Conformational ENtropy CALCulations

CENCALC Users' Manual

Version 0.2.2

Ernesto Suárez Álvarez
ernesto.suarez.a@gmail.com

December 2012

Preface

This software has been designed to estimate the conformational entropy of single molecules from extended computer simulations, especially Molecular Dynamics (MD) simulations. On input CENCALC needs both trajectory coordinates and topology information in order to characterize the conformational states of the molecule of interest. The molecular conformers are identified by discretizing the time evolution of internal rotations. After this transformation, CENCALC determines the probability mass functions of the individual torsions and uses them for conformational entropy estimations. CENCALC can use up to four different methodologies for approaching to the full conformational entropy: the classical Mutual Information Expansion (MIE), the Approximate MIE (AMIE), the so-called Multibody Local Approximation (MLA), and the default method that corresponds to the correlation corrected MLA (CC-MLA). All of these techniques can also be combined with a distance-based cutoff criterion. In this case, CENCALC requires as additional input an inter-atomic distance matrix containing the mean distance values derived from the MD trajectory in order to include only correlation effects among torsion angles whose mean separation is below a predefined cutoff. The best cutoff for a given amount of sampling can be determined using the CC-MLA method.

All the assumptions and equations defining the various techniques available in CENCALC have been discussed in the literature as well as in the accompanying paper. Users are therefore encouraged to consult those references cited below before using the software.

0-License and Citation

Any use of the CENCALC software or derivative should include at least the following citation:

1) E. Suárez, N. Díaz, J. Méndez and D. Suárez. CENCALC: A Computational Tool for Conformational Entropy Calculations from Molecular Simulations. *Submitted.*

The methods implemented in CENCALC are fully described in the following references:

2) E. Suárez, N. Díaz and D. Suárez. Entropy Calculations of Single Molecules by Combining the Rigid–Rotor and Harmonic-Oscillator Approximations with Conformational Entropy Estimations from Molecular Dynamics Simulations *J. Chem. Theor. Comput.* **2011**, *7*, 2638-2653.

3) E. Suárez, D. Suárez. Multibody Local Approximation: Application to Conformational Entropy Calculations on Biomolecules. *J. Chem. Phys.* **2012**, *137*, 084115.

All questions regarding the usage and distribution of CENCALC or bug reports should be addressed to Ernesto Suárez (esuarez@pitt.edu ; ernesto.suarez.a@gmail.com).

The software, which is distributed under the GNU public license, together with numerical examples and this user's manual are available in the Supporting Information of reference 1. Future releases of the CENCALC software will only be available at:
http://sourceforge.net/projects/cencalc/.

1-Compliling the code

The CENCALC software consists mainly of two standalone codes written in the FORTRAN90 language, `cencalc_prep.f90` and `cencalc_omp.f90`. The first program carries out various preparatory tasks prior to the main entropy calculations that are performed by `cencalc_omp.f90`. As the bulk of the effort to obtain the conformational entropy values is expended by `cencalc_omp.f90`, this program takes advantage of shared-memory parallel computers through the OpenMP Application Program Interface.

Important notice: The FORTRAN90 codes in CENCALC have been compiled and fully tested using the `Linux` version of the **gfortran compiler** (v. 4.4.5):

```
>>  gfortran  cencalc_prep_0.2.2.f90 -o cencalc_prep
>>  gfortran  -fopenmp cencalc_omp_0.2.2.f90 -o cencalc_omp
```

In addition CENCALC also includes a Python program (`get_tor.py`) that reads the topology information from AMBER *parm* files and identifies all the rotatable bonds that are required to characterize the conformational state of the molecule of interest. The `cencalc_prep` and `cencalc_omp` binaries together with the *`get_tor.py`* script should be moved to a run directory included in the `PATH` environment variable.

2-Running the code

2.1-Files needed to perform conformational entropy calculations

The following Table describes the data files, in plain text format, that are needed by CENCALC in order to carry out conformational entropy calculations.

Content	Filename(s)*	Observations

Time series of torsion angles	`d0001.dat`
	`d0002.dat`
	…
	`d000`***`M`***`.dat`
		Mandatory

Square matrix containing the mean values of interatomic distances	`distance_matrix.dat` *`(Format: F9.3)`*	Required only for calculations with cutoff
ID numbers of the central atoms involved in the *M* torsions	`atoms_in_tor.info`	Required only for calculations with cutoff

(*)Default filenames are indicated, but users can choose other filenames using command line options (see below).

In principle the `d0001.dat, d0002.dat, … , d000`*`M`*`.dat` files have each two columns corresponding to the time counter and the torsion angle value, respectively. In any case CENCALC reads only one column per file whose column index can be specified using a command line option (see below). Torsion angles must be specified as dihedral angles measured in degrees. Of course the `d0001.dat, d0002.dat, …` files should all have the same number *N* of lines corresponding to the number of MD snapshots being considered. It may be noted that a time space of 1*ps* between the MD snapshots seems appropriate for most applications of CENCALC. On the other hand, the value of *N* will depend on the dimensionality of the problem as well as on the conformational flexibility of the molecular system: for medium-sized systems one should expect that millions of configurations would be required to reach converged entropies.

The file `atoms_in_tor.info` must have *M* lines, with *M* being the number of torsion data files. If, for example, `d0001.dat` and `d0002.dat` store the time series of the torsion angles defined by the *1-2-3-4* and *2-3-4-5* atom ID numbers, then the first two lines in `atoms_in_tor.info` should be simply *2 3* and *3 4*, respectively. On the other hand, the file `distance_matrix.dat` should

contain, **in format F9.3**, the inter-atomic mean distance matrix of the solute, or at least, the inter-atomic mean distance matrix of the first *K* atoms in the topology, where *K* is greater or equal than the highest atom ID number in `atoms_in_tor.info`. Note that coordinates of solvent molecules and counterions should be better removed before generating `distance_matrix.dat`, otherwise the resulting distance matrix could be huge and very expensive to compute.

Of course CENCALC assumes that a mutually consistent atom numbering is used in the construction of the `atoms_in_tor.info` and `distance_matrix.dat` files.

2.1.1-Obtaining the required input files with the help of `get_tor.py` *(only for AMBER users)*

Users of the *AmberTools* and/or *Amber* packages (http://ambermd.org) can generate automatically all the data files needed by CENCALC using the `get_tor.py` python script. Starting with the MD trajectory and its corresponding *Amber* topology file, `get_tor.py` generates an input file for *ptraj* (the analysis program distributed in *AmberTools*) and the `atoms_in_tor.info` file.

***Usage of get_tor.py* (only for *Amber* topology files)**

SYNOPSIS:

get_tor.py [OPTIONS] *amber_topology.top* [> input_of_ptraj]

OPTIONS:

–h Select only torsion angles involving only heavy atoms (i.e., not Hydrogen)

–b Select only torsion angles defining the backbone conformation of polypeptide molecules.
(Note that –**h** –**b** is partially redundant).

–**s** Opposite to –**b**, select only torsions that are not involved in backbone conformation of polypeptide molecules (*i.e.*, side chain torsions).
(Note that -b and -s are incompatible options)

–**sel** *SELECTION*
Select only torsions involving a given set of residues, for example:
–**sel** 5-20 selects torsions in residues from 5 to 20.
–**sel** 1,5-20,33 as above, but including two more residues: 1 and 33.

–help Print this quick help.

EXAMPLES:

```
>> get_tor.py peptide6.top > input_dihedrals.ptraj
>> get_tor.py -h -s -sel 3,4,7-10  peptide4.top > input.ptraj
```

While identifying potentially rotatable bonds in the topology file, `get_tor.py` makes sure that only one torsion angle X-A-B-Z for each bond A-B is selected in order to avoid redundancy in the entropy calculations.[1]

To illustrate the output generated by `get_tor.py` , let us suppose that a small peptide *Ace-Gly-Ile* that has been simulated in a periodic box of solvent molecules with a topology *parm* file named `pep3.top`. Then the following call to `get_tor.py`

```
>> get_tor.py pep3.top > input.ptraj
```

makes the program to read all the torsion angles of the solute molecule from `pep3.top` and write the following `input.ptraj` file on the standard output:

```
trajin PUT-YOUR-TRAYECTORY-HERE
dihedral  d0001   @6 @5 @7 @9  out  d0001.dat
dihedral  d0002   @5 @7 @9 @12  out  d0002.dat
dihedral  d0003   @13 @12 @14 @16  out  d0003.dat
dihedral  d0004   @12 @14 @16 @18  out  d0004.dat
dihedral  d0005   @7 @9 @12 @13  out  d0005.dat
dihedral  d0006   @24 @18 @16 @31  out  d0006.dat
dihedral  d0007   @20 @18 @24 @27  out  d0007.dat
dihedral  d0008   @18 @16 @31 @32  out  d0008.dat
dihedral  d0009   @4 @2 @5 @6  out  d0009.dat
dihedral  d0010   @26 @24 @27 @28  out  d0010.dat
dihedral  d0011   @23 @20 @18 @24  out  d0011.dat
matrix dist :*@* :*@* out distance_matrix.dat
```

Users should edit manually the `ptraj` input file generated by `get_tor.py` in order to specify the correct path to their MD trajectory file(s). For large periodic systems in explicit solvent, it is strongly recommended to modify the last line corresponding to the *matrix* action by selecting only the solute residues (otherwise, the distance matrix could be huge). For the *Ace-Gly-Ile* example, the *matrix* action command should process only the first 3 residues corresponding to the solute molecule:

```
....
matrix dist :1-3@* :1-3@* out distance_matrix.dat
```

[1] As CENCALC analyzes the results of *classical* molecular simulations, it turns out that individual atoms in a covalently bound molecule are treated as distinguishable particles. Hence, in some particular occasions, it may be necessary to remove from the CENCALC $S_{conform}$ values the conformational entropy that arises from the number of possible rearrangements that a single molecule can formally undergo through internal rotations about bonds to terminal symmetrical groups (*e.g.*, $-CH_3$) without altering any molecular property. See reference 2 for further details.

Finally, execution of *ptraj* would produce the required text files for the time series of the torsion angles (`d0001.dat, d0002.dat, …., d0011.dat`) as well as the interatomic mean distance matrix (`distance_matrix.dat`).

```
> ptraj peptide4.top input.ptraj
```

2.2-Runing `cencalc_prep`

The `cencalc_prep` code estimates first the probability density functions of the *M* torsion angles and characterizes their maxima and minima critical points. Basing on these data, `cencalc_prep` transforms subsequently the initial time series of *N* real numbers per torsion angle θ into a set of *N* integer numbers labeling the conformational states populated by θ. On output, all the information is saved in a file named `MATRIX.dat`, which has *N* rows (*i.e*, the number of MD snapshots) and *M* columns (the number of rotatable bonds). Thus, the *i*–th row of `MATRIX.dat` is an array of integer numbers that represent the conformational state at the *i*–th snapshot. In principle, `MATRIX.dat` should have as many columns as times series read on input (*d0001.dat, d0002, …*). However, `cencalc_prep` removes by default all the frozen torsions because they do not represent conformational changes and do not affect the conformational entropy and, consequently, *M* could be lower on output than on input.

In addition `cencalc_prep` reads the `atoms_in_tor.info` and `distance_matrix.dat` files and generates a new distance matrix file named `reduced_dist_matrix.dat` that contains the mean *distance* between every pair of torsions. These distance values are derived by applying the following rules: a) The distance *d(A,A)* between every torsion *A* and itself is zero; b) the distance *d(A,B)* between two different torsions A_1-A_2-A_3-A_4 and B_1-B_2-B_3-B_4 is the mean distance $d(A_i,B_j)$ between the central pair of atoms of both torsions.

A typical execution of `cencalc_prep` assuming default filenames looks like:

`>> cencalc_prep  d????.dat`	**Any other regular expression besides** `d????.dat` **can be used. The program will also look for** `atoms_in_tor.info` **and** `distance_matrix.dat`**.**
`>> cencalc_prep -nocut d????.dat`	**Calculations with no cutoff. The program will not look for the** `atoms_in_tor.info` **and** `distance_matrix.dat` **files.**

For the more general case, we provide the full help of `cencalc_prep`:

Usage of *cencalc_prep*

SYNOPSIS:

cencalc_prep [OPTIONS] *file1.dat file2.dat …*
(Default names should be *d0001.dat, d0002.dat* …)

OPTIONS:

–u/–usecol *NUMCOL* Default: 2
Column number in *file1.dat*, *file2.dat* that contains the times series of the torsion variable. This value is normally 2 since first column often corresponds to the time or the snapshot number variable.

–dist *DIST_MATRIX_FILE_NAME* Default: `distance_matrix.dat`
This variable specifies the filename of the inter-atomic mean distance matrix.

–i/–info *TOR_INFO_FILE_NAME* Default: `atoms_in_tor.info`
This file specifies the atoms involved in the torsion angles (only the two central atoms). For example, if the first row in *TOR_INFO_FILE_NAME* reads as 3 4, then *file1.dat* contains the time series for rotation about the 3–4 bond.

–nocut Default: Use cutoff
Using this option no cutoff will be applied and the options –dist/–info are thereby not needed.

–s/–simplify *yes/no* Default: yes
Remove all frozen torsions.

–k *K_VALUE* Default: 1.0
The k_value ($\hat{\kappa}$) sets the smoothing parameter v in the von–Mises kernel density estimation as proposed in Eq.(7) in *Computational Statistics & Data Analysis*, **2008**, *52*, 3493–3500. By default $\hat{\kappa} = 1$ as this value ensures slightly over-smoothed Probability Density Functions (PDFs) of individual torsion angles, what is convenient for searching critical points.

–ag *yes/no* Default: no
If *yes* then analytic gradients are used for locating the minima critical points of the von-Mises PDFs of individual torsion angles. The default option (*no*) uses instead a sufficiently accurate and fast linear interpolation scheme for PDF gradient evaluation.

–step *STEP_SIZE* Default: 5 (degrees)
PDF minimization step size: $x_{n+1}=x_n$ – STEP_SIZE * GRADIENT

–crit *CONVERGENCE_THRESHOLD* Default: $1.0 \cdot 10^{-4}$
Gradient convergence threshold for the PDF minimizations.

–maxitr *MAX_NUMBER_OF_ITERATIONS* Default: 1000
Maximum number of iteration for PDF minimizations.

–maxconf *MAX_NUMBER_OF_CONFORMERS* Default: 3
Maximum number of conformers (from 2 to 9) generated by internal rotation about a single bond.

–help
Print this quick help

EXAMPLES:

```
>> cencalc_prep –help
>> cencalc_prep f*.dat
>> cencalc_prep –nocut f*.dat
>> cencalc_prep –u 1 –s yes file?.dat
>> cencalc_prep –s no –info /newpath/atoms_in_tor.info file?.dat
```

2.3-Runing cencalc_omp

The `cencalc_omp` program reads the input data from `MATRIX.dat` and `reduced_dist_matrix.dat` and calculates the probability mass function of the torsion angles by means of the maximum likelihood method. Next the program computes the conformational entropy using one of the available methods, which have been fully described in the literature, and that are summarized in the following Table.

Method	Description	Remarks
MIE	This method is based on the classical mutual information expansion and corresponds to Eq. 5 (no cutoff) and Eq. 6 (with cutoff) in reference 1. (See License and Citations above)	The expansion is computed up to a given order *n*. This method can be computationally expensive and is more suitable for small systems.
AMIE	The approximate MIE method (Eq. 8 in reference 1).	AMIE constitutes a computationally efficient reformulation of MIE at the cost of introducing a very small numerical error.
MLA	The Multibody Local Approximation (Eq. 9 in reference 1) takes into account all the *n*-order effects within a predefined cutoff.	If no cutoff is applied (infinite cutoff), MLA would be computationally very expensive as the MLA entropy would be the exact (but biased) sampled entropy.
CC-MLA	The correlation corrected MLA combines the MLA method with an empirical entropy estimator as proposed in reference 2. The cutoff value that minimizes the CC-MLA entropy is the *best* cutoff for a given amount of sampling.	Best used with cutoff. **CC-MLA is the default method in `cencalc_omp` and the recommended one for medium-sized and large systems.**

To take advantage of shared-memory multiprocessor execution, `cencalc_omp` must have been compiled with the *-openmp/-fopenmp* options as above indicated. At runtime, users also need to set two environment variables: `OMP_NUM_THREADS`, which specifies the number of threads, and `KMP_STACKSIZE`, which sets the number of bytes allocated for each OpenMP thread to be used as private stack.

Setting the number of OpenMP threads equal to *N*

```
>> export OMP_NUM_THREADS=N      (for Bash, Bourne and Korn shells)
>> setenv OMP_NUM_THREADS N      (for C or T shells)
```

Setting the `KMP_STACKSIZE` environment variable

```
>> export KMP_STACKSIZE =100m     (for Bash, Bourne and Korn)
>> setenv KMP_STACKSIZE 100m      (for C or T shell)
```

Typically, assigning a 100 MB value to `KMP_STACKSIZE` should be enough for running `cencalc_omp` jobs. If this variable is underestimated, execution of `cencalc_omp` will cause a runtime error and the program stops.

Once that the environment variables are set, the entropy calculation for a given system can be run through a simple command:

```
>> cencalc_omp > output.out
```

In this simple execution `cencalc_omp` does just one entropy calculation with default options (method CC-MLA, cutoff=8Å, etc.) using all data in the `MATRIX.dat` and `reduced_dist_matrix.dat` files. However, `CENCALC` users will probably be more interested in examining the time evolution of conformational entropy. To this end, `cencalc_omp` can compute a series of data points over a time interval as shown in the following example:

```
>> cencalc_omp  -cutoff 7.5 -ns 5000 100000 5000  > output
```

In this case, `cencalc_omp` performs 20 CC-MLA entropy calculations with a 7.5 Å cutoff, starting all of them at the first frame in `MATRIX.dat` and varying the final frame from 5000 to 100000 using a 5000 offset. During the runtime, the program prints periodically information about the progress of the calculations. After completing the task, `cencalc_omp` will print out, on a separate file with default name `TABLE.out`, a short summary of the results in a tabular form suitable for further analysis.

The following box lists all the `cencalc_omp` options as shown by the program quick help (option **–help**).

Usage of `cencalc_omp`

SYNOPSIS:

cencalc_omp [OPTIONS] [> Output_file.out]

OPTIONS:

–c/–cutoff *CUTOFF* Default: 8
A negative value, or a value greater than the maximum number in `reduced_dist_matrix.dat`, is interpreted as infinite cutoff. The same units must be used in specifying the cutoff value and in the input file `distance_matrix.dat` processed by `cencalc_prep`.

–ns *LAST_SNAP_1 LAST_SNAP_2 OFFSET* Default: Use all snapshots
Calculations are run starting at the first frame in `MATRIX.dat` and varying the final frame from *LAST_SNAP _1* to *LAST_SNAP*_2 using the given *OFFSET* . If only *LAST_SNAP _1* is specified, a single calculation is run using frames from 1 to *LAST_SNAP _1*.

–mie Use the MIE method Default: ccmla

–amie Use the AMIE method Default: ccmla

–o/–order *ORDER* Default:4
Maximum order in the MIE or AMIE calculations.
This option can also be used in MLA or CCMLA calculations
in combination with **–e**.

–e Default: Deactivated
Eliminate the *additional* terms up to the given order in –o.

–mla Use the MLA method Default: ccmla

–ccmla Use the CCMLA method Default: ccmla

–t/–table *TABLE_FILE_NAME* Default: `TABLE.out`
Summary of results is printed in *TABLE_FILE_NAME*

–dist *REDUCED_DISTANCE_MATRIX_FILENAME* Default: `reduced_dist_matrix.dat`

–data *DATA_MATRIX_FILENAME* Default: `MATRIX.dat`

–nc *NUMBER_OF_COLUMNS* Default: Use all columns
Set the exact number of columns from `MATRIX.dat` (*i.e.*, torsion angles) to be used in the entropy calculations.

–nor Default: Deactivated
Do not reorder data in `MATRIX.dat` before calculations.

–cs Activate Chao–Shen Entropy Estimator Default: Deactivated

–ml Activate Maximum Likelihood Entropy Estimator Default: Activated

–help Print this quick help.

EXAMPLES:

```
cencalc_omp > output
cencalc_omp -mie -order 3 -c 9 -ns 10000 100000 5000 > output
cencalc_omp -cutoff 7 -ns 10000 100000 5000 -table ccmla_cut7.tab
cencalc_omp -ns 200000
```

The choice of the entropy method is made by specifying one of the corresponding **–mie**, **–amie**, **–mla** or **–ccmla** options, the default one being –**ccmla**. As previously mentioned, any of the entropy options can be combined with a cutoff value (**–c/–cutoff**). Furthermore, the default choice is to use a 8.0 Å cutoff if no explicit cutoff option is provided. In relationship with the method selection, the option **–o** specifies the maximum order (4^{th} order by default) to be used in the expansion methods (MIE and AMIE). However, **–o** applies to the four methods in combination with the **–e** option. When **–e** is specified, `cencalc_omp` can eliminate *additional* entropy terms up to the given order that result from the list–based approach for including correlation effects within a cutoff region (see reference 3 for a full description of the so-called *additional* entropy terms). Most likely, the **–e** option would be used rarely by CENCALC users, although it provides interesting features as obtaining (probably faster) the exact MIE entropy by combining the **-e** and **–amie** options.

Other options of `cencalc_omp` allow users to particularize all the input/output filenames, what is clearly convenient when several calculations are run in the same directory. For example, by using the **–t/–table** option, users make sure that the final summary (`TABLE.out`) is not overwritten by consecutive runs. Similarly, other options like **–dist** and **–data** assign specific filenames to `MATRIX.dat` and `reduced_dist_matrix.dat`. The **–nc** option introduces some flexibility in `cencalc_omp` runs because this option specifies the exact number of columns (*i.e.*, torsion angles) in `MATRIX.dat` to be considered in the entropy calculations. For example, by choosing only a small subset of torsion angles, users can run test jobs very fast. The **–nor** option deactivates the reordering of columns in `MATRIX.dat` that is carried out by default before starting the entropy calculations in order to minimize the small numerical errors of AMIE with respect to MIE (see reference 3).

Finally, the –**ml** and **–cs** options set the desired entropy *estimator*: Thus, **–ml** (maximum likelihood) is the default choice so that the estimation of discrete probabilities for the individual torsion angles is done by taking the most likelihood values and the entropy is computed by the classical Shannon expression. On the other hand, by activating the **–cs** option, the entropy is

computed by the Chao-Shen estimator, both the estimation of the probabilities and the functional form of the Shannon entropy are transformed in order to reduce the entropy bias (Chao, A.; Shen, T.-J. Nonparametric Estimation of Shannon's Index of Diversity When There Are Unseen Species in Sample. *Environmental and Ecological Statistics* **2003**, *10*, 429-443). The Chao-Shen strategy is not recommended for the default entropy method, CC-MLA, but could be used in combination with the rest of the methods especially when the number of snapshots is relatively small.

It may be worth mentioning that currently `cencalc_omp` can use only one of the four entropy methods at runtime (*i.e.*, MIE and MLA entropies cannot be computed in the same run) and always adopts the last value assigned to a given options. For example, if we set the cutoff option twice as in the in the following example

```
>> cencalc_omp  -cutoff 8  -cutoff 7 –o 2 > output
```

the program will only use a 7.0Å cutoff.

2.3.1-Optimizing the cutoff value for the CC-MLA calculations

In reference 3, an empirical strategy for selecting the best cutoff *R* for a given amount of sampling has been proposed that is based on the CC-MLA entropy estimator. Its derivation implies that all the entropy bias in the calculation is entirely due to *false* correlation effects and, consequently, the reliability of the CC-MLA entropies improves as the marginal entropies of all the rotatable bonds level off along the simulation. Thus, before optimizing *R*, users are advised to run at least one previous job using one of the expansion methods at low order and check thus the convergence properties of the marginal entropies. For example:

```
>> cencalc_omp  -amie –o 2 > output
```

Note that although the default cutoff in CENCALC is *R*=8 Å, this is not necessarily the optimal value for the problem of interest. As demonstrated in reference 3, the minimum value of the CC-MLA entropy as a function of *R* corresponds to the best cutoff for the available information, because that value provides the lowest upper bound to the exact entropy. The following *bash* script illustrates a simple strategy for obtaining a plot of the CC-MLA entropy versus *R*:

```
#!/bin/bash
export OMP_NUM_THREADS=8
export KMP_STACKSIZE=20m

for cut in  0 3 4 5 7 8 9 10
do
 cencalc_omp  -c $cut –table ccmla_cut${cut}entry.tab  > ccmla_cut${cut}.out
done
```

In this example, `cencalc_omp` uses the data of the whole MD trajectory available in `MATRIX.dat` (recommended) and performs a single CC-MLA entropy calculation per run.

Once that the optimum *R* cutoff is known, it is also convenient to examine the evolution of the CC-MLA along the trajectory to make sure that a proper convergence is reached. This can be done through a single run of `cencalc_omp` like:

```
>> cencalc_omp -c 7.5 -ns 50000 1000000 50000 -t ccmla_evol_c7.5.tab > ccmla.out
```

which obtains the accumulated entropy values ongoing from the first MD frame to 50000-1000000 using a 50000 offset and an *R* value of 7.5 Å.

2.3.2-Output example

The following box reproduces partially the information printed out by `cencalc_omp` on the standard output. The corresponding job computes the evolution of the CC-MLA entropy with a 7.5 Å along a 1.0 μs MD trajectory (10^6 snapshots saved for analyses) of the *Ace-Gly-Ile-Pro-Phe-Glu-Gln-Arg-Leu-Val* peptide that has a total of 44 rotatable bonds. During the runtime of `cencalc_omp` the program prints out periodically information about the progress of the calculations. In the following, only the beginning of the `ccmla.out` output file is shown.

```
||  _____                                                 |
||  \\    ||                                              |
||   \\          PROGRAM CENCALC                          |
||   //                    v0.2                           |
||  //___||                                               |
||_____________________________________________________|

Reading the distance matrix...
filename: reduced_dist_matrix.dat
Number of columns and rows:    44

Reading the data matrix...
filename: MATRIX.dat
Number of columns:    44

-------------------------------------------
CALCULATING THE ENTROPY
-------------------------------------------

CALC:  1
=========================
Estimator:      Max-Likelihood
Method:           CC-MLA
Cut-Off:            7.500
Num of Snapshots:   50000
Num of columns  :      44
...
10 %
30 %
40 %
50 %
60 %
70 %
```

This is the first of twenty calculations ongoing from `CALC:1` (50000 frames) to `CALC:20` (1000000 frames)

```
80 %
90 %

ESTIMATED ENTROPY (cal/mol-Kelvin)
-----------------------------------------------------
SNAPSHOTS     ENTROPY
    50000     39.2115
DONE

Partial cpu-time for CALC  1:      65.06 seconds    0.00 hours
Partial real-time for CALC  1:       8.82 seconds     0.00 hours
…
…
.    The program continues up to CALC: 20
```

On exit of the program, the `ccmla_evol_c7.5.tab` file contains a two column Table, entropy vs. number of snapshots, that can be readily imported by spreadsheet software for data analysis:

Contents of `ccmla_evol_c7.5.tab`

```
# NumSnap     Entropy
    50000     39.2115
   100000     40.8601
   150000     42.5934
   200000     43.1909
   250000     43.9758
   300000     45.0394
   350000     45.3476
   400000     45.6141
   450000     45.8421
```

3-Software and Example files

Both the software and the example files are zipped into a single compressed file `CENCAL_v02.zip`. The set of example files, which should help guide new users through a typical application of CENCALC, comprises the *Amber* trajectory and topology files for a 5.0 *ns* MD trajectory of the **GNR** decapeptide as well as the corresponding CENCALC input & output files for CC-MLA calculations that are provided for comparative purposes. Note that the small Amber trajectory is only included so that users can have a trial run of CENCALC and that the resulting entropies would not be converged.

Once that `CENCAL_v02.zip`. has been downloaded and extracted using either a Windows utility such as *Winzip* or a Linux utility such as *unzip*, the working directory should contain the following files and folders:

```
COPYING.txt  README.txt  run_evolution.bash  scan_cutoff.bash run_mie_o1.bash

./AMBER_FILES:
gnr.top  gnr_5ns.mdcrd  input.ptraj
```

```
./CENCALC_PREP_INPUT_FILES:
atoms_in_tor.info d0006.dat d0012.dat d0018.dat d0024.dat d0030.dat d0036.dat
d0042.dat d0048.dat d0054.dat d0060.dat distance_matrix.dat
d0001.dat   d0007.dat   d0013.dat   d0019.dat   d0025.dat   d0031.dat   d0037.dat
d0043.dat d0049.dat d0055.dat d0061.dat
d0002.dat   d0008.dat   d0014.dat   d0020.dat   d0026.dat   d0032.dat   d0038.dat
d0044.dat d0050.dat d0056.dat d0062.dat
d0003.dat   d0009.dat   d0015.dat   d0021.dat   d0027.dat   d0033.dat   d0039.dat
d0045.dat d0051.dat d0057.dat d0063.dat
d0004.dat   d0010.dat   d0016.dat   d0022.dat   d0028.dat   d0034.dat   d0040.dat
d0046.dat d0052.dat d0058.dat d0064.dat
d0005.dat   d0011.dat   d0017.dat   d0023.dat   d0029.dat   d0035.dat   d0041.dat
d0047.dat d0053.dat d0059.dat d0065.dat

./CENCALC_OMP_INPUT_FILES:
MATRIX.dat  reduced_dist_matrix.dat

./CENCALC_OMP_OUTPUT_FILES:
ccgla_cut0.out  ccgla_cut3.out  ccgla_cut4.out  ccgla_cut5.out  ccgla_cut7.out
ccgla_cut8.out ccgla_cut9.out ccmla_cut7_evol.out mie_o1_evol.out
ccgla_cut0.tab  ccgla_cut3.tab  ccgla_cut4.tab  ccgla_cut5.tab  ccgla_cut7.tab
ccgla_cut8.tab ccgla_cut9.tab ccmla_cut7_evol.tab mie_o1_evol.tab

./CODE:
cencalc_omp.f90  cencalc_prep.f90  get_tor.py
```

After having compiled the two FORTRAN codes as above described and placed the resulting binaries and the Python script `get_tor.py` in their `PATH`, users may try the suggested test example.

3.1- A step by step guide for carrying out a simple CC-MLA calculation

For *Amber* users, the topology file, `gnr.top`, and the trajectory file `gnr_5ns.mdcrd` that contains a small fraction (5 *ns* saved every 1ps) of the total trajectory of the **GNR** model peptide are provided in the `AMBER_FILES` folder. It is recommended to copy these two files in a temporary folder for running the test. By using **`get_tor.py`**, the `atoms_in_top.info` and an input file for *ptraj* are obtained by entering:

```
>> get_tor.py gnr.top > input.ptraj
```

Both `atoms_in_top.info` and the `ptraj` input should be written in the working directory. The contents of `input.ptraj` should look like:

input.ptraj:

```
trajin PUT-YOUR-TRAYECTORY-HERE

dihedral  d0001   @6 @5 @7 @9  out  d0001.dat
dihedral  d0002   @5 @7 @9 @12  out  d0002.dat
dihedral  d0003   @13 @12 @14 @16  out  d0003.dat..
...
matrix dist :*@* :*@* out distance_matrix.dat
```

To be edited by user

Only the heading lines of the dihedral commands…

Command for obtaining the mean interatomic distances

This file must be edited for replacing the `PUT-YOUR-TRAYECTORY-HERE` chain by `gnr_5ns.mdcrd` (*i.e.*, the filename of the trajectory file). Then, the *ptraj* program distributed in the *Ambertools* suite must be executed:

```
>> ptraj gnr.top input.ptraj
```

After the *ptraj* execution, the working directory should contain 65 files (`d0001.dat`, `d0002.dat`, …, `d0065.dat`) containing the time series of the torsion angles of the **GNR** peptide along 5000 MD snapshots in addition to the `distance_matrix.dat`. Hence, all the files required for running the `centro_prep` program are available, and this can be done simply as follows:

```
>> cencalc_prep d????.dat
```

resulting in the `MATRIX.dat` and `reduced_dist_matrix.dat` files needed by `cencalc_omp`. Note that after the discretization of the torsion angle evolution, only 31 rotatable bonds (*i.e.*, that present some conformational variability during the 5 *ns* sampling) are kept for the entropy calculations.

Before carrying out a more demanding calculation like CC-MLA, it is important to check the convergence properties of the sum of marginal entropies (SME). Note that using `cencalc_omp` there are many ways to obtain the SME. One of them could be to compute the MIE entropy up to the first order using, for instance, the following script called `run_mie_o1.bash`.

run_mie_o1.bash:

```
#!/bin/bash
export OMP_NUM_THREADS=2
export KMP_STACKSIZE=100m

cencalc_omp  -ns 1000 5000 1000 –o 1 -t mie_o1_evol.tab > mie_o1_evol.out
```

Then the time evolution of the SME can be visualized and plotted from the file `mie_o1_evol.tab`. In our particular instance only 5000 snapshots were used and, therefore, entropy convergence cannot be expected. In any case, the next step in the test calculation would be to find the best cutoff for the entropy estimation.

To search for the optimal CC-MLA cutoff, it is recommended to execute the script `scan_cutoff.bash`, which is available at the CENCALC root directory. Users can play with `cencalc_omp` options to alter the width and spacing of the cutoff scanning.

scan_cutoff.bash:

```
#!/bin/bash
export OMP_NUM_THREADS=2
export KMP_STACKSIZE=100m

for cut in  0 3 4 5 7 8 9
do
  cencalc_omp  -c $cut -table ccmla_cut${cut}.tab  > ccmla_cut${cut}.out
done
```

By executing this script, a series of seven files `ccmla_cut?.tab/ccmla_cut?.out` are obtained. By inspecting the entropy values, users can verify that the optimal cutoff is 7 because it gives the lowest entropy value, 27 *cal/mol-K* (and the lowest upper bound to the exact entropy).

Finally, it may be interesting to check the entropy convergence plot with the optimal cutoff. For this purpose, only a single `cencalc_omp` execution is needed as shown in the following script:

run_evolution.bash

```
#!/bin/bash
export OMP_NUM_THREADS=2        #Number of cores
export KMP_STACKSIZE=100m

cencalc_omp  -ns 1000 5000 1000 -cutoff 7 \
         -table ccmla_cut7_evol.tab  > ccmla_cut7_evol.out
```

Note that a poor convergence should be expected due to the very small amount of sampling (5 *ns*) that is included in this test example. The output is included in the `ccmla_cut7_evol.out` and `ccmla_cut7_evol.tab` files.

ccmla_cut7_evol.tab:

```
# NumSnap     Entropy
     1000     23.8052
     2000     25.5658
     3000     26.0759
     4000     26.9043
     5000     27.3973
```

Note: The entropy values are in *cal/mol-K* as specified in the output file `ccmla_cut7_evol.out`.

Acknowledgments

The CENCALC software and manual were made possible by the following Grants: FICyT (Asturias, Spain) IB05-076 and MEC (Spain) CTQ2007-63266.