
WORDOM User's Guide

Version 0.21-rc3

October 2008

Developers: Michele Seeber, Marco Cecchini, Ran Friedman, Stefanie Muff, Francesco Raimondi, Francesco Rao, Gianni Settanni

License: GPL

URL: <http://www.biochem-caflisch.uzh.ch/wordom/>

Maintainer: mseeber@gmail.com

Description: Wordom is a command line tool written to deal with structure and molecular dynamics files. It allows to access the information in such files, convert it, and analyse it. New analysis modules can be easily added.

Contents

1	Introduction	3
1.1	Contacting the authors	3
1.2	Citation Reference	3
1.3	Acknowledgments	4
1.4	Copyright and Disclaimer Notices	4
2	Installation	5
2.1	Note for Gromacs Users	5
3	Files and Selections	7
3.1	File Formats	7
3.2	Selections	7
3.2.1	Wild Cards	8
3.2.2	Pattern Matching	8
3.2.3	What Else?	9
4	Coordinates Manipulation	10
4.1	Extraction of a trj frame to a mol file	10
4.2	Extraction of a frames series from a trj to a trj	11
4.3	Merging of trjs into a single trj	11
4.4	Appending a trj to another trj	11
4.5	Appending a structure's coordinates to a trj	12
4.6	Conversion of a structure to a 1-frame trj	12
4.7	Conversion of file formats	13
4.8	Conversion of a trj to a concatenated xyz	13
4.9	Showing a trj's headers	13
4.10	Modifying a dcd's headers	14
4.11	Summing several trjs to a single trj	14
4.12	Average over a trj	15

5	Trajectory Analyses	16
5.1	Distances	17
5.2	Contacts	17
5.3	Dihedral angles	18
5.4	Hydrogen Bond Detection	18
5.5	Radius of giration	19
5.6	RMSD	19
5.7	DRMS	20
5.8	RMSD- and DRMS-based Clustering	20
5.8.1	2-pass Clustering	22
5.9	Orientational Parameters P1+P2	22
5.10	Principal Component Analysis	23
5.10.1	Principal Component Analysis Projection	23
5.11	Q Entropy	24
5.12	Secondary Structure	25
6	More Analyses	26
6.1	Kinetic Grouping Analysis (KGA)	26
6.1.1	A bit of Theory - kinetic grouping	26
6.1.2	Modules usage: kinetic grouping	28
6.2	Cut-based free-energy profiles (FEPs)	29
6.2.1	Pfoldf (Pfold fast)	29
6.2.2	Mfpt (Mean first passage time)	30
6.2.3	Modules usage: Cut-based free-energy profiles (FEPs)	32
6.2.4	A posteriori equilibration of out of equilibrium simulations	34
6.3	Elastic Network Modes (ENM)	35
6.4	Protein Structure Network Graph (PSG)	35
	Bibliography	35

Chapter 1

Introduction

Wordom is a (simple) command line utility conceived to spare the user some time in manipulating and converting pdb, crd, dcd, xtc and xyz files. Wordom is also a versatile program for a broad range of analysis of molecular dynamics trajectories. As a plus, it's easy to use Wordom both from the command line and in shell scripts. Due to its simplicity, it is very easy and straightforward to add your own analysis module. Basically, all you have to do is write the algorithm. The data are made available by the existing wordom i/o modules.

1.1 Contacting the authors

Wordom has been developed by Michele Seeber with the crucial help of M. Cecchini, R. Friedman, S. Muff, F. Raimondi, F. Rao and G. Settanni. For bug-alerts, requests or questions about Wordom you can contact him at mseeber@gmail.com. Although Wordom development and maintenance are not his only (nor main) activity he will do his best to answer and/or help. Wordom is in more or less constant development, so bugs may appear and the contribution of users is highly regarded as a polishing tool.

1.2 Citation Reference

If you use Wordom in your work, we would like you to cite the original paper on Wordom:

M. Seeber, M. Cecchini, G. Settanni, F. Rao and A. Caffisch;
Wordom: a program for efficient analysis of molecular dynamics simulations;
Bioinformatics (2007); 23(19), 2625-2627; doi: 10.1093/bioinformatics/btm378

1.3 Acknowledgments

Wordom has been developed with the help of a number of people who contributed with requests, suggestions, bits of code, extensive testing and debugging and so forth. Among others we would like to mention and thank N. Majeux, A. Cavalli, P. Kolb and D. West.

1.4 Copyright and Disclaimer Notices

Wordom is Copyright © 2003-2008 the University of Zurich.

Wordom is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

Wordom is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Wordom. If not, see <http://www.gnu.org/licenses/>

Chapter 2

Installation

Wordom is distributed as binaries for some popular operating systems (OS), such as commonly used linux flavours, and source code which can be compiled on other platforms.

Both binaries and source code can be downloaded at the website:

<http://www.biochem-caflisch.uzh.ch/wordom>

Compilation requires a C compiler, the *make* program and the BLAS/LAPACK libraries (Basic Linear Algebra System/Linear Algebra PACKage). These are light requirements, since free C compilers are widely available and *make* and BLAS/LAPACK are fairly ubiquitous.

Just typing *make* at the prompt in the source directory should generate a *wordom* binary file in the bin directory. You might have to edit the Makefile, where several (commented) lines are provided, which are more appropriate for different systems. Also, some definitions - such as those pertaining the presence of blas/lapack, the kind of compiler and the like, should be checked and, if necessary, corrected/commented out.

2.1 Note for Gromacs Users

To enable the use of XTC files, you have to have Gromacs (GMX) installed. This version of Wordom was tested with GMX 3.3.3 but should be able to work with other (recent) versions. *work is in progress to check with Gromacs 4.0*

Before compilation, uncomment the appropriate lines in the Makefile. Specifically, you should enable Gromacs ($GMX = 1$) and put the relevant values in GMXSRC, GMXHEADER and GMXLIB.

A final note: wordom may be very slow when reading multiple trajectories and in some analysis functions with Gromacs. This seems to be caused due to a bug in libxdrf.c in Gromacs 3.3.3. To prevent this, you can apply the following changes

before compiling wordom:

1. Replace libxdrf.c in `$GMXSRC/src/gmxlib/libxdrf.c` with the version supplied with the file included with wordom, where `$GMXSRC` is the location of your Gromacs source.
2. Recompile and reinstall Gromacs.
3. Uncomment `GMXFIX = 1` in the Makefile.

Wordom performance while running analysis on large xtc files is anyway dreadful, due to the difficulty in randomly accessing frames in xtc files. You might be faster by simply converting your xtc file to a dcd and running your analysis on the latter. Of course, this is especially true if you have multiple analysis to run.

Hopefully, things will be smoother with the next version(s) of Gromacs and Wordom.

Chapter 3

Files and Selections

3.1 File Formats

Wordom can deal with pdb and crd as structure files, *i.e.* containing both a coordinate set and structure information (residues, atoms etc). Informations about these file formats can be found, respectively, at the Protein Data Bank:

<http://www.wwpdb.org/docs.html>

and in the Charmm documentation:

<http://www.charmm.org/document/Charmm/c32b2/io.html#Coordinate>

Wordom can read trajectories in the dcd and xtc formats. Dcd is the native format for the Charmm program. NAMD[®] uses a very similar format that can be read by Wordom. These files only contain coordinate sets (called “frames”), so that, when dealing with them, it is often necessary to also load a structure file.

If you have Gromacs installed, Wordom can also deal with .xtc files:

<http://www.gromacs.org/documentation/reference/online/xtc.html>

However, you should be aware that not all functions are currently working with xtc files, and that wordom does not (yet) deal with some kind of PBC as implemented in Gromacs. The *few* modules that understand and use PBC data require a rectangular box (the easiest to deal with).

In this manual, structure files such as pdb and crd files will be collectively referred to as *mol*, while trajectory files such as dcd and xtc will be collectively referred to as *trj*

3.2 Selections

For analysis and manipulation it is often necessary to select a subset of atoms. Wordom has a selection mechanism which employs a string structured as follows:

`/segname/resnumber/atomtype`

Note: segname is the 12th field in the pdb (3rd after coordinates) and 8th in the crd (1st after coordinates). It is a 4 letter field, not to be confused with the chain (single character) field found after the residue type in the pdb (5th field).

3.2.1 Wild Cards

Wild cards such as * (any number of any character), ? (any single character), [abc] (any single character among a, b and c) and [!abc] (any single character except a, b and c) are supported.

<code>/**/CA</code>	-> all alpha carbons
<code>/MOL1/**</code>	-> all atoms in segment MOL1
<code>/MOL?/**</code>	-> all atoms in MOL1, MOL2, MOLA etc.
<code>/**/C[AB]</code>	-> all alpha and beta carbons

3.2.2 Pattern Matching

Moreover, Wordom selection supports ksh-style pattern matching, such as:

- `?(pattern-list)` → The pattern matches if zero or one occurrences of any of the patterns in the pattern-list allow matching the input string.
- `*(pattern-list)` → The pattern matches if zero or more occurrences of any of the patterns in the pattern-list allow matching the input string.
- `+(pattern-list)` → The pattern matches if one or more occurrences of any of the patterns in the pattern-list allow matching the input string.
- `@(pattern-list)` → The pattern matches if exactly one occurrence of any of the patterns in the pattern-list allows matching the input string.
- `!(pattern-list)` → The pattern matches if the input string cannot be matched with any of the patterns in the pattern-list.

Where pattern-list is a | (pipe) separated list of patterns. A dash can be used to indicate a range of values in the residue number

<code>/**/@(CA C N)</code>	-> backbone atoms
<code>/**/!(CA C N O H OT1 OT2)</code>	-> non-backbone atoms
<code>/MOL1/@(1 3 5)/*</code>	-> residues 1, 3 and 5 of segment MOL1
<code>/MOL1/@(1-5)/*</code>	-> residues 1 to 5 of segment MOL1
<code>/MOL1/@(1-5 10)/*</code>	-> residues 1 to 5 and 10 of segment MOL1

3.2.3 What Else?

At this stage, the selection routine is very simple and reasonably effective, but not particularly flexible or powerful. We are aware of that. Work on it is not over, but, alas, not a very active part of development.

Chapter 4

Coordinates Manipulation

These are the basic Wordom capabilities. The analysis functions themselves rely on this subset of functions to access the data. You can use them to convert your coordinate files in the way better suited to your needs.

In the following examples, valid trj files are dcd and xtc files and valid structure files are pdb and crd files, unless otherwise stated.

The general syntax is organized as follows:

```
-imol    input structure file
-omol    output structure file
-itrj    input trj file
-otrj    output trj file
-sele    selection string or index file
```

4.1 Extraction of a trj frame to a mol file

Options:

```
-f        frame_number
-itrj     trj file
-imol     input structure file
-omol     output structure file
```

Frame number frame_number is read from the trj file and written to output structure file. The input structure file is needed as a reference since trjs do not have any information regarding structure.

Example:

```
wordom -f 35 -itrj trajectory.dcd -imol reference.pdb -omol frame35.pdb
```

4.2 Extraction of a frames series from a trj to a trj

Options:

```
-F      framelistfile
-itrj   inTRJfile
-otrj   outTRJfile
```

Several frames, listed (one frame per line) in a specified file are read from a trj and written to a newly created trj. If the keyword *all* is used, all frames are used. If a reference mol and a selection are specified, only the selected atoms will be part of the new trajectory. In combination with the *all* feature, this is a way to isolate part of your system.

Example:

```
wordom -F framelist -itrj orig_trj.dcd -otrj newtray.dcd
wordom -F all -itrj orig_trj.dcd -imol file.pdb -sele "/A/*/*" -otrj newtray.dcd
```

4.3 Merging of trjs into a single trj

Options:

```
-E      trjlistfile
-otrj   TRJfile
-skip   skipstep
```

The trj files listed in a specified file (one filename per line) are merged into a newly created trj file. Optionally, a skip step can be specified and only one every skipstep frames are considered.

Example:

```
wordom -E trjlist.txt -otrj newtray.dcd
```

4.4 Appending a trj to another trj

Options:

```
-atrj   TRJ1
-otrj   TRJ2
```

TRJ1 is appended to TRJ2, *i.e* all frames of TRJ1 are placed at the bottom of TRJ2, whose frame number is raised accordingly. Both TRJs must have the same number of atoms.

Example:

```
wordom -atrj trajpiece.dcd -otrj wholetraj.dcd
```

4.5 Appending a structure's coordinates to a trj

Options:

```
-amol    MOLfile  
-otrj    TRJfile
```

MOL is appended to TRJ, *i.e* the coordinate set from the structure file (be it a pdb or a crd) is placed at the bottom of TRJ, whose frame number is raised accordingly. Both TRJs must have the same number of atoms. Header values such as timestep, skipstep and the like are taken from the original target trajectory

Example:

```
wordom -amol mymolecule.pdb -otrj traj.dcd
```

4.6 Conversion of a structure to a 1-frame trj

Options:

```
-mono  
-imol    MOLfile  
-otrj    TRJfile
```

The “mono” flag activates the module. PDB or CRD is converted to a TRJ, *i.e* a trajectory file with a single coordinate set (frame) taken from the structure file. The header of the TRJ is correct, but values for timestep and the like are arbitrary.

Example:

```
wordom -mono -imol mymolecule.pdb -otrj smalltraj.dcd
```

4.7 Conversion of file formats

Options:

```
-conv  
-imol    MOL1file  
-omol    MOL2file  
-itrj    TRJ1file  
-otrj    TRJ2file
```

The -conv flag activates the file conversion, which works both between mol and trj formats. Dcd to xtc conversion also requires a reference pdb. The box size and structure are taken from the CRYT1 section of the pdb file, and is not updated.

Example:

```
wordom -conv -imol mymolecule.pdb -omol mymolecule.crd  
wordom -conv -itrj mytrj.dcd -imol mymolecule.pdb -otrj mytrj.xtc  
wordom -conv -itrj mytrj.xtc -otrj mytrj.dcd
```

4.8 Conversion of a trj to a concatenated xyz

Options:

```
-conv  
-itrj    TRJfile  
-omol    XYZfile
```

A trj file is converted to an ASCII file with xyz coordinates of each (selected) atom on a line. Frames are separated by a line reporting the frame number as XYZ framenumbers

Example:

```
wordom -conv -itrj mytray.dcd -oxyz xyzfile.xyz
```

4.9 Showing a trj's headers

Options:

```
-head    TRJfile
```

A TRJ's headers are read and printed out. This gives information about the size of the system, the length of the simulation and some simulation setup. In Wordom-generated TRJ this settings are arbitrary and do not have any meaning. It also gives the "claimed" number of frames versus the "real" number of frames, so that it is possible to know how far the computation has been running if dealing with a TRJ that is being produced by an ongoing simulation.

Example:

```
wordom -head mytray.dcd
```

4.10 Modifying a dcd's headers

Options:

```
-mod      flag=value  
-itrj     DCDfile
```

A DCD's headers can be modified. This might be necessary if some setting's values have not been conserved. This is a peculiar feature, not widely used or (in general) particularly useful. There shouldn't be many occasions where you will need it

Example:

```
wordom -mod timestep=200 -itrj mytray.dcd
```

4.11 Summing several trjs to a single trj

Options:

```
-S        trjlistfile  
-imol     MOLfile  
-otrj     TRJfile  
-sele     selectionstring
```

The trj files listed in a specified file (one filename per line) are summed into a newly created trj. That is, every frame is given by the sum of the differences of each listed TRJ's corresponding frame with respect to the reference structure, added to the reference structure itself. This is used after a PCA run (and a *projection* module run), to sum the projections along different eigenvectors to a single trajectory. The *-sele* argument is NOT optional and, at the moment, MUST be

equal to the selection given in the PCA (and *projection*) run.

Example:

```
wordom -S trjlist.txt -imol reference.pdb -otrj newtray.dcd -sele "/*/*/CA"
```

4.12 Average over a trj

Options:

```
-avg  
-imol    MOLfile  
-itrj    TRJfile  
-omol    averagePDB
```

An average is computed on all the frames in a TRJ file and written to a pdb file

Example:

```
wordom -avg -imol mypdb.pdb -itrj mytrj.dcd -omol average.pdb
```


Chapter 5

Trajectory Analyses

Wordom can run analysis along a trajectory. All these analysis modules need a structure file (pdb or crd) as a reference and a trajectory file (dcd or xtc) to provide the coordinate sets (it is possible to provide a list of trj files). An input file is also required, in which the required module is to be specified along with the appropriate options.

```
wordom -iA analysis.inp -imol file.pdb -itrj file.dcd
```

Or, as an alternative, the command line can be used. Here, the desired module has to be passed as an argument of the `-ia` option, while all parameters for the module itself must be passed just like they would appear in the input file, ie with leading `--` and in uppercase. Fields like selections must also be appropriately shielded from the shell :

```
wordom -ia rmsd --SELE "/*/*/CA" -imol file.pdb -itrj file.dcd
```

The `-E` option (obsolete) can be used instead of the `-itrj` to pass a list of trj files (`-E list.txt`), with one dcd name per line. Or, even simpler, the list filename can be passed directly to the `-itrj` option. Wordom takes files ending with `.txt` as lists and behaves accordingly. The list can thus also list mol (*ie* pdb or crd) files.

```
wordom -iA analysis.inp -imol file.pdb -itrj trjlist.txt
```

Last but not least wordom can run an analysis on a subset of frames. It is possible to specify the first frame to consider (`-beg`), the last one (`-end`), a skip step (`-skip`) or give a list of frames to consider (`-F filename.txt` ; one frame number per line).

The input file begins with the “BEGIN modulename” flag that call the desired module, and ends with the “END” flag.

The “--TITLE title1” flag allows you to give a title (here *title1*): this will be written in the appropriate column of the output time series and/or used to name extra output files (clustering module, pca module).

Output is, unless specified, a time-series of the required parameter computed over the trajectory. Standard output (hence *stdout*) is to the terminal, unless the -otxt outfile.txt option is used (results written to outfile.txt).

Wordom can run more than one analysis at once, so it is possible to write an input files where different modules are called - or the same module is called with different parameters. Keep in mind, however, that if a module works on the results of another analysis (such as PCA projections working with PCA results or 2-pass clustering), you have to run wordom twice with separate input files.

5.1 Distances

This module computes the distance between two atoms or group of atoms. In case more than one atom is selected the geometric center is considered. Since the only option is a double selection, no flag is used. Also, it is possible to compute more than one distance inside the same BEGIN/END group, splitting the different distance selection with a TITLE. This may be faster than using separate BEGIN/END grouping.

Sample Input:

```
BEGIN distance
--TITLE thisdist
--SELE /A/12/CA : /A/26/CA
--TITLE moredist
--SELE /A/13/CA : /A/25/CA
--TITLE sidechaindist
--SELE /A/14/!(CA|N|O|C|HN) : /A/24/!(CA|N|O|C|HN)
END
```

Sample Command Line:

```
wordom -ia distance --SELE "/A/1/N : /A/8/O" -imol file.pdb -itrj file.trj
```

5.2 Contacts

This module checks whether two atoms or group of atoms are within a user-defined cutoff. In case more than one atom is selected the geometric center is considered.

Since the only option is a double selection plus a distance (in Ångstrom), no flag is used. Also, it is possible to compute more than one contacts inside the same BEGIN/END group, splitting the different contact selection with a TITLE.

Sample Input:

```
BEGIN contacts
--TITLE contact1
--SELE /A/12/CA : /A/26/CA : 4
--TITLE contact2
--SELE /A/13/CA : /A/25/CA : 4
--TITLE sidechaincont
--SELE /A/14/!(CA|N|O|C|HN) : /A/24/!(CA|N|O|C|HN) : 4
END
```

5.3 Dihedral angles

This module computes the dihedral angle between four selected atoms. The atoms are selected with four separated selection, each of which must select one and only one atom. Again, no flag is used to give the selections.

Sample Input:

```
BEGIN dihedral
--TITLE dihe1
--SELE /A/12/CA : /A/12/C : /A/13/N : /A/13/CA
END
```

5.4 Hydrogen Bond Detection

The user selects three atoms: an oxygen atom, an hydrogen atom, and an heavy atom to which the hydrogen is bound. If the distance between the O and the H is less than 3.6 Å and the angle formed by the three atoms (O-H-X) is more than 130° an hydrogen bond is accepted as present and the module outputs a “1” (as opposed to a “0” when the bond is not present). It is possible to compute more than one contacts inside the same BEGIN/END group, splitting the different contact selection with a TITLE.

Sample Input:

```

BEGIN hbond
--TITLE hb1
--SELE /A/12/O : /A/26/H : /A/26/N
END

```

5.5 Radius of giration

The radius of gyration is defined as:

$$rgyr = \sqrt{\frac{\sum_i \left((x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2 \right)}{N}} \quad (5.1)$$

No mass informations are used by default in this calculation, so it should be considered a “geometrical” RoG. The mass-weighted RoG can be computed by adding the --MASS flag and providing a crd/pdb with the masses written in the wmain/beta factor field.

Sample Input:

```

BEGIN rgyr
--TITLE rgyrCA
--SELE /A/*/CA
END

```

5.6 RMSD

The Root Mean Square Deviation is defined as:

$$RMSD = \sqrt{\frac{\sum_i \left((x_i - x_{ref})^2 + (y_i - y_{ref})^2 + (z_i - z_{ref})^2 \right)}{N}} \quad (5.2)$$

The *reference* structure is, unless the --PROGRESSIVE flag is used (see below) the structure given from the command line with the -imol or -imol options.

Options:

Flag	Argument	Input
--SELE	sele_string	atoms selection
--PROGRESSIVE	none	if specified, wordom compute the rmsd of each frame with respect to the previous frame. This allows to better visualize conformational changes <i>along</i> the dynamic rather than with respect to the reference structure
--NOSUPER	none	if specified, no superposition is carried out before RMSD computing
--DCDOUT	filename.dcd	if specified, a dcd is written with the aligned frames - do not use together with --NOSUPER

Sample Input:

```
BEGIN rmsd
--TITLE rmsd1
--SELE /A/*/CA
--DCDOUT alignedtrj.dcd
END
```

5.7 DRMS

The Distance Root Mean Square (Deviation) is the RMSD of the internal distances matrix of each frame with the one computed on the reference structure.

Sample Input:

```
BEGIN drms
--TITLE drms1
--SELE /A/*/CA
END
```

5.8 RMSD- and DRMS-based Clustering

Clustering of the structures of a trajectory can be accomplished using different methods (algorithms) and different criteria to judge structure similarity.

Options:

Flag	Argument	Input
--SELE	sele_string	atoms selection
--METHOD	hiero <i>or</i> qt <i>or</i> leader	the algorithm to be used
--DISTANCE	rmsd <i>or</i> drms	the “distance” used to compute the similarity between any two structures
--NOSUPER	none	if specified, no superposition is carried out before RMSD computing
--STEP	int	the skip step to use while reading the trajectory file. Do <u>not</u> use the command line -skip option.
--CUTOFF	float	the cutoff to be used in clustering

Hierarchical (hiero) and quality threshold (qt) yield similar results and are very robust with respect to frames order but are quite performance-hungry. Expect to use $\text{nframe} \times \text{nframe} \times \text{sizeof(float)}$ bytes for a hiero or qt run. That means that 2 GB and a $\text{sizeof(float)}=4$ will allow you to cluster no more than 23000 frames. A big system (lots of atoms) will restrict you even further since those have to be stored for the computation of the distance even before the real clustering begins. For a description of the theory see:

http://en.wikipedia.org/wiki/Data_clustering

Leader-like is much faster and uses much less memory but is frame-order dependent.

The output is different depending on the chosen algorithm. Both hiero and qt leave the *stdout* empty, *ie* with only a list of the frames, and generate an output file (named as the job --TITLE) where the the cluster are listed. For each cluster a progressive number, its population, the center (a representative frame) and a list of the belonging frames are given. Cluster #0 is not a real cluster: it is actually made up by all isolated frames.

Leader-like clustering, on the other hand, puts its results in the *stdout*. All frames are listed, but only those used in clustering (every --STEP steps) have additional data: an int indicating the cluster to which the frame belongs and a float which reports the DISTANCE (rmsd or drms) from the cluster center (which has 0.000 as distance and is the first frame belonging to the cluster).

Sample Input:

```
BEGIN cluster
--TITLE c1
--SELE /A/*/CA
```

```
--DISTANCE rmsd
--METHOD hiero
--CUTOFF 5
--STEP 10
END
```

5.8.1 2-pass Clustering

It is possible to run a 2-pass clustering using the hierarchical or the quality threshold algorithm: a subset of frames is clustered and, in a second pass, all the frames are assigned to the clusters found in the first step. The second pass needs to read in the results of the first with the `--FILE` option. Keep in mind that the CUTOFF has a slightly different meaning in the 2nd pass. While in the first pass every conformation belonging to a cluster must be below the cutoff with respect to any other structure in the cluster, in the 2nd pass a structure needs only to be below the CUTOFF with respect to the center of the cluster identified in the 1st pass. Thus, we might (roughly) say that the CUTOFF is the *diameter* of the cluster in the 1st pass, and the *radius* in the 2nd.

Sample Input:

```
BEGIN cassign
--TITLE c-2pass
--FILE c1.out
--SELE /A/*/CA
--DISTANCE rmsd
--CUTOFF 5
--STEP 5
END
```

5.9 Orientational Parameters P1+P2

This module computes the polar (P1) and nematic (P2) order parameters[1] between selected segments along a trajectory. The segments are selected specifying the first and last atom of the fragment. No flags are employed.

Sample Input:

```
BEGIN orienta
--TITLE ps
--SELE /A/1/C : /A/10/N
```

```
--SELE /B/1/C : /B/10/N
--SELE /C/1/C : /C/10/N
END
```

5.10 Principal Component Analysis

PCA is computed on a selected set of atoms. Common procedure is to first superimpose the whole trajectory (with the *RMSD* module, --DCDOUT option) to a reference structure, then compute the average with the -avg command line option (see above, in *Coordinate Manipulation*) and then run the PCA analysis on the superimposed trajectory with the average structure as reference structure.

Options:

Flag	Argument	Input
--SELE	sele_string	atoms selection
--PROGRESSIVE	none	activates the PROGRESSIVE procedure
--NPRINT	int	how many eigenvectors are checked in PROGRESSIVE
--VERBOSE	none	intermediate eigenvectors are written

Sample Input:

```
BEGIN PCA
--TITLE pca1s
--SELE /A/*/CA
--NPRINT 5
--PROGRESSIVE 1000
--VERBOSE
END
```

The module does not output to *stdout* - only a list of frames is to be found there. Wordom-PCA writes 3 files with the PCA results: title-eigvec.txt with the eigenvectors in columns, title-eigval.txt with the eigenvalues and title-matrix.txt with the covariance matrix

5.10.1 Principal Component Analysis Projection

The projection of each frame along a selected eigenvector is computed. This is what is often found in literature graphs where PCA1 vs PCA2 projections (pro-

jections along the first and second eigenvectors) are plotted

Options:

Flag	Argument	Input
--SELE	sele_string : sele_string	atoms selection
--FILE	filename.txt	eigvec file from previous PCA analysis run
--VECTOR	int	the eigenvector to use for the projection
--DCDOUT	filename.dcd	if specified, a dcd with the projection is written
--RANGEFILL	float	if specified... see description below

The optional `--DCDOUT` flag generates a trajectory with only the motion along the eigenvector.

The `--RANGEFILL` flag is for representation purposes: a spurious trajectory file is created to illustrate the motion spotted by the eigenvector. The two extremes in the projection are picked and the range evenly divided with a `RANGEFILL` spacing. A frame is generated for each interval and written to a `TITLE.rangefill.dcd` file. Visualization of this trajectory show the progress from an extreme of the motion to the other. Visualization of all the frames together makes for a cool picture, especially if coloured according to the λ factor described in the PCA section, or to the same factor weighted according to the frame and the range of this fake trajectory. A tool written in python to handle the trj and accomplish this representation(s) is (will soon be) available in the tools section.

Sample Input:

```
BEGIN project
--TITLE proj1
--FILE pca1s-eigvec.txt
--VECTOR 1
--SELE /A/*/CA : /A/*/CA
--DCDOUT outfile.dcd
--RANGEFILL 0.25
END
```

5.11 Q Entropy

The calculation of the quasiharmonic entropy is based on the diagonalization of the mass-weighted covariance matrix of the atomic fluctuations. A processed structure

file is to be supplied, with the mass of each atom in the β -factor field. Coordinates superposition to the given reference structure (which ought to be an average over the trajectory) with rmsd minimization is automatically carried out. The default temperature at which entropy is computed is 300.

The vibrational entropy, quasiharmonical vibrational energy and vibrational specific heat are listed at the top of the eigenvalues file.

For details see the relevant paper by Andricioaei and Karplus[2].

Sample Input:

```
BEGIN entropy
--TITLE qentr1
--SELE /A/*/CA
--TEMP 330
END
```

5.12 Secondary Structure

This module computes the secondary structure of the conformations listed in a trj. Different algorithms can be employed. The --DSSP option uses a DSSP-like algorithm which fairly reproduces the results of the DSSP program[3]. The --DSSPCONT option mimics the results of the DSSPcont program[4, 5]. Since the algorithms were re-written from scratch with just the guideline of the relevant papers and some knowledge about secondary structure the results are bound to be different, though quite comparable.

Sample Input:

```
BEGIN sstruct
--TITLE ss1
--DSSPCONT
END
```

Sample Command Line:

```
wordom -ia sstruct --TITLE ss1 --DSSPCONT 1 -imol file.pdb -itrj file.dcd
```

Chapter 6

More Analyses

Although Wordom was conceived to run analysis on trajectory files, i.e., on series of coordinates sets, there are also modules that operate on single structures, small sets of structures or even different kinds of data (usually extracted from trajectories, though).

6.1 Kinetic Grouping Analysis (KGA)

KGA can be used for the identification of free-energy basins, not according to geometrical characteristics (such as the fraction of native contacts or RMSD from the folded structure) but rather according to fast relaxation at equilibrium. More explicitly, two coarse-grained conformations are grouped if along the MD trajectory their snapshots interconvert in more than 50% of the cases within a short commitment time τ_{commit} , which represents a typical relaxation time within basins of the investigated system [6]. The idea behind this approach is that if two conformations interconvert rapidly, they are not separated by a barrier and therefore belong to the same basin.

6.1.1 A bit of Theory - kinetic grouping

The commitment time τ_{commit} . The “typical relaxation time within basins” mentioned above, τ_{commit} , is a characteristic of the investigated system. It is an important parameter of KGA and defines the lense of resolution with which basins are isolated. A short τ_{commit} will group structures only locally or if the free-energy surface is very smooth. A longer τ_{commit} is more generous and might group sub-basins isolated by a short τ_{commit} into larger basins. The first passage time to the native node (or a representative node of another basin), plotted as a free energy on a logarithmic x-axis, usually reflects two timescales: the inter- and intrabasin

relaxation times. The barrier that separates the two regimes can give a good indication for the (upper bound) relaxation time.

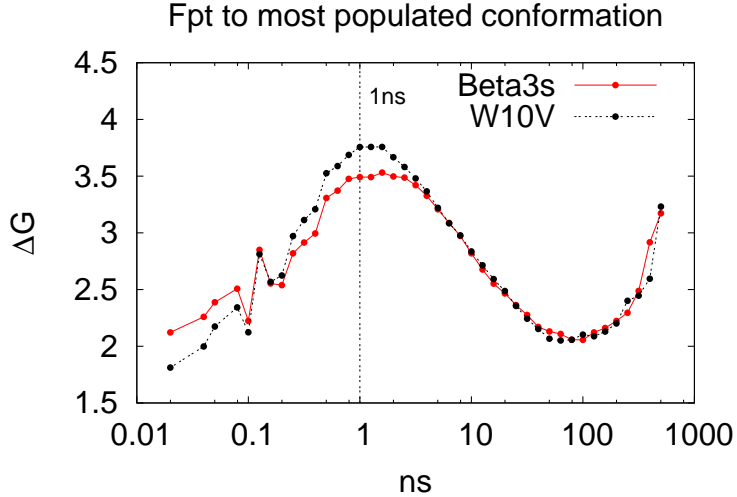


Figure 6.1: Distribution of first passage times $P(fpt)$ to the most populated node. Values are calculated as $\Delta G = -k_B T \ln(P(fpt))$ and given in kcal/mol. This projection is very useful to determine the transition between intra-basin and inter-basin relaxation, which is emphasized by the logarithmic binning (10 bins/decade) without normalization of the binsize. The peak at about 0.1 ns is an artifact of the binning but does not have any effect on the results.

Isolation of all relevant basins at once. For a fixed commitment time τ_{commit} a matrix with interconversion (commitment) probabilities p_{commit} between any pair of nodes can be calculated in principle, and nodes with $p_{commit} \geq 0.5$ are grouped together. Since the computational cost of all-against-all calculations increases quadratically, in practice one selects a subset of highly populated nodes (e.g., the 500 most populated nodes), calculates the p_{commit} -matrix and divides them into basins. In a postprocessing step, all other nodes are assigned commitment probabilities to the isolated basins and grouped to a basin if $p_{commit} \geq 0.5$ for one of the basins. Otherwise, these nodes remain unassigned. Both the heavy-node calculation and the postprocessing is done by wordom in the same function.

Isolation of a single basin. If only one basin is of interest or the basins have different relaxation times and one wants to isolate them one-by-one, p_{commit} is

calculated only with respect to a given node (typically the representative/most populated node of a basin) and all nodes with $p_{commit} \geq 0.5$ are grouped into the investigated basin.

6.1.2 Modules usage: kinetic grouping

logbin module: First passage time - plot to find τ_{commit}

Reads in the timeseries of noderanks (i.e., a one-column file, each row indicating the rank of the population of the node, e.g., most populated node=1, second most populated node=2, etc.) and gives back the x- and y-coordinates of the logarithmically binned free-energy fpt-plot with respect to a selected node.

Option	Argument	Input
-ia	logbin	option to call the logbin module
--CLUSFILE	string	timeseries of noderanks
--BPD	int	bins per decade
--TARGET	int	nodename with respect to which the first passage time should be calculated

Example:

```
wordom -ia logbin --CLUSFILE noderank.tt --BPD 10 --TARGET 1
```

ka module: Kinetic grouping analysis to isolate all basins at once

Reads in the timeseries of noderanks and groups nodes into basins according to KGA. The procedure calculates the all-against-all matrix for a selected number of most populated nodes and assigns all other nodes in a postprocessing step.

Option	Argument	Input
-ia	ka	option to call the KGA module
--CLUSFILE	string	timeseries of noderanks
--TCOMM	int	commitment time τ_{commit} (number of frames)
--NNODES	int	number of nodes for all-against-all

Example:

```
wordom -ia ka --CLUSFILE noderank.tt --TCOMM 50 --NNODES 500
```

basin module: Kinetic grouping analysis to isolate a single basin

Reads in the timeseries of noderanks. The output is the list of commitment probabilities (p_{commit}) of all nodes to the target node. The last part of the output is a list of all nodes in the basin of the selected targetnode.

Option	Argument	Input
-ia	basin	option to call the basin isolation module
--CLUSFILE	string	timeseries of noderanks
--TCOMM	int	commitment time τ_{commit} (number of frames)
--TARGET	int	nodename (rank) of the target node

Example:

```
wordom -ia basin --CLUSFILE noderank.tt --TCOMM 50 --TARGET 1
```

6.2 Cut-based free-energy profiles (FEPs)

A progress coordinate that preserves the barriers and minima in the order that they are met during folding/unfolding events was introduced by Krivov and Karplus [7]. It uses the relative partition function as the progress coordinate and determines the free energy barriers as a function of the coordinate by a method based on p_{fold} . The procedure gives almost identical results if p_{fold} is replaced by the mean first passage time (mfpt) to a selected node [8].

6.2.1 Pfoldf (Pfold fast)

Given the transition network with symmetrized links (equilibrium kinetic network or EKN) and two nodes A and B, corresponding to the “folded” and “denatured” node, the p_{fold} of node i is the solution of the equation $p_i = \sum_j p_{ji} \cdot p_j$ with boundary conditions $p_A = 1$ and $p_B = 0$. In a 2-state system with two enthalpic basins, one corresponding to the folded and one to the unfolded state, the two nodes A and B are the representative (most populated) nodes of the system.

However, in many systems, a node such as B does not exist, because there are multiple basins and/or an entropic state that cannot be represented by a single node. Thus, as in the balanced minimum-cut procedure [9], an extra node B is introduced and connected to all nodes in the network with capacity $\lambda \tilde{w}$, where λ is a Lagrange multiplier (usually < 0.01). The p_{fold} calculations are performed on the

EKN with the extra node and the nodes are sorted according to their p_{fold} . Each value p_c between 0 and 1 can then be used to cut the network into set A containing all nodes with $p_{fold} > p_c$ and set B containing the nodes with $p_{fold} < p_c$. For each cut a point ($x = Z_A/Z, y = -kT \ln(Z_{AB}/Z)$) of the FEP is obtained; Z_A/Z is used as the progress coordinate and Z_{AB} is the number of EKN-transitions between the two sets. The minimal cut value Z_{AB} between two sets split by the p_{fold} variable is a good approximation of the minimal cut between A and B [7, 8], implying that the maximal value of $-kT \ln(Z_{AB}/Z)$ is a good approximation of the barrier.

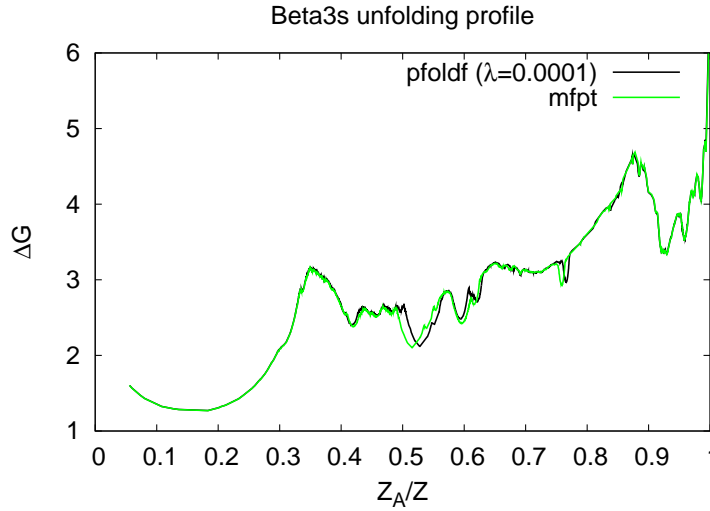


Figure 6.2: Unfolding free-energy profiles from the native basin of Beta3s [8]. The results of the pfoldf and mfpt procedures are essentially identical, indicating the robustness of the procedure on the underlying variable.

6.2.2 Mfpt (Mean first passage time)

In the pfoldf procedure described above the input are two nodes. Pfoldf is therefore appropriate to find barriers between two well-defined basins. However, sometimes it is useful to plot unfolding profiles with respect to only one node, especially if no representative node in the denatured state exists. Pfoldf solves the problem by introducing the extra node, but it is simpler to change the progress variable and use mean first passage time (mfpt) to the node of interest, because mfpt is defined only with respect to one node.

Given the EKN, the mfpt of node i is the solution of the equation $mfpt_i = \Delta t + \sum_j p_{ji} \cdot mfpt_j$ with boundary condition $mfpt_A = 0$ [10]. The timestep Δt

corresponds to the saving frequency of 20 ps, i.e., the mfpt of a node is defined as one timestep plus the weighted average of the mfpt values of its adjacent nodes. Mfpt has explicit time dependence through the occurrence of Δt in the equations. The resulting large system of linear equations differs from the one of pfoldf only by the Δt constant and the boundary conditions. Therefore, both can be solved with the same efficiency by iterative multiplication. Mfpt does not require an extra node, because mfpt is not defined between a pair of nodes, but only with respect to one node. To calculate the FEP, the nodes are sorted according to their mfpt value. For all node-values $mfpt_c$ a point $(x = Z_A/Z, y = -kT \ln(Z_{AB}/Z))$ on the FEP can be calculated, where A is the set of all nodes with $mfpt_i < mfpt_c$ and B the set of nodes with $mfpt_i > mfpt_c$. The differences between unfolding feps of the Beta3s peptide for pfoldf with $\lambda=0.0001$ and mfpt are marginal (see Figure above).

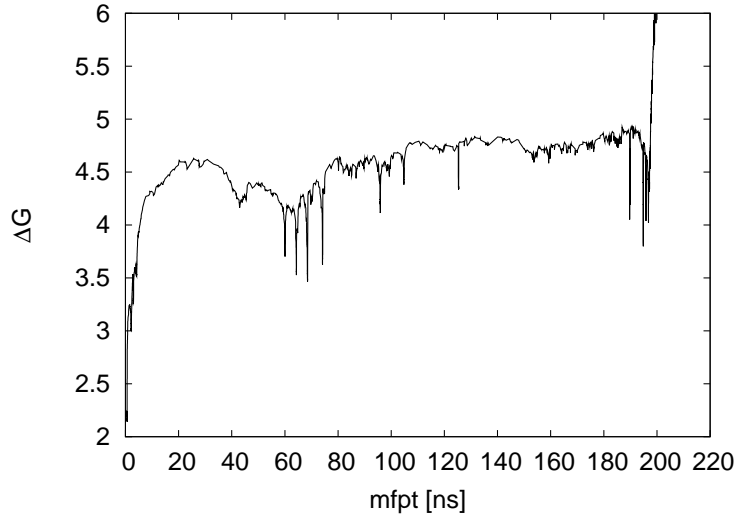


Figure 6.3: The Z_A/Z coordinate is substituted by the mfpt values of the nodes.

6.2.3 Modules usage: Cut-based free-energy profiles (FEPs)

Pfoldf

Option	Argument	Input	Default
-ia	pfoldf	option to call the module	
--CLUSFILE	string	timeseries of noderanks	
--TEMP	float	temperature of the system	300K
--LAMBDA	float	Lagrange multiplier	0.0001 (if target2=0) 0 (else)
--TARGET	string	start node (pfold=1)	
--TARGET2	string	stop node (node with pfold=0)	0 (= extra node)
--NIT	int	# of iterations to solve the equations	50000
--NONSymm		prevents symmetrization of network	no argument

Note that the output file contains nodes sorted according to their weight (number of snapshots). Therefore, to plot the profile it is possible to stop the calculation after a desired number of output pairs and then sort according to column 1 (e.g., | head -2000 | sort -nk1). The columns in the output file are \$1= Z_A/Z , \$2= ΔG , \$3= p_{fold} , \$4=rank of node

By default a symmetrized (detailed balance is imposed) network is used: this can be prevented by specifying the --NONSymm option.

ATTENTION: If pieces of trajectories from different simulations are concatenated, insert a line with the entry "0" in between to prevent spurious transitions, i.e., to avoid them to be treated as a continuous timeseries. "0" must not be used otherwise.

Example:

```
wordom -ia pfoldf --CLUSFILE noderank.tt --TARGET 1 --TARGET2 0
--LAMBDA 0.0001 --NIT 100000
```

Pfoldfnet

Instead of reading the timeseries of the noderanks, it is also possible to calculate the profile from the linkfile (i.e., the network). Note that this is more efficient than using the option `--CLUSFILE`.

Option	Argument	Input	Default
<code>-ia</code>	<code>pfoldfnet</code>	option to call the module	
<code>--LINKFILE</code>	string	links and weights (three-column file)	
<code>--TEMP</code>	float	temperature of the system	300K
<code>--LAMBDA</code>	float	Lagrange multiplier	0.0001 (if target2=0) 0 (else)
<code>--TARGET</code>	string	start node (pfold=1)	
<code>--TARGET2</code>	string	stop node (node with pfold=0)	0 (= extra node)
<code>--NIT</code>	int	# of iterations to solve the equations	50000
<code>--NONSymm</code>		prevents symmetrization of network	no argument

The output format is identical to the one from the `pfoldf` function.

Example:

```
wordom -ia pfoldfnet --LINKFILE linkfile.txt --TARGET 1 --TARGET2 0
--LAMBDA 0.0001
```

Mfpt

Option	Argument	Input	Default
<code>-ia</code>	<code>mfpt</code>	option to call the module	
<code>--CLUSFILE</code>	string	timeseries of noderanks	
<code>--TEMP</code>	float	temperature of the system	300K
<code>--TARGET</code>	string	start node (pfold=1)	
<code>--NIT</code>	int	# of iterations to solve the equations	50000
<code>--NONSymm</code>		prevents symmetrization of network	no argument

As for `pfoldf`, the output file contains nodes sorted according to their weight (number of snapshots). Therefore, to plot the profile it is possible to stop the calculation after a desired number of output pairs and then sort according to column 1 (e.g., `| head -2000 | sort -nk1`). To use `mfpt` as reaction coordinate instead of Z_A/Z : the columns in the output file are \$1= Z_A/Z , \$2= ΔG , \$3=`mfpt`, \$4=`rank of node`. Therefore, (x=\$1,y=\$2) is the usual ΔG vs. Z_A/Z plot, while (x=\$3, y=\$2) is the ΔG vs. `mfpt` plot, where the separation from the target basin is measured by

a distance in time units.

By default a symmetrized (detailed balance is imposed) network is used: this can be prevented by specifying the `--NONSymm` option.

ATTENTION: If pieces of trajectories from different simulations are concatenated, insert a line with the entry "0" in between to prevent spurious transitions, i.e., to avoid them to be treated as a continuous timeseries. "0" must not be used otherwise.

Example:

```
wordom -ia mfpt --CLUSFILE noderank.tt --TARGET 1 --NIT 100000
--NONSymm --TEMP 330
```

Mfptnet

In analogy to the `pfoldfnet` function, also `mfpt` profiles can be calculated by giving the linkfile as input. This is more efficient than using the `clusfile`.

Option	Argument	Input	Default
<code>-ia</code>	<code>mfptnet</code>	option to call the module	
<code>--LINKFILE</code>	string	inks and weights (three-column file)	
<code>--TEMP</code>	float	temperature of the system	300K
<code>--TARGET</code>	string	start node (pfold=1)	
<code>--NIT</code>	int	# of iterations to solve the equations	50000
<code>--NONSymm</code>		prevents symmetrization of network	no argument

Example:

```
wordom -ia mfptnet --LINKFILE linkfile.txt --TARGET 1 --TEMP 330
```

6.2.4 A posteriori equilibration of out of equilibrium simulations

(Rao & Krivov, work in preparation) For many systems long equilibrium simulations are unfeasible. Therefore, the focus is often put on the sampling of transitions following only one direction, starting in regions of relatively high energy and terminating once the system reaches the low-energy region of interest (e.g., the folded state). The result is an ensemble of out of equilibrium (kinetic) simulations, which do not represent a thermodynamically correct picture of the free-energy surface.

The low-energy state will remain underrepresented, because the simulation time-scale is much shorter than the expected unfolding time.

Despite the incorrect overall thermodynamic picture, it can be assumed that locally the transition probabilities ($p_{ij} = n_{ij}/n_j$ to go from node j to i within one time step, given that the trajectory is currently in j) are correct. By solving the system of equations, corresponding to the correct population probabilities p_i^{eq} of the nodes, new weights are assigned to all nodes and links. The system of equations

$$\begin{aligned} p_i^{eq} &= \sum_j p_{ij} \cdot p_j^{eq} \\ \sum_i p_i^{eq} &= 1 \end{aligned}$$

is used to “equilibrate” the weights of the nodes (where $n_j^{eq} = p_j^{eq} N$, with N being the total number of snapshots in the trajectory). The weights of the links are then “equilibrated” by

$$n_{ij}^{eq} = p_{ij} \cdot n_j^{eq}.$$

Option	Argument	Input	Default
-ia	equil	option to call the module	
--LINKFILE	string	inks and weights (three-column file)	
--NIT	int	# of iterations to solve the equations	100'000

Example:

```
wordom -ia equil --LINKFILE linkfile.txt --NIT 80000
```

6.3 Elastic Network Modes (ENM)

Module is working, but the manual is work in progress

6.4 Protein Structure Network Graph (PSG)

Module is \pm working, but the manual is work in progress

Bibliography

- [1] M. Cecchini, F. Rao, M. Seeber, and A. Caflisch, *J. Chem. Phys.*, **2004**, *121*.
- [2] I. Andricioaei and M. Karplus, *J. Chem. Phys.*, **2001**, *115*, 6289–6292.
- [3] W. Kabsch and C. Sander, *Biopolymers*, **1983**, *22*(12), 2577–637.
- [4] C. A. F. Andersen, A. G. Palmer, S. Brunak, and B. Rost, *Structure*, **2002**, *10*, 174–184.
- [5] P. Carter, C.A.F. Andersen, and B. Rost, *Nucleic Acids Research*, **2003**, *31*(13), 3293.
- [6] S. Muff and A. Caflisch, *Proteins: Structure, Function, and Bioinformatics*, **2008**, *70*, 1185–1195.
- [7] S. V. Krivov and M. Karplus, *J. Phys. Chem. B*, **2006**, *110*, 12689–12698.
- [8] S. V. Krivov, S. Muff, A. Caflisch, and M. Karplus, *J. Phys. Chem. B*, **2008**, *112*, 8701–8714.
- [9] S. V. Krivov and M. Karplus, *J. Chem. Phys.*, **2002**, *117*, 10894–10903.
- [10] M. Apaydin, D. Brutlag, C. Guesttin, D. Hsu, and J. Latombe, "In International Conference on Computational Molecular Biology (RECOMB)", **2002**.