Computer Physics Communications ••• (••••) •••–•••

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Eucb: A C++ program for molecular dynamics trajectory analysis ☆

Ioannis G. Tsoulos [a], Athanassios Stavrakoudis [b,*]

[a] Department of Communications, Informatics and Management, Technological Educational Institute of Epirus, Arta, Greece
[b] Department of Economics, University of Ioannina, Ioannina, Greece

ARTICLE INFO

Article history:
Received 20 June 2010
Received in revised form 20 November 2010
Accepted 30 November 2010
Available online xxxx

Keywords:
Conformational analysis
Molecular dynamics
Protein structure
Trajectory analysis

ABSTRACT

Eucb is a standalone program for geometrical analysis of molecular dynamics trajectories of protein systems. The program is written in GNU C++ and it can be installed in any operating system running a C++ compiler. The program performs its analytical tasks based on user supplied keywords. The source code is freely available from http://stavrakoudis.econ.uoi.gr/eucb under LGPL 3 license.

Program summary

Program title: Eucb
Catalogue identifier: AEIC_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEIC_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 31 169
No. of bytes in distributed program, including test data, etc.: 297 364
Distribution format: tar.gz
Programming language: GNU C++
Computer: The tool is designed and tested on GNU/Linux systems
Operating system: Unix/Linux systems
RAM: 2 MB
Supplementary material: Sample data files are available
Classification: 3
Nature of problem: Analysis of molecular dynamics trajectories.
Solution method: The program finds all possible interactions according to input files and the user instructions. Then it reads all the trajectory frames and finds those frames in which these interactions occur, under certain geometrical criteria. This is a blind search, without a priori knowledge if a certain interaction occurs or not. The program exports time series of these quantities (distance, angles, etc.) and appropriate descriptive statistics.
Running time: Depends on the input data and the required options.

1. Introduction

Analyzing trajectory data is very often the bottleneck in obtaining biological information from computer simulation molecular dynamics. Here we present a new software tool called eucb (**Eu**clidean **c**omputational **b**iology, from the name of the famous Greek mathematician Euclid of Alexandria, known as the "Father of Geometry"). It is written in GNU C++ and can perform a lot of geometrical type calculations. The eucb program does not depend on any external mathematical or graphics library. The program was developed in GNU/Linux, but surely it can be installed in any operating system running the GNU C++ compiler. The distribution comes in a compressed tar ball and it must be compiled before usage. Nevertheless, the only outcome of the compilation is a binary file without the need for additional resource files and hence, the user can execute the program from any directory of the system.

There is a continuing interest in trajectory analysis and several software tools have appeared in the literature during last years [1,14,13,15]. The eucb program has many more options in comparison to other similar programs. It performs the required calculations based on user instructions given by command line options. The program works with NAMD/CHARMM [2] compatible trajectories

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: astavrak@cc.uoi.gr (A. Stavrakoudis).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2010.11.032

and can be used for several type calculations, such as H-bonds, beta-turns, noe type interactions, finding stacking residues, weak interactions like NH/aromatic hydrogen bonds, hydrophobic clusters, water bridged hydrogen bonds, etc. The final outcome of such calculations is a series of files, including time series files, files with descriptive statistics, moving averages, histogram files, etc. The program accepts three files as input:

1. A .psf file, which describes the molecular structure of the system in CHARMM/XPLOR format.
2. A .dcd file, which holds in binary format the molecular dynamics trajectory.
3. A .pdb file, which holds the reference coordinates of the system in PDB format.

The most striking features of the program are:

1. It calculates any type of geometry, distance, angle or dihedral from the MD trajectory and performs all the standard calculations (RMSF, RMSD, standard torsions, etc.).
2. It scans MD trajectories for specific type of interactions, such as hydrogen bonds, hydrophobic interactions, salt bridges, stacking side chains, etc., using only high level keywords, such as hbonds, salt2, salt3, stack, etc.
3. It identifies hydrophobic clusters through the MD trajectory.
4. It calculates the instant water coordination number [5]. Thus, it identifies isolated water molecules from the bulk solvent.
5. It calculates water bridged hydrogen bond interactions.
6. It can calculate NMR related quantities from the MD trajectory, like noe distances and J coupling constants.
7. It export histogram and descriptive statistics of the calculated variables.

The program is highly configurable and all parameters can be customized from the user. Thus, it is expected to facilitate the biological implementation of simulation data. The basic advantage of `eucb` is its high level instructions. The user can ask human type questions like *what are the close contacts between chains A and B in a protein complex*? and can implement this query in one single command, without having to deal with calculations of the specific distances that characterize these interactions. Beyond calculating time series, the program also exports smoothed time series, descriptive statistics and histograms that are suitable for graphs.

2. 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` can consult the online manual of the program available at the URL http://stavrakoudis.econ.uoi.gr/eucb.

The program is distributed under `tar.gz` compressed format and it can be downloaded from http://stavrakoudis.econ.uoi.gr/eucb. The source code release is distributed under the LGPL3 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 sub-folder `bin` of the folder `eucb`.

The program comes with a set of test files. Details about modeling and simulation of the protein complex used in these test files can be found in Ref. [6].

3. 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 protein.psf -dcd protein.dcd -pdb protein.pdb
-rmsd noh -seq A,C
```

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

3.1. Input files

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

1. The file with the molecular structure and the associated atom connectivity (.psf file). All atom names, residue names, chain identifiers, etc. are taken directly by this file in subsequent references to atoms, residue names, residue sequence, etc.
2. The file with the molecular dynamics trajectory (.dcd file).
3. The file with the coordinates of the structure (.pdb file), this file is needed only in calculations with a reference structure.

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 [2] or CHARMM simulation procedure. This will ensure that `eucb` treats all input file in a right way.

3.2. 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. **.log** Summary of the calculations.
2. **.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.
3. **.sda** The file with smoothed (block averages) time series in the same format as the **.dat** file.
4. **.stat** The file which contains descriptive 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. Statistics of angular data are calculated with Yamartino's method [16]. Descriptive statistics are exported for the whole trajectory and for blocks of the trajectory as well.

Please cite this article in press as: I.G. Tsoulos, A. Stavrakoudis, `Eucb`: A C++ program for molecular dynamics trajectory analysis, Computer Physics Communications (2010), doi:10.1016/j.cpc.2010.11.032

5. **.hist** The file with frequencies of the measured quantities (count of frames where certain geometrical criteria are met), useful for histogram plots.
6. **.hist2** In some cases, like calculations of backbone torsions ϕ, ψ it is useful to have a two-dimensional frequencies suitable for contour plots. This type of calculations is exported in .hist2 file.
7. **.histd**, **.hista** When both distance and angle (or dihedral) quantities are calculates (like with -**bturn**, -**stack**, -**hbonds**, etc. keywords) the program produces separated files for distance and angle (or dihedral) frequency statistics in **.histd** and **.hista** files respectively.

4. 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.

4.1. General options

1. -**first** F, where F (integer) is the first frame of the .dcd file to be processed. Default value is 1.
2. -**last** F, where F (integer) is the last frame of the .dcd file to be processed. Default value is the last frame of the .dcd file.
3. -**skip** F, where F (integer) is the number of frames to be skipped from the calculation procedure. Default value is 1.
4. -**seq** S, where S a string value and defines a residue range. The user must supply the chain and residue number in the form chainId:resId. Residue ranges within a protein chain can be identified with the "–" separation. Multiple sequences can be supplied, if separated with comma. The string S must not contain blank space. Examples of this sequence are (a) C, which means all the atoms of the chain C, (b) C:50–60 which means all the atoms of the residues 50–60 of the chain C, (c) C:50, C:55–60 which defines the atoms of residue C:50 and the atoms of residues C:55–60, 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.
5. -**cutoff** P,D,A. P defines the minimum percentage of the trajectory frames (expressed in fraction of unity) that a certain interaction exists in order for the program to export data and statistics. These interactions are defined with geometrical criteria that involve a distance (D) or an angle (A). These three values must be separated with comma, without any white space.
6. -**bindist** D, sets the bin distance (Å) in frequencies of distance calculations, stored in the histogram files (**.hist**, or **.histd**). For example: -**bindist** 0.5 will guide the program to export frequency statistics with 0.5 Å bins.
7. -**binangle** A, sets the bin angle in frequencies in angle calculations, stored in the histogram files (**.hist**, or **.hista**).
8. -**bindihe** D, sets the bin dihedral in frequencies in torsion or dihedral calculations, stored in the histogram files (**.hist**, or **.hista**).
9. -**smooth** F1,F2 where F1, F2 are positive integers. The use of the second (F2) parameter is optional. If only one parameter is given (F1) then the time series data (**.dat** file) are averaged every F1 frames (simple moving average) and the averaged values are stored at the smoothed file (**.sda** file). If the user supplies both parameters, then the averages are calculated every F2 frames with F1 overlapping frames.
10. -**block** F, sets the number of frames to be used for block descriptive statistics calculations, stored in **.stat** files. Default value is 1/10 of the total number of trajectory frames. Thus, the descriptive statistics of the calculated quantities (distances, angles, count of existence, etc.) are calculated every F frames.
11. -**smart** F,D. In some intensive calculations the program needs a neighbor list atoms. In order to avoid the update of this list at every frame for every atom, the use of -**smart** option enables the update of the neighbor list every F every within D distance. It is thus similar to cutoff performed in non-bonded energy calculations of the MD machine. It is advisable that the user "plays" with the combination of the parameters in order to obtain maximum performance results, depending on the type of calculations. If a molecular system undergoes fast conformational changes, then F has to be set in lower values. Also, if a molecular system undergoes big conformational changes, then D has to be set in bigger values. For example, the command:

```
eucb -mol protein -salt3 -seq C -cutoff 0.3,4.5 -smart 20,7
```

will search for complex salt bridges in the chain C. A certain complex salt bridge is accepted if it occurs for at least 30% of the trajectory frames where the atom distance is less than 4.5 Å. For every charged side chain atom, such as Arg:NE, or Glu:OE1, the program constructs a neighbor list of other charged atoms. Such an atom enters the list if it lies within 7 Å of the reference atom. The searching is performed only within the atoms in the neighbor list. The program updates the list every 20 frames.

4.2. Computing options

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

1. -**angle** atom1-atom2-atom3. Computes the angle between three atoms. The user must explicitly define the atoms in the form of chainId:resId:atomName. For example, the command:

```
eucb -mol protein -angle A:13:N-A:13:HN-A:25:O
```

computes the corresponding angle (N–H^N–O, that corresponds to a possible hydrogen bond interaction) through the trajectory frames.
2. -**aroHN** P,D,A. Finds possible hydrogen bond interactions of backbone amide groups (HN) and aromatic side chains or residues PHE, TYR or TRP. Here, the program sees an aromatic side chain as hydrogen bond acceptor. It accepts an interaction if the distance is less than D, and the angle less than A for at least P fraction of the trajectory. The distance is calculated as between the nitrogen atom (donor) and the centroid of the aromatic side chain. Th angle is calculated as the angle of the N–H vector and the aromatic plane (vector–plane angle). See [3] for more details about this calculation. For example, the command:

```
eucb -mol protein -aroHN 0.1,4.3,90 -seq C
```

will calculate week hydrogen bonds between backbone amide groups and aromatic side chains. Here, the cutoff distance of 4.3 Å is defined for demonstrating reasons.
3. -**aroHC** P,D,A. Similar as the previously described -**aroHN** keyword, but this ones takes backbone hydrogen atoms (attached to CA atoms) for calculating week hydrogen bond interactions.
4. -**bturn** P,D,A. This option also requires the -**seq** option (see below for details). The program scans the protein sequence

for possible β-turns. The sequence must contain at least 4 residues. The program accepts β-turn if for at least P frames (expressed as fraction of unity) the distance C_i^α–C_{i+3}^α is less than D Å and the dihedral angle C_i^α–C_{i+1}^α–C_{i+2}^α–C_{i+3}^α is less than A degrees. The program also makes a classification of β-turn in type I, II, etc. The program exports (in the .dat file) the distance, the dihedral angle, the type of β-turn, the four backbone dihedral angles of the that are related with a β-turn, the distance and angle of the possible backbone hydrogen bond $i \leftarrow i+3$. The C_i^α–C_{i+3}^α distance frequency statistics are stored in the .histd file. The C_i^α–C_{i+1}^α–C_{i+2}^α–C_{i+3}^α dihedral angle frequencies are stored in the .hista file. The file .hist2 contains two-dimensional frequencies, useful for contour plots. The frequency files are affected by the -bindist and -bindihe options. If the sequence contains more than four residues, then a moving window of four residues is applied to the whole sequence. Thus, for $N \geqslant 4$ residues, there are $N-3$ possible beta-turns, and all of them are searched one-by-one. For example, the command:

```
eucb -mol protein -bturn 0.5,7,90 -seq C:10-20
```

will scan the sequence C:10-20 for possible β-turns that occur for at least 50% of the trajectory frames. The program accepts an occurrence of a β-turn if the distance C_i^α–C_{i+3}^α is less than 7 Å and the dihedral angle C_i^α–C_{i+1}^α–C_{i+2}^α–C_{i+3}^α is less than 90 degrees. For recent applications of this type of calculation see Refs. [9,11].

5. -**center1** atomselection1 -**center2** atomselection2. Calculate the distance between two centroids defined by the user using specific string values as atom selections. The user must also supply a sequence for both centers. Possible values for atomselections are:
   (a) **ca**, C^α atoms (DEFAULT)
   (b) **noh**, non-hydrogen heavy atoms
   (c) **sidechain**, non-hydrogen heavy side chain atoms
   (d) **all**, all atoms (including hydrogens)
   For example, the command:

```
eucb -mol protein -center1 ca -seq A -center2 ca -seq C
```

   calculates the distance between the average position of C^α of chain A and the average position of C^α of chain C.
6. -**contact1** atomtype -**contact2** atomtype. The program performs analysis of close contacts between heavy atoms that lie in close proximity. Atomtype keyword can be one of:
   (a) **noh**, all heavy atoms (default)
   (b) **ca**, C^α atoms
   (c) **backbone**, backbone heavy atoms
   (d) **sidechain**, sidechain heavy atoms
   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 user must also supply a .pdb file, as a reference structure. 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. For example, the command:

```
eucb -mol protein -contact1 sidechain -seq A
-contact2 sidechain -seq C -center2 ca -seq B -cutoff 0.5,4
```

   calculates the contacts between side chain atoms in chains A and C respectively. An interaction (salt or vdw) is accepted if the distance of the corresponding atoms is less than 4 Å for at least 50% of the trajectory frames.
7. -**dihedral** atom1-atom2-atom3-atom4. Computes the dihedral angle between four atoms. The user must explicitly define the atoms in the form of chainId:resId:atomName. For example:

```
eucb -mol protein -dihedral A:12:CA-A:13:CA-A:14:CA-
A:15:CA
```

   computes the corresponding dihedral angle through the trajectory frames.
8. -**distance** atom1-atom2. Computes the dihedral angle between four atoms. The user must explicitly define the atoms in the form of chainId:resId:atomName. For example:

```
eucb -mol protein -distance A:12:CA-A:15:CA
```

   computes the corresponding distance.
9. -**hbonds** Finds the hydrogen bond interactions within a user supplied sequence range. The program searches the .psf file in order to find hydrogen bonds donors or acceptors and constructs two lists. Then it assigns a hydrogen bond according to geometrical criteria. The program merges multiple hydrogen atoms attached to a donor atom (for example $-NH_3$ groups) into one, according to the minimum acceptor–hydrogen distance. A hydrogen bond is assigned is the Donor–Acceptor distance is less than a cutoff value (in Å) and the Donor–Hydrogen–Acceptor angle is bigger than a cutoff value. Default values are 3.3 Å and 120° respectively. For example, the command:

```
eucb -mol protein -hbonds -seq A,C -cutoff 0.5,3.3,145
```

   computes the hydrogen bond interactions between atoms in all residues of chains A or C (both intra-chain and inter-chain interactions are considered). In the above example, a pair of atoms is considered to have a hydrogen bond interaction if, for at least the 50% of the trajectory frames, the Donor–Acceptor distance is less than 3.3 Å and the Donor–Hydrogen–Acceptor angle is bigger than 145°.
10. -**hpc** The program searches for clusters of residues whose side chains are in close contact and no heavy atom of these side chains is in close contact with any water molecule. A side chain is considered to be non-hydrated if there is no water oxygen atom in distance less than 3.5 Å from any heavy atom of the side chain. The proximity of side chains is calculated similarly with the -**side** keyword. This type of calculation requires (obviously) the solvated trajectory to be used. For example, the command:

```
eucb -mol complex -hpc -seq C -cutoff 0.5,4 -smart 10,8
```

    will identify residues of the C chain, with non-hydrated side chains, which lie in distance of less than 4 Å for at least 50% of the trajectory frames. See Algorithm 3 for more details about this calculation. The program finds networks of such interaction residues and assigns clusters of such residues. Each residue belongs to one cluster. For a recent application of this type of calculation see Ref. [11].
11. -**JHNHA** A,B,C. Back-calculation of ^{3}J coupling constants of backbone HA and HN proton atoms, useful for comparison of NMR derived data. The calculation is based on the Karplus type equation [17]:

$$^3\mathrm{J} = A\cos\left(\phi - \frac{\pi}{3}\right)^2 - B\left(\phi - \frac{\pi}{3}\right) + \mathrm{C}$$

Algorithm 1 The algorithm for the calculation of finds contacts between residues with hydrophobic side chain.

1. Create two list of residues: the first list includes the hydrophobic residues of chain A and the second list includes the hydrophobic residues of chain C.
2. Define the distance $D(a, b, f)$ of two residues a and b as the minimum Euclidean distance between any atom of residue a versus every atom of residue b at the frame f of the trajectory, given by

$$D(a, b, f) = \min_{x \in a,\ y \in b} d(a, b)$$

where d is the so-called Euclidean distance.
3. For every residue a of the first list and for every residue b of the second list
   (a) Set $L = \emptyset$
   (b) For every frame f of the trajectory
      i) Calculate the distance $D(a, b, f)$.
      ii) Set $L = L \cup D(a, b, f)$
   (c) Set C as the percentage of values in L that are below a predefined threshold c.
   (d) If $C \geqslant p$, where p is a predefined critical percentage then store the list L in a separate file as a time series and calculate and store in additional files statistics and histograms.

Algorithm 2 The water–bridge algorithm.

1. Create the list L_1 with the (donor, acceptor) atoms of the first sequence.
2. Create the list L_2 with the (donor, acceptor) atoms of the second sequence.
3. For every element i, $i = 1, \ldots, |L1|$, of L_1
   (a) Set N_i as the list of atoms from L_2, that are neighbors of L_{1i} using smart criteria.
   (b) Set W_i as the list of water atoms, that are neighbors of L_{1i} using smart criteria.
4. Set $S = \emptyset$
5. For every element i, $i = 1, \ldots, |L1|$ of L_1
   (a) If the atom L_{1i} makes water–bridge with any of its neighbors from N_i,
      i) $S = S \cup L_{1i}$
6. Return S as the result

Algorithm 3 The hydrophobic cluster algorithm.

1. Set as L_1 the list of the sidechain residues
2. Set $L_2 = \emptyset$
3. For every item a of L_1
   (a) For every item b of L_1
      i) If items a, b are near (using smart criteria) then $L_2 = L_2 \cup a$
4. Set $L_3 = \emptyset$
5. For every item a of L_2
   (a) Set w_a the water atoms that are near (using smart criteria) to a
   (b) If the set w_a is empty, then $L_3 = L_3 \cup a$
6. Return L_3 as the result

The user can supply the values of A, B and C parameters. Otherwise, the program uses the default values of 6.4, −1.4 and 1.9 respectively. Calculation of ^{3}J coupling constants is related with the backbone dihedral angle ϕ (also appeared in the above equation), so the keyword -**JHNHA** must follow the calculation of backbone dihedral angles with the -**tors** keyword. For example, the command:

```
eucb -mol protein -tors phi,psi -JHNHA -seq A
```

will calculate the backbone dihedral angles of ϕ and ψ and also the corresponding ^{3}J coupling constants of all residues found in chain A. Summary of the calculated ^{3}J coupling constants will appear in the JHNHA.log file. Descriptive statistics of the calculated values will appear in .stat files (one separate file per residue). For a recent application of this type of calculations, see Ref. [7].

12. -**iwcn** P,D,N. Calculation of the instance water count number. The program finds water molecules that lie close to the protein in less than D Å for at least P fraction of the trajectory, that have maximum N other water molecules in their neighborhood. This option is useful for identifying isolated water molecules in protein interfaces. For more details about this calculation see [5]. For example, the command:

```
eucb -mol complex -iwcn 0.1,3.3,0 -seq C:31-40
```

will identify those water molecules that lie in distance less than 3.3 Å from any protein heavy atom for at least 10% of the trajectory frames, and they have not any other water molecules their neighborhood.

13. -**noe** atoms. Calculates close contacts between hydrogen atoms, useful for comparison with NOE derived distance from NMR studies [12]. The **atoms** is comma separated list (enclosed in quotes) with hydrogen atoms of interest. The user must also supply a protein sequence to be searched (-seq keyword). Optionally, the user can also define a cutoff of accepted interactions (percentage, distance). The program will calculate hydrogen–hydrogen distance (D) and assign each interacting pair as strong (D < 2.8 Å), medium (2.8 < D < 3.5 Å) or weak (3.5 < D < 5.5Z Å) noe interaction, according to interatomic distance. A relevant application with more details of this type of calculation can be found at Ref. [7]. For example, the command:

```
eucb -mol protein noe “HA,HN” -seq A -cutoff 0.8,3.5
```

computes the pairs of HA, HN (HA–HA, HN–HN, HA–HN) atoms (backbone hydrogen atoms) of the protein chain A that lie in distance of less than 3.8 Å for at least 80% of the trajectory frames.

14. -**pdbwrite** Write the dcd coordinates into a series of .pdb files. The user must supply a compatible reference file in pdb format. For example, the command:

```
eucb -psf protein.psf -dcd protein.dcd -pdb protein.pdb
-pdbwrite -seq A -last 5
```

will extract the coordinates of the first 5 frames of chain A into separate .pdb files.

15. -**pdbtors** Similar to -**tors** keyword, but instead of calculating the value of torsional angles, it calculates the absolute differences between values of the trajectory frames and values of a reference structure in pdb format. It is useful for direct comparison of a trajectory against a reference structure, for example the starting conformation.
16. -**pdo** n. Computes a dihedral angle of orientation of charged side chains, ARG or LYS from one side and GLU or ASP from the other side. The parameter n stands for the difference in the chain sequence of the two residues needed to define a pdo angle. Default value is 2. The program scans the sequence of the protein and searches for pairs of ARG–GLU, ARG–ASP, LYS–GLU or LYS–ASP separated by n residues. Then it computes the CB–CB distance and the ARG:NE-GLU:CD, ARG:NE-ASP:CG, LYS:NZ-GLU:CD or LYS:NZ-ASP:CD distance, depending on the case, as a metrics of side chain separation in space. It also computes the dihedral angle CB–CA–CA–CB and X–CA–CA–Y, where X, Y are the side chain atoms ARG:NE, LYS:NZ, GLU:CD or ASP:CD. For example, the command:

```
eucb -mol protein -pdo 2 -seq C
```

will search the sequence of chain C and find pairs of opposite charged residues the have parallel orientation of their side chains. For a recent application of this type of calculation, see Ref. [8].

Please cite this article in press as: I.G. Tsoulos, A. Stavrakoudis, Eucb: A C++ program for molecular dynamics trajectory analysis, Computer Physics Communications (2010), doi:10.1016/j.cpc.2010.11.032

17. -**psfanal** Analyze the .psf file, number of chains, residue statistics, etc.
18. -**rmsf** Root Mean Square Fluctuation of CA atoms. The program exports one file per protein chain. For example, the command:

```
eucb -mol protein -rmsf -seq A,C
```

will calculate the RMSF of CA atoms.
19. -**rmsd** option, Root Mean Square Deviation of backbone using the Kabsch algorithm [4]. 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
For example, the command:

```
eucb -mol protein -rmsd backbone3 -seq A,C
```

will calculate the RMSD of the backbone atoms of chain A, C. RMSD values are exported at the rmsd.dat file, one column per chain plus a column for the total atoms (all chains).
20. -**salt2** Calculation of salt bridges between charged side chain groups. The program searches for possible positive–negative interactions. It merges the multiple atoms of charged groups (for example OD1, OD2 in ASP residue, or NE, NH1, NH2 in ARG residue) into one, according to their minimum distance. The user can optionally supply a sequence (-**seq**) to be searched for salt bridges, or define a cutoff (-**cutoff** percentage,distance) for accepting salt bridges. For example, the command:

```
eucb -mol protein -salt2 -cutoff 0.5,4
```

will search for possible salt bridges in the entire sequence and will accept those that have distance less than 4 Å for at least 50% of the trajectory frames.
21. -**salt3** Similar to the previously described -**salt2** keyword, but it searches for complex salt bridges. For example, the command:

```
eucb -mol protein -salt3 -cutoff 0.5,4
```

will search for possible salt bridges in the entire sequence and will accept those that have distance less than 4 Å for at least 50% of the trajectory frames. For a recent application of this type of calculation see Ref. [11].
22. -**side1** type -**side2** type. Calculation of side chain interactions based on distance between side chain atoms. The parameter type can be:
(a) aromatic
(b) aliphatic
(c) hydrophobic (either aromatic or aliphatic)
(d) positive
(e) negative
For example, the command:

```
eucb -mol protein -side1 aromatic -seq A -side2 aliphatic
-seq C -cutoff 0.5,4.5
```

will search for possible interactions between residues of chain A with aromatic side chains and residues of chain C with aliphatic side chains, that have distance less than 4.5 Å for at least 50% of the trajectory frames. For recent applications of this type is calculations see Refs. [10,11].
23. -**stack** P,D,A. Find planar residues with "stacking" interactions, thus with parallel arrangement of their planar/ring side chain. Distance (D) calculation of two side chains is based on centroid distance between the corresponding side chains. Dihedral angle (A) is assumed as the dihedral angle between two planes (from the corresponding planar side chains). The program searches the molecular dynamics trajectory file and exports pairs of interacting residues if for at least a fraction of frames (P) the distance between side chains is less than a specified cutoff distance (D) and the dihedral angle between the side chains is bigger than a specified cutoff angle (A). Both geometrical criteria must be met, in order to accept a "stacking" interaction. 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 -mol protein -stack 0.1,5,45 -seq A
```

finds all pairs of residues with planar/ring side chains where the side distance is less than 5 Å and the dihedral angle defined by the two planar/ring groups is no more than 45°, for at least 10% of the trajectory frames. For a recent application of this type of calculation see Ref. [7].
24. -**tors** anglelist. The string argument anglelist is a comma separated list of the following strings. Multiple option values 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) **chi**2, the $\chi 2$ torsion
(f) **chi**3, the $\chi 3$ torsion
(g) **chi**4, the $\chi 4$ torsion
(h) **backbone**, alias for ϕ, ψ, ω torsions
(i) **all**, (DEFAULT), compute all torsions
For example, the command:

```
eucb -mol protein -tors backbone -seq A,C
```

will compute all the standard ϕ, ψ, ω backbone torsion angles for all residues found in protein chains A or C. Along with time series data (.dat files) statistics (.stat) and histograms will be exported. The files with extensions .hist2 are useful for plotting the Ramachandran maps. For recent application of this type of applications see Refs. [7,9].
25. -**watbridge1** sequence1 -**watbridge2** sequence2. The program searches for possible occurrence of water bridged hydrogen bonds between residues of the two sequences. See Algorithm 2 for more details about the calculation. For example, the command:

```
eucb -mol complex -watbridge1 -seq A -watbridge2
-seq C -cutoff 0.2,3.3,120 -smart 10,8
```

will search for possible water bridges between residues of chain A and residues of chain C. It will accept those interactions that occur for at least 20% of the trajectory frames. This option, can be used with the -**hbonds** option for a more integrated analysis of hydrogen bond interactions. This type of calculation requires substantial CPU time, and can consume considerable amount of physical memory. The -**smart** option enables the update of the neighbor list between protein atoms

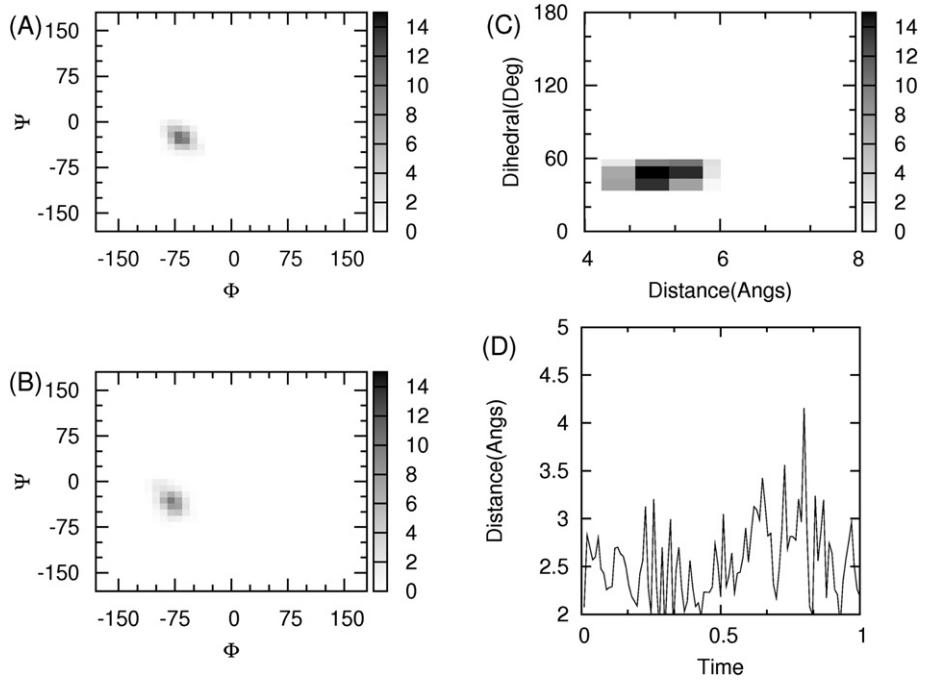

Fig. 1. Characterization of a β-turn in C8γ sequence (see Ref. [6]). Ramachandran plot of ϕ, ϕ dihedral angles of residues Ile_{13C} (A) and Ser_{14C}\$ (B), contour plot of the $Pro_{12C}:C^{a}$–$Thr_{15C}:C^{a}$ distance (D) and $Pro_{12C}:C^{a}$–$Ile_{13C}:C^{a}$–$Ser_{14C}:C^{a}$–$Thr_{15C}:C^{a}$ dihedral angle (C).

and water molecules every 10th frame, based on a distance cut-off of 8 Å.

26. -**help** C, where C is a string value, corresponding to a keyword/option of the `eucb` program. The program displays information about the keyword C.

5. Example runs

We are taking a previous study [6] as a test case to explore the capabilities of the `eucb` software. The accompanied files can be downloaded from the relevant `eucb` site. The target here is to investigate the hydrophobic interactions between the protein chains A and C. This is not at all a trivial task with the available software tools and involves considerable user effort. With `eucb` this can be simplified as a single command. The program interprets high level instructions to all necessary calculations (see Algorithm 1) for an algorithm representation of the calculations and export time series, statistics, histograms of interaction residues. For example, the command:

```
eucb -psf protein.psf -dcd protein.dcd -side1 hydrophobic
-seq A -side2 hydrophobic -seq C -cutoff 0.7,4
```

finds contacts between residues with hydrophobic side chain with the interacting distance to be less than 4 Å for at least the 70% of the trajectory frames. This command produces a series of files and the name of each file is composed with the prefix side (indicating the sidechain computing option), the names of the residues and suffix with the following meaning:

- The **.dat** suffix indicates time series data. The file contains 4 columns. These columns are the frame number, the distance between the two atoms in contact and in the last two columns the names of these atoms.
- The **.stat** suffix indicates files with statistics over the time series between the contacting residues (average distance, minimum distance, maximum distance, etc.).
- The **.hist** suffix indicates files with information that it can be plotted as a histogram. The .hist files have three columns, a central distance value, the percentage of frames that have distance in the .dat file near to this central distance and the absolute number of these frames.

The program identifies the interacting residues and calculates the relevant data and statistics, with only one user supplied keyword “hydrophobic”, thus minimizing the coding time from the user perspective.

Another task is to identify the presence of β-turns, which are characterized by several geometrical features. The search for β-turns can be utilized by using the `eucb` software and the `bturn` keyword. For example, the command:

```
eucb -mol protein -bturn -seq C:10-24 -cutoff 0.5,7,90
```

scans the sequence C:10-24 for beta turns. In Fig. 1 we plot two-dimensional statistics of distance and angle for β-turn of residues 12–15 of the chain C.

One difficult task to be performed with other similar MD analysis tools is to identify to the water bridges in hydrogen bond interactions. This task can be easily accomplished with the following command:

```
eucb -mol complex -watbridge1 -seq A -watbridge2 -seq C
-cutoff 0.2,3.3,120 -smart 10,8
```

finds water-bridged hydrogen bond interactions, between chains A and C.

An interesting task is to identify hydrophobic clusters of the protein structure, thus the interaction of residue sidechains without any involvement of water molecules in their neighborhood. This is mainly applied in residues with hydrophobic side chains, but the program can identify isolated polar side chains in the protein structure interior as well. This type of calculation requires

(obviously) the solvated trajectory to be used. For example, the command:

```
eucb -mol complex -hpc -seq C -cutoff 0.5,4 -smart 10,8
```

will identify residues of the C chain, with non-hydrated side chains, which lie in distance of less than 4 Å for at least 50% of the trajectory frames. The program will find 9 clusters of interacting residues in the protein interior.

6. Conclusions

Practitioners of MD simulations can benefit from the use of `eucb` software both by speeding up common type calculations and by performing more elegant queries to MD trajectories. High level instructions have been introduced to facilitate the analysis of MD trajectories from biological perspective. The program is highly configurable, instructions are programmable via shell scripts and new features can be easily added with minimum effort in C++ coding. Bug fixes, new options and code improvements are regularly announced from the `eucb` web site: http://stavrakoudis.econ.uoi.gr/eucb.

References

[1] N.M. Glykos, Carma: A molecular dynamics analysis program, Journal of Computational Chemistry 27 (2006) 1765–1768.
[2] J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, K. Kalé, K. Schulten, Scalable molecular dynamics with NAMD, Journal of Computational Chemistry 26 (2005) 1781–1802.
[3] G. Tóth, R.F. Murphy, S. Lovas, Stabilization of local structures by π-CH and aromatic–backbone amide interactions involving prolyl and aromatic residues, Protein Engineering 14 (2001) 543–547.
[4] W. Kabsch, A solution of the best rotation to relate two sets of vectors, Acta Crystallographica 32 (1976) 922.
[5] P.M. Petrone, A.E. Garcia, MHC-peptide binding is assisted by bound water molecules, Journal of Molecular Biology 338 (2004) 419–435
.
[6] A. Stavrakoudis, A disulfide linked model of the complement protein $C8\gamma$ complexed with $C8\alpha$ indel peptide, Journal of Molecular Modeling 15 (2009) 165–171.
[7] A. Stavrakoudis, I.G. Tsoulos, Z.O. Shenkarev, T.V. Ovchinnikova, Molecular dynamics simulation of antimicrobial peptide arenicin-2: β-hairpin stabilization by noncovalent interactions, Biopolymers 92 (2009) 143–155.
[8] A. Stavrakoudis, Conformational studies of the 313–320 and 313–332 peptide fragments derived from the α-IIb subunit of integrin receptor with molecular dynamics simulations, International Journal of Peptide Research and Therapeutics 15 (2009) 263–272.
[9] A. Stavrakoudis, Computational modelling and molecular dynamics simulations of a cyclic peptide mimotope of the CD52 antigen complexed with CAMPATH-1H antibody, Molecular Simulation 26 (2010) 127–137.
[10] V. Tatsis, I.G. Tsoulos, A. Stavrakoudis, Molecular dynamics simulations of the TSSPSAD peptide antigen in free and bound with CAMPATH-1H Fab antibody states: The importance of the β-turn conformation, International Journal of Peptide Research and Therapeutics 15 (2009) 1–9.
[11] V.A. Tatsis, I.G. Tsoulos, C.S. Krinas, C. Alexopoulos, A. Stavrakoudis, Insights into the structure of the PmrD protein with molecular dynamics simulations, International Journal of Biological Macromolecules 44 (2009) 393–399.
[12] D. Trzesniak, A. Glättli, B. Jaun, W.F. van Gunsteren, Interpreting NMR data for β-peptides using molecular dynamics simulations, Journal of the American Chemical Society 127 (2005) 14320–14329.
[13] S. Kalat, G. Mann, J. Hermans, Qmd-plot: A graphical utility for rapid preliminary analysis of time series of fluctuating data, developed in the context of molecular dynamics simulations, Journal of Computational Chemistry 23 (2001) 184–188.
[14] T. Verstraelen, M. Van Houteghem, V. Van Speybroeck, M. Waroquier, MDTRACKS: A productive solution for the advanced analysis of molecular dynamics and Monte Carlo simulations, Journal of Chemical Information and Modeling 48 (2008) 2414–2424.
[15] M. Mezei, M. Filizola, TRAJELIX: A computational tool for the geometrical characterization of protein helices during molecular dynamics simulations, Journal of Computer-Aided Molecular Design 20 (2006) 97–107.
[16] R.J. Yamartino, A comparison of several single-pass estimators of the standard deviation of wind direction, Journal of Climate and Applied Meteorology 23 (1984) 1362–1366.
[17] A.C. Wang, A. Bax, Reparametrization of the Karplus relation for 3J(H.alpha.-N) and 3J(HN-C') in peptides from uniformly 13C/15N-enriched human ubiquitin, Journal American Chemical Society 117 (1995) 1810–1813.

Please cite this article in press as: I.G. Tsoulos, A. Stavrakoudis, `Eucb`: A C++ program for molecular dynamics trajectory analysis, Computer Physics Communications (2010), doi:10.1016/j.cpc.2010.11.032