Eucb: a C++ program for trajectory analysis

Ioannis G. Tsoulos, Athanassios Stavrakoudis

January 2010

Abstract

`Eucb` is an acronym from Euclidean Computational Biology. Basically, it is a geometrical analysis of protein structures, thus it pays some honor to Euclid of Alexandria, the father of geometry. `Eucb` performs trajectory analysis of molecular dynamics simulations of proteins. The program is written in GNU C++ and it can be installed in any operating system running a C++ compiler.

1 Installation

This is a short quick start guide for the installation of the program and contains some small examples that explain the basic features of if. The user who wants to work with eucb should consult the online manual of the program available at the url `http://195.130.120.154/wiki/index.php/Eucb`.

The program is distributed under `tar.gz` compressed format and it can be downloaded from `http://stavrakoudis.econ.uoi.gr/eucb` under the name `eucb.tar.gz` The source code release is under GPL2 license. The program is written in ANSI C++ and it does not require any external library to be compiled and hence the only requirement for the installation of the program is the compiler GNU C++, which it is distributed freely for the majority of operating systems from the relevant directory `http://www.gnu.org/`. The user should issue the following commands in order to install the program

1. `gunzip eucb.tar.gz`. This command creates the file `eucb.tar`
2. `tar xfv eucb.tar` This command creates the folder `eucb.`
3. cd eucb
4. make

After the above procedure the executable eucb is created and it is located under the subfolder `bin` of the folder `eucb`.

2 Usage

The program requires a series of input files to work properly and it performs the required computations dictated by the command line options. The program produces a series of output files, that are time series files accompanied by statistics, moving averages, histograms and log files. For example the command

```
eucb -psf complex.psf -dcd complex.dcd -pdb complex.pdb
-rmsd noh -seq A,C
```

computes the RMSD of the trajectory `complex.dcd` frames after fitting the structures on the structure of the `complex.pdb` file. Non-hydrogen (heavy) atoms of segments A,C are taken into consideration.

2.1 Input files

`Eucb` requires a series of files in order to work properly:

1. The file with the molecular structure and the associated connectivity (.psf file).
2. The file with the coordinates of the structure (.pdb file).
3. The file with the molecular dynamics trajectory (.dcd file).

Files .pdb and .psf are usually the input files of the simulation. All files must be compatible with each other, for example they must contain the same number of atoms. It is advised that the user uses the same .psf .pdb as prepared for NAMD [1] or CHARMM simulation procedure. This will ensure that eucb treats all input file in a right way.

2.2 Output files

The produced files are stored in the directory where the eucb executable was invoked and hence the user must have write permissions in that directory. File names start with a prefix which is relevant to the required option such as rmsf, rmsd etc. In the name of the file could be information such as the chain name, the atom name etc. After the termination of the computation the program will create a series of files with different extensions. The meaning of these extension is the following:

1. **.dat** The file with time series in columns. The first column is usually the frame number and all the other columns are the computed quantities.
2. **.sda** The file with smoothed time series in the same format as the **.dat** file.
3. **.stat** The file which contains statistics of the measured quantities. These statistics could be: average value, minimum value, maximum value, standard deviation etc. depending on the specified command line option.

4. **.hist** The file with frequencies of the measured quantities, useful for histogram plots.

3 Options

The `eucb` program has a variety of command line options, that are divided into general options and computing options. The general options are used in order to define some flags of the program and the computing options are used to compute some quantities and to produce the required time series files.

3.1 General options

1. **-binangle** A, where A is double precision number. Set as A the frequency count in angle calculation, used in the creation of histogram files.

2. **-bindist** D, where D a double precision number. Set as D the frequency count in distance calculation, used in the creation of histogram files.

3. **-cutoff** P,D,A , where P,D,A are double precision numbers. Specify cutoff values used in many computations (percentage, distance, angle).

4. **-first** F, where F an integer value. Define F as the first atom of the dcd file, from which the computation will be started.

5. **-last** F, where F an integer value. Define F as the last atom of the dcd dile, where the computation will be terminated.

6. **-seq** S, where S a string value. Define a sequence of atoms. Examples of this sequence are a) C, which means all the atoms of chain C, C:1-14 which means all the atoms of the residues 1-14 of the chain C etc. This option is used after a computing option and it used to define a sequence of atoms, where the computing option will be applied.

7. **-skip** F, where F is an integer value. Define F as the amount of frames that will be skipped in every reading action of the dcd file.

8. **-smooth** a,b where a,b are positive integers. The use of the second (b) parameter is optional. If only one parameter is given then the time series data (.dat file) are averaged every a frames and stored at .sda file. If the use supplies two parameters, like -smooth 10,20 hen averaging is applies as follows: first the average value corresponding to frames 1-20 is calculated. Then the average value corresponding to frames 11-30 and so on. Thus there is a 10 frame overlapping in the calculating values.

9. **-smart** skip,distance where skip is the number of frames to be skipped and distance is distance of the outer sphere The application of -smart keyword greatly accelerates the calculations of big structures (more than 100 residues) when there is no significant changes in the global structure.

3.2 Computing options

The most significant computing options of the program are the following:

1. **-angle** atom1-atom2-atom3. Angle between the three user - supplied atoms. For example:

   ```
   eucb -psf protein.psf -dcd protein.dcd -angle
       A:13:N-A:13:HN-A:25:0
   ```

   computes the angle between A13:N, A13:HN and A25:0 atoms.

2. **-bturn**. Scanning for beta turns. Scanning is performed on a four-residue basis. This means that at least four residues must be present in the sequence under investigation. If the sequence contains more than four residues, then a moving window of four residues is applied to the whole sequence. Thus, for $N \geq 4$ residues, there are N-3 possible beta-turns, and all of them are searched one-by-one.

3. **-center1** atomselection1 **-center2** atomselection2. Calculate the distance between two centroids defined by the user using specific string values as atomselections. Possible values for atomselections are:

   (a) **ca**, C^{α} atoms (DEFAULT)

   (b) **noh**, non-hydrogen heavy atoms

   (c) **all**, all atoms (including hydrogens)

   For examples:

   ```
   eucb -psf protein.psf -dcd protein.dcd -center1 ca -seq A
       -center2 ca -seq B -smooth 10
   ```

   calculates the distance between the average position of C^{α} of chain A and the average position of C^{α} of chain B. Data are also averaged every 10 frames (**-smooth** keyword).

4. **-closewater** Search for the closest waters in a series of aminoacids.

5. **-contact1** type1 **-contact2** type2. The program perform analysis of close contacts between heavy atoms that lie in close proximity. In general, three type of contacts are considered: vdw (van der Waals), salt (salt bridges) and hb (hydrogen bonds). This is of course quite general, but also very helpful in order to get an idea about the type and extent of interactions between fragments and/or different chains of protein sequences. The analysis is performed in two levels: an initial conformation in pdb format is analyzed and a trajectory in dcd format, so the comparison is easy and direct. A general example is:

   ```
   eucb -psf protein.psf -pdb protein.pdb -dcd protein.dcd
       -contact1 type -seq sequence -contact2
       type -seq sequence -cutoff P,D
       -center2 ca -seq B -smooth 10
   ```

   where type can be:

(a) **all**, all heavy atoms (default)

(b) **backbone**, backbone heavy atoms

(c) **sidechain**, sidechain heavy atoms

6. **-dihedral** atom1-atom2-atom3-atom4 Calculate dihedral angle between four user - supplied atoms.

7. **-distance** atom1-atom2 Distance between two user - supplied atoms.

8. **-hbonds** Calculation of hydrogen bonds. The general syntax is

   ```
   eucb -psf psffile -dcd dcdfile -hbonds -cutoff P,D,A
   ```

   Hydrogen bonds are calculated with simple geometrical criteria. If the Donor-Acceptor distance is less than D Åand the Donor-Hydrogen-Acceptor angle is bigger than A degrees, where D, A have default values 3.2 Åand 120o respectively. The program `eucb` exports hydrogen files that occur for at least P fraction of frames.

9. **-jhnha** Calculation of 3J coupling constants.

10. **-noe** protons, where protons is a list of comma separated hydrogen atom types, such as HN, HA, HB, etc. This option is used for the calculation of distances that correspond to proton - proton close contacts (NOEs).

11. **-pdbwrite** Write the dcd coordinates into pdb files. For example the command

    ```
    eucb -psf protein.psf -dcd protein.dcd -first 1 -last 100 -pdbwrite
    ```

    will write the coordinates of the system into 100 different pdb files for frames for 1 to 100.

12. **-pdbtors** angle_list The arguments in the angle_list are the same as in the option **-tors**. Enable the search for dihedral angles versus pdb angles. The user should provide a sequence to search using the **-seq** option.

13. **-pdo** n, where n is an integer value. Compute the Pseudo angle of Orientation, which is a measurement of directionality between opposite charged side chain groups. The value of parameter n is the difference in sequence. Default value for n is 2. The program will search for X-Y (or Y-X) pairs, where X=Arg,Lys and Y=Asp,Glu residues, that differ n positions in the sequence. Two distances (proximity) and two dihedral angles (orientation) are measured.

14. **-psfanal** Analyze the psf file displaying information about it. The user can constrain the analysis providing a sequence with the **-seq** option.

15. **-rmsf** Root Mean Square Fluctuation of CA atoms. The results are stored on rmsf_X.dat files, where X is the chain name.

16. **-rmsd** option, Root Mean Square Deviation of backbone using the Kabsch algorithm[2]. The parameter option accepts the following values:

(a) **ca**, selects CA atoms

(b) **backbone3**, backbone atoms N, CA, C

(c) **bacbkbone4**, backbone atoms N, CA, C, O

(d) **backbone**, an alias of backbone3 (DEFAULT value)

(e) **sidechain**, all heavy (non-hydrogen) atoms of side chains

(f) **noh**, all heavy (non-hydrogen) atoms

17. **-salt2** Calculation of salt bridges between charged groups. The atoms in the groups defined by a -seq parameter that should follow the -salt2 option.

18. **-salt3** Calculation of complex salt bridges between charged groups. The atoms in the groups defined by a -seq parameter that should follow the -salt2 option.

19. **-side1** group1 **-side2** group2. Calculation of side chain interactions. The parameters group1 and group2 can accepts the following values:

    (a) aromatic

    (b) aliphatic

    (c) positive

    (d) negative

    (e) hydrophobic

20. **-stack** p,d,a where p,d,a where a is the maximum dihedral angle between the planar/ring side chains, d is the centroid distance between side chains and p is a minimum percentage of the frames that meet the geometrical criteria. This option is used for the calculation of stacking interactions between residues with planar or ring side chains such as Arg, His, Tyr, Phe, Trp and Pro. For example the command

    ```
    eucb -stack 0.1,5,30 -seq A -psf protein.psf -dcd protein.dcd
    ```

    finds all pairs of residues with planar/ring side chains where the distance is less than 5 Åand the dihedral angle defined by the two planar/ring groups is no more than 30o, for at least 10% of the trajectory frames.

21. **-tors** angle_list. The string argument angle_list is a comma separated list of the following strings. Multiple option value can be entered and separated by commas. The **-tors** keyword must be followed by the -seq keyword, thus a sequence is required. The allowed values in the string list are the following:

    (a) **phi**, ϕ backbone torsion

    (b) **psi**, the ψ backbone torsion

    (c) **ome**, the ω backbone torsion

(d) **chi1**, the $\chi 1$ torsion

(e) **backbone**, alias for ϕ, ψ, ω torsions

(f) **all**, (DEFAULT), compute all torsions

22. **-watbridge1** sequence1 -watbridge2 sequence2. Calculation of interactions between polar groups that are mediated by a water molecule.

23. **-help** C, where C a string value. The program displays information about the command C.

4 Examples

A series of examples are listed in order to demonstrate some of the capabilities of the `eucb`. The examples of this directory have been applied to dcd and psf files of a previous work[3]. The relevant files can be downloaded from the site of the program.

4.1 Rmsd calculation example

The command

```
eucb -psf complex.psf -dcd complex.dcd -pdb complex.pdb -
rmsd ca -seq A
```

calculates the RMSD of the CA atoms only of the dcd trajectory over the PDB structure. The command

```
eucb -psf complex.psf -dcd complex.dcd -pdb complex.pdb -
rmsd backbone3 -seq A,B
```

calculates the RMSD of backbone atoms (N,CA,C) of both chains A,B. The results are stored in separate column in the rmsd.dat file.

4.2 Distance calculation example

The command:

```
eucb -psf complex.psf -dcd complex.dcd
     -distance A:13:CB-A:19:CB
```

computes the distance between A13:CB and A19:CB atoms and exports the files:

- dist_A_13_CB_A_19_CB.dat, The time series file.
- dist_A_13_CB_A_19_CB.stat, The file with some basic statistics of the time series.
- dist_A_13_CB_A_19_CB.hist, The frequencies file, used in histogram plots.

4.3 Stacking calculation example

The command

```
eucb -stack 0.1,5,30 -seq A -psf complex.psf -dcd complex.dcd
```

finds all pairs of residues with planar/ring side chains where the distance is less than 5 Åand the dihedral angle defined by the two planar/ring groups is no more than 30o, for at least 10% of the trajectory frames.

5 Implementation issues

In this section the major classes and variables of the program are analyzed and extensibility hints are given.

5.1 Critical structures and variables

The most significant basic classes and structures in the program are the following:

1. **atom** This is the most significant structure of the program and it is used to describe each atom in psf files.
2. **Command** An abstract class, used to describe any computing option of the program. The user must override this class in order to extend the capabilities of `eucb` by adding a new computing feature.
3. **Dcd** This class provides access to dcd files.
4. **Options** This class is used in order to parse the command line of the program and to break it to options.
5. **Team** This class describes the atom sequences, used as a filter in many computing options with the **-seq** option.

The major variables in the program (located in the file `globals.cc`) are the following:

1. **backbone_critical_percent** The minimum percentage of frames used in order to determine the existence of hydrogen bond. The default value is 0.01
2. **backbone_critical_distance** The maximum distance in Åused to determine the existence of hydrogen bond. The default value is 3.4
3. **backbone_critical_angle** The amount of angles used to determine the existence of hydrogen bond. The default value is 120.
4. **bindist** A value used by the program as frequency count in distance calculations. The default value is 0.5 Å

5. **binangle** A value used by the program as frequency count in angle calculations. The default value is 10.0

6. **bindihe** A value used by the program as frequency count in dihedral calculation. The default value is 10.0

7. **command_list** A vector holding a pointer to each computing option that is implemented by the program.

8. **critical_percent** The minimum percentage of frames used in many distance calculations in order to determine the existence of a bond. The default value is 0.05

9. **critical_distance** The maximum distance in Åused in distance calculations in order to determine the existence of a bond. The default value is 5.5

10. **critical_angle** The minimum angle used in many distance calculations in order to determine the existence of a bond. The default value is 120.0

11. **dcd** A dynamic object of the class **Dcd**, holding critical information about the dcd file that has opened for reading and processing.

12. **dcdfile** The name of the dcd file, that has opened.

13. **first** The first frame in the dcd file, from which the processing will be started. The default value is 1.

14. **histflag** A flag variable (with values 0 or 1), determine if the program will print histogram files during the execution of computing options. The default value is 1.

15. **last** The last frame of the dcd file, where the execution of any computing option will be terminated. The default value is the amount of frames inside the dcd file.

16. **psffile** The name of the psf file that has opened.

17. **pdbfile** The name of the pdb file that has opened.

18. **smart_skip** The amount of frames that will be skipped, if the **-smart** option will be used. The default value is 20

19. **smart_distance** The maximum distance that will be used if the **-smart** option will be used. The default value is 6.8 Å

20. **step** The frames that will be skipped in every reading action in the dcd file. The default value is 1.

21. **table** A vector holding all the atoms located in the psf file.

5.2 Adding a new command

In order to add a new command at least the following steps must be executed:

1. The user must write a class that inherits the basic class **Command**. In the new class at least the method `Run()` must be overrides. This method implements the computing capabilities of the new option.

2. The user must add a new record in the file `eucbhelp` located under the `src` subfolder. This new record should explain the capabilities of the new option and the necessary arguments.

3. Extra code must be added in the file `getoptions.cc` and especially in the function `parse_cmd_line()` in order to support the new option.

References

[1] Phillips,J.C. Braun,R. Wang,W. Gumbart,J. Tajkhorshid,E. Villa,E. Chipot,C. Skeel,R.D. Kalé,K. and Schulten,K. (2005) Scalable Molecular Dynamics with NAMD, *Journal of Computational Chemistry*, **26**, 1781-1802.

[2] Kabsch, W. (1976) A solution of the best rotation to relate two sets of vectors, *Acta Crystallographica,* **32**, 922.

[3] Stavrakoudis,A. (2009) A disulfide linked model of the complement protein $C8\gamma$ complexed with $C8\alpha$ indel peptide, *Journal of Molecular Modeling*, **15**, 165-171.