) & q_0q_0-q_1q_1+q_2q_2-q_3q_3 & 2(q_2q_3-q_0q_1) \\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0q_0-q_1q_1-q_2q_2+q_3q_3 \end{vmatrix}$$

The first column of matrix **R** corresponds to x-axis of the base, the second to y-axis, and the third to z-axis.

The origin coordinate of the base can be calculated using equation below:

$$O = E_{ave} - S_{ave}R^T$$

The E_{ave} is the vector of the average of experimental coordinates, and the S_{ave} is the vector corresponding to the average of the standard coordinates of base.

Now the fitting coordinates can be calculate from the rotation matrix **R** and origin **o** using the formula below:

$$F = SR^T + o$$

The upper algorithm consulted the 3DNA software package.

CHAPTER

THREE

FILE FORMAT LIST

3.1 PDB

3.1.1 Introduction to Protein Data Bank Format

Protein Data Bank (PDB) format is a standard for files containing atomic coordinates. Structures deposited in the Protein Data Bank at the Research Collaboratory for Structural Bioinformatics (RCSB) are written in this standardized format. The short description provided here will suffice for most users. How- ever, those actually creating PDB files should consult the definitive description (see http://www.rcsb.org/pdb/info.html#File_Formats_and_Standards).

The complete PDB file specification provides for a wealth of information, including authors, literature references, and the identification of substructures such as disulfide bonds, helices, sheets, and active sites. Users should bear in mind that modeling programs can be unforgiving of incorrect input formats.

3.1.2 Description

Record Type	Columns	Data	Justfication	DataType
ATOM	1-4	"ATOM"	left	character
	7-11	Atom serial number	right	integer
	13-16	Atom name	left*	character
	17	Alternate location indicator		character
	18-20	Residue name	right	character
	22	Chain identifier		character
	23-26	Residue sequence number	right	integer
	27	Code for insertions of residues		character
	31-38	X orthogonal Angstrom coordinate	right	floating
	39-46	Y orthogonal Angstrom coordinate	right	floating
	47-54	Z orthogonal Angstrom coordinate	right	floating
	55-60	Occupancy	right	floating
	61-66	Temperature factor	right	floating
	73-76	Segment identifier (optional)	left	character
	77-78	Element symbol	right	character
	79-80	Charge (optional)		character

3.2 PQR

This format is a modification of the PDB format which allows users to add charge and radius parameters to existing PDB data while keeping it in a format amenable to visualization with standard molecular graphics programs. The

origins of the PQR format are somewhat uncertain, but has been used by several computational biology software programs, including MEAD and AutoDock. UHBD uses a very similar format called QCD.

APBS reads very loosely-formatted PQR files: all fields are whitespace-delimited rather than the strict column formatting mandated by the PDB format. This more liberal formatting allows coordinates which are larger/smaller than $\pm$ 999 Å.

APBS reads data on a per-line basis from PQR files using the following format:

Field_name Atom_number Atom_name Residue_name Chain_ID Residue_number X Y Z Charge Radius

---

where the whitespace is the most important feature of this format. The fields are:

- Field_name

  A string which specifies the type of PQR entry and should either be ATOM or HETATM in order to be parsed by APBS.

- Atom_number

  An integer which provides the atom index.

- Atom_name

  A string which provides the atom name.

- Residue_name

  A string which provides the residue name.

- Chain_ID

  An optional string which provides the chain ID of the atom. Note chain ID support is a new feature of APBS 0.5.0 and later versions.

- Residue_number

  An integer which provides the residue index.

- X Y Z

  3 floats which provide the atomic coordiantes.

- Charge

  A float which provides the atomic charge (in electrons).

- Radius

  A float which provides the atomic radius (in Å).

Clearly, this format can deviate wildly from PDB due to the use of whitespaces rather than specific column widths and alignments. This deviation can be particularly significant when large coordinate values are used. However, in order to maintain compatibility with most molecular graphics programs, the PDB2PQR program and the utilities provided with APBS (see the Parameterization section) attempt to preserve the PDB format as much as possible.

3.3 GRO

Files with the gro file extension contain a molecular structure in Gromos87 format. gro files can be used as trajectory by simply concatenating files. An attempt will be made to read a time value from the title string in each frame, which should be preceded by 't=', as in the sample below.

A sample piece is included below:

MD of 2 waters, t= 0.0

6

1WATER OW1 1 0.126 1.624 1.679 0.1227 -0.0580 0.0434

1WATER HW2 2 0.190 1.661 1.747 0.8085 0.3191 -0.7791

1WATER HW3 3 0.177 1.568 1.613 -0.9045 -2.6469 1.3180

2WATER OW1 4 1.275 0.053 0.622 0.2519 0.3140 -0.1734

2WATER HW2 5 1.337 0.002 0.680 -1.0641 -1.1349 0.0257

2WATER HW3 6 1.326 0.120 0.568 1.9427 -0.8216 -0.0244

1.82060 1.82060 1.82060

Lines contain the following information (top to bottom):

- title string (free format string, optional time in ps after 't=')
- number of atoms (free format integer)
- one line for each atom (fixed format, see below)
- box vectors (free format, space separated reals), values: v1(x) v2(y) v3(z) v1(y) v1(z) v2(x) v2(z) v3(x) v3(y), the last 6 values may be omitted (they will be set to zero). Gromacs only supports boxes with v1(y)=v1(z)=v2(z)=0.

This format is fixed, ie. all columns are in a fixed position. Optionally (for now only yet with trjconv) you can write gro files with any number of decimal places, the format will then be n+5 positions with n decimal places (n+1 for velocities) in stead of 8 with 3 (with 4 for velocities). Upon reading, the precision will be inferred from the distance between the decimal points (which will be n+5). Columns contain the following information (from left to right):

- residue number (5 positions, integer)
- residue name (5 characters)
- atom name (5 characters)
- atom number (5 positions, integer)
- position (in nm, x y z in 3 columns, each 8 positions with 3 decimal places)
- velocity (in nm/ps (or km/s), x y z in 3 columns, each 8 positions with 4 decimal places)

Note that separate molecules or ions (e.g. water or Cl-) are regarded as residues. If you want to write such a file in your own program without using the GROMACS libraries you can use the following formats:

C format

> "%5d%5s%5s%5d%8.3f%8.3f%8.3f%8.4f%8.4f%8.4f"

3.4 XYZ

The XYZ file format is a chemical file format. There is no formal standard and several variations exist, but a typical XYZ format specifies the molecule geometry by giving the number of atoms with Cartesian coordinates that will be read on the first line, a comment on the second, and the lines of atomic coordinates in the following lines. The file format is used in computational chemistry programs for importing and exporting geometries. The units are generally in Ångströms. Some variations include using atomic numbers instead of atomic symbols, or skipping the comment line. Files using the XYZ format conventionally have the .xyz extension.

3.4.1 Format

The formatting of the .xyz file format is as follows:

<number of atoms>

comment line

atom_symbol1 x-coord1 y-coord1 z-coord1

atom_symbol2 x-coord2 y-coord1 z-coord2

...

atom_symboln x-coordn y-coordn z-coordn

3.4.2 Example

The methane molecule can be described in the XYZ format by the following:

5

methane molecule (in [[Ångström]]s)

C 0.000000 0.000000 0.000000

H 0.000000 0.000000 1.089000

H 1.026719 0.000000 -0.363000

H -0.513360 -0.889165 -0.363000

H -0.513360 0.889165 -0.363000

3.5 amber TOP

3.6 amber CRD

CHAPTER

FOUR

THE MODULES OF MDPACKAGE

4.1 atomlib

4.1.1 Variables

BASE_AG_LIST = [’N9’, ‘C8’, ‘N7’, ‘C5’, ‘C6’, ‘N1’, ‘C2’, ‘N3’...

It’s a list of atoms which used to fit the standard structure for the A and G base.

BASE_CTU_LIST = [’N1’, ‘C2’, ‘N3’, ‘C4’, ‘C5’, ‘C6’]

It’s a list of atoms which used to fit the standard structure for the C,T and U base.

RESIDUE_NAME_LIST = [’A’, ‘T’, ‘C’, ‘G’, ‘DA’, ‘DT’, ‘DC’, ‘DG...

just for nucleic now.

BASE_A_DICT = {‘C2’: (-1.912, 1.023, 0.0), ‘C4’: (-1.267, 3.12...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Adenine base.

BASE_C_DICT = {‘C2’: (-1.472, 3.158, 0.0), ‘C4’: (0.837, 2.868...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Cytosine base.

BASE_G_DICT = {‘C2’: (-1.999, 1.087, 0.0), ‘C4’: (-1.265, 3.17...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Guanine base.

BASE_T_DICT = {‘C2’: (-1.462, 3.135, 0.0), ‘C4’: (0.994, 2.897...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Thymine base.

BASE_U_DICT = {‘C2’: (-1.462, 3.131, 0.0), ‘C4’: (0.989, 2.884...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Uracil base.

BASE_A_array = array([[-1.291, 4.498, 0. ...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Adenine base.

BASE_C_array = array([[-1.285, 4.542, 0. ...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Cytosine base.

BASE_G_array = array([[-1.28900000e+00, 4.55100000e+00, 0...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Guanine base.

BASE_T_array = array([[-1.284, 4.5 , 0. ...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Thymine base.

BASE_U_array = array([[-1.284, 4.5 , 0. ...

The cartesian coordinates of non-hydrogen atoms in the standard reference frames of the Uracil base.

CHAPTER

FIVE

CHANGE LOG

5.1 version 0.1.0

5.1.1 Modules and Scripts

- Parallel_analysis
- Modified_Coor

Modules List

- atomlib

CHAPTER

SIX

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*