TimeScapes: transforming timeseries into spatial images

Julio Kovacs, ODU

General goal: discovering connections between local and global processes

- Fast, local processes: $X_i(t)$
- Slow, global process: $a(t)$ ("activity function")
- Ranking of the i^{th} process:

$$R_{X,a}(i) = I\left(\left|\frac{dX_i(t)}{dt}\right|, a(t)\right)$$

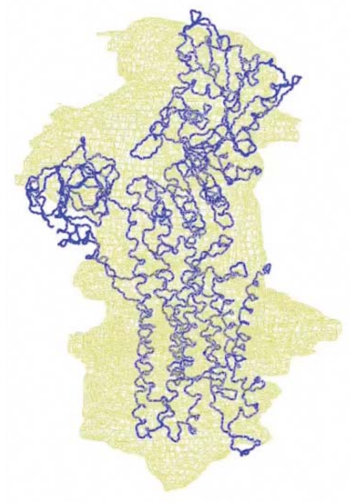
This ranking provides a measure of the relevance or contribution of each of the local processes $X_i(t)$ to the global process $a(t)$.

- I is a statistical measure of dependence, such as:
  - Pearson correlation coefficient (current release).
  - Mutual Information (in the upcoming release).

The approach is of general applicability

- The index *i* could correspond to any type of spatial attribute.
- It could also denote connections between pairs, triplets, … of locations in space.
- Our applications will deal with single and pairwise locations.

Flexible fitting: DDFF and MDFF


Fitting of Ca-ATPase by
“Damped Dynamics Flexible Fitting”
(Kovacs et al., *Biophys. J.*, 2008)

MDFF is the best known and *de facto* approach for flexible fitting at atomic detail, but it requires a full MD trajectory.

As such, MDFF can benefit from the TimeScapes analysis presented here.

Application to Molecular Dynamics trajectories

- *TimeScapes* can perform two fundamental types of analyses on MD trajectories:
  1. Hinge-bending of protein molecules (“pivot analysis”).
  2. Pairwise residue distance geometry (“contact analysis”).
- The results of these analyses (i.e., the $R_{X,a}(i)$ as a function of i) can be mapped onto the 3D molecular structure for visualization.
- The “heat maps” thus obtained provide a picture of the regions of the molecule that are relevant for the significant events encoded in the long trajectory.

First step: calculation of the activity functions

- Three types of activity functions can be computed in *TimeScapes*:
  1. RMS fluctuations in a sliding time-window.
  2. Rate of contact-forming and -breaking events. This uses a coarse-grain model of 1 atom per side chain, and can be done by means of either of 2 types of graph:
     a. A simple distance-cutoff graph;
     b. A “Generalized Masked Delaunay” graph.

     This approach needs a preliminary smoothing of the distance time-series in order to suppress high-frequency noise.

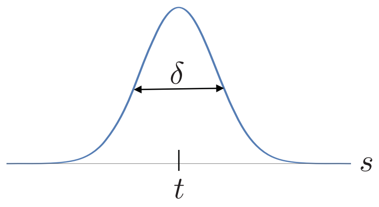
RMS fluctuation activity

- Computed as “standard deviation” of the whole ensemble of atomic Cartesian coordinates over a time-window centered at successive time frames. The values being averaged are weighted with a Gaussian function:

$$\sigma_i^2(t) = \sum_{s=t-\delta}^{t+\delta} \|p_i(s) - \bar{p_i}(t)\|^2 \cdot G(s-t) \qquad \bar{p_i}(t) = \sum_{s=t-\delta}^{t+\delta} p_i(s) \cdot G(s-t)$$

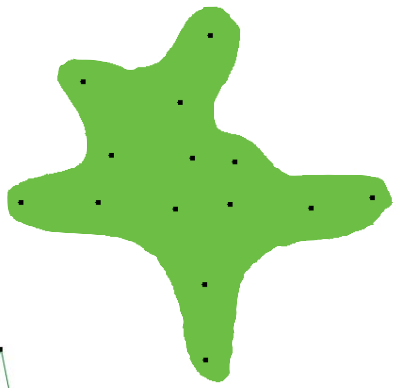
$$\mathrm{RMS}(t) = \sqrt{\frac{1}{N}\sum_i \sigma_i^2(t)}$$

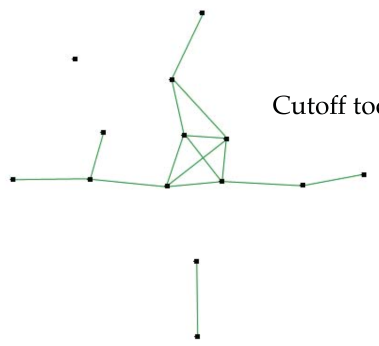
$G(s-t)$ = Gaussian whose FWHM is δ

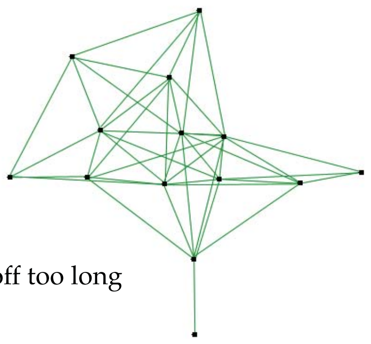


Simple distance-cutoff graph

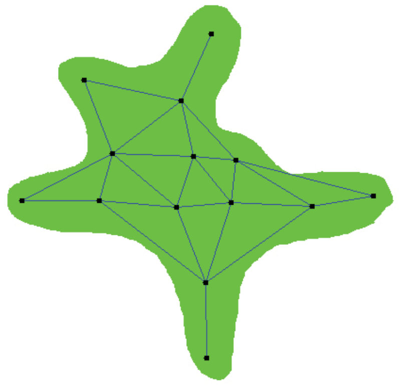
- Contacts are defined by pairs of representative atoms that are closer than a prescribed cutoff distance.
- Careful choice of the cutoff distance is important.

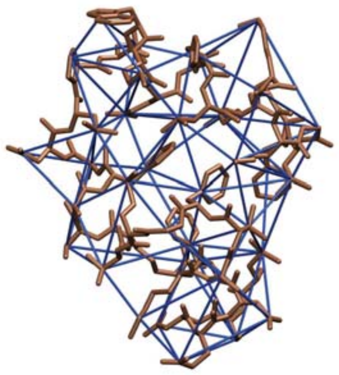



Cutoff too short


Cutoff too long

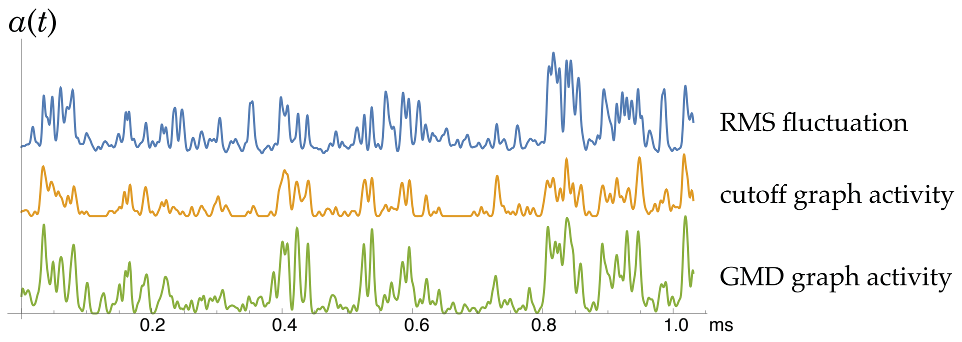
Generalized Masked Delaunay graphs (GMD)


Masked Delaunay graph: masking out edges of the Delaunay graph not contained in the protein


3D example: Villin

The *Generalized* Masked Delaunay graphs are higher-order versions of the above, and are used to define a “contact metric” for a “recrossing filter”, which suppresses spurious events due to noise and sampling granularity. (Metric = smallest-order graph that contains a given edge.)

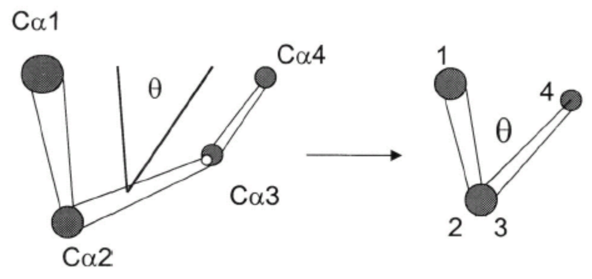
Example: activity measures for BPTI



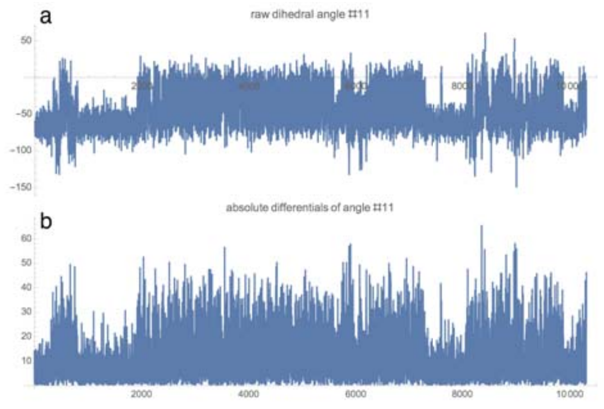
These three activity functions look very similar to one another.
(But this isn’t always the case; see Wriggers et al., *J. Chem. Theory Comput.*, 2009, 5:2595-2605.)

BPTI trajectory is from the *Anton* millisecond simulation from Shaw et al., 2010.

Pivot analysis: detection of hinging hot spots



Pseudo-dihedral angle θ generated by 4 consecutive Cα atoms



a. Example of a particular pseudo-dihedral angle time series from the MD trajectory of BPTI.

b. The absolute values of its time differentials.

Pivot analysis: detection of hinging hot spots

For this, we apply our basic equation:

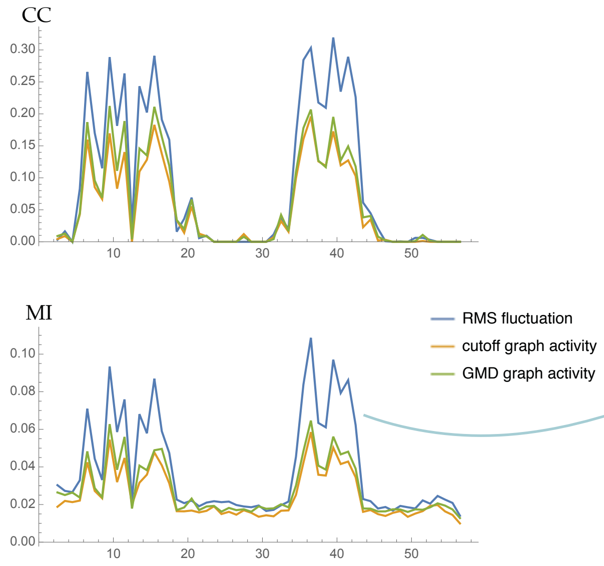
$$R_{X,a}(i) = I\left(\left|\frac{dX_i(t)}{dt}\right|, a(t)\right)$$

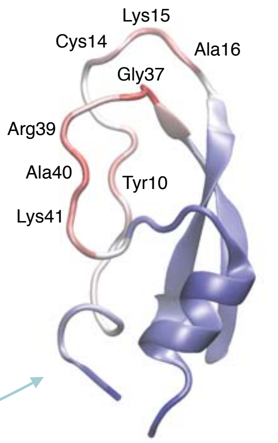
to the variables X_i = pseudo-dihedral angle # i,
with I = either CC or MI,
and $a(t)$ = each of the 3 activity functions:

- RMS fluctuations
- cutoff graph
- GMD graph

Pivot analysis for BPTI

In this example, the 3 curves are very consistent:



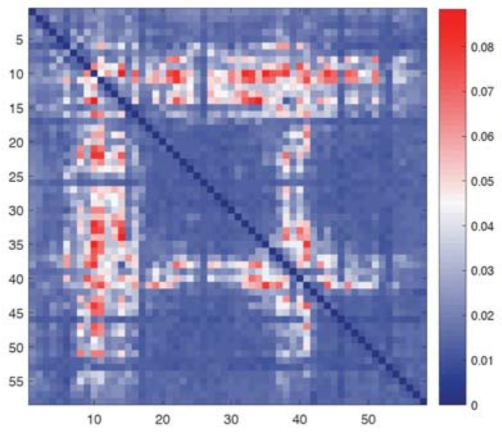


Heat map of pivot angle MI for RMS fluctuation

Contact analysis

Here, the index i ranges over *pairs* of residues, and the variable $X_i(t)$ is the distance between the corresponding pair of residues.

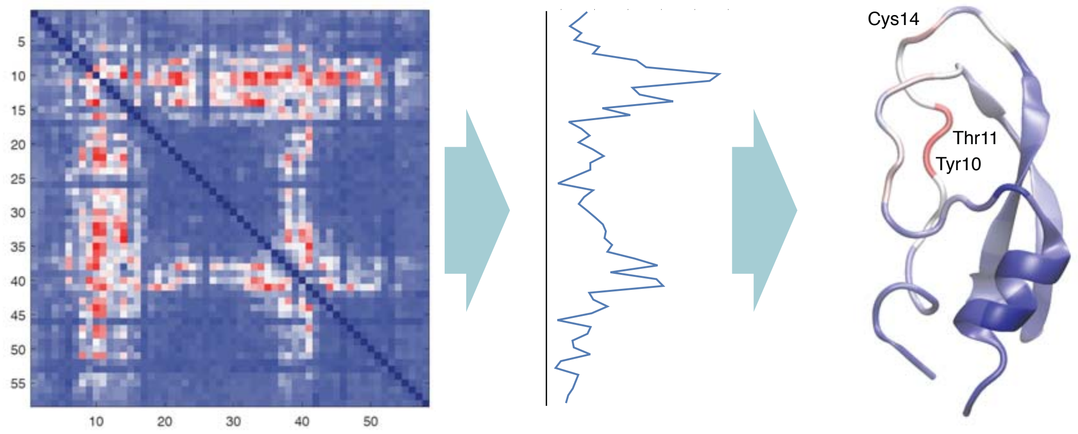
Then the ranking function $R_{X,\alpha}(i)$ can be displayed as a matrix:



MI contact matrix for BPTI, with $\alpha(t)$ = RMS fluctuation

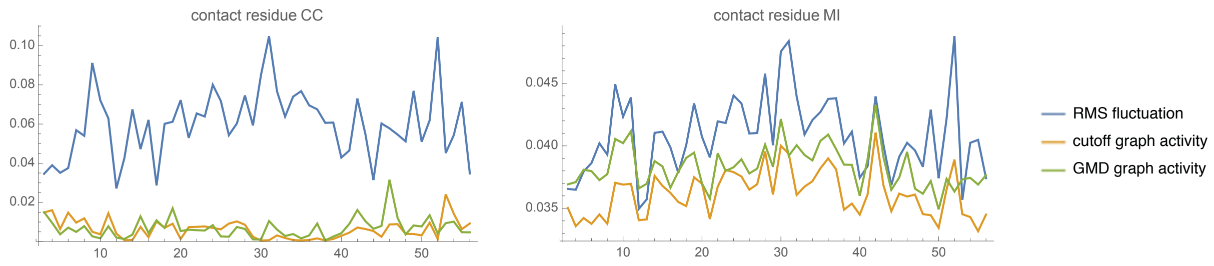
Contact analysis

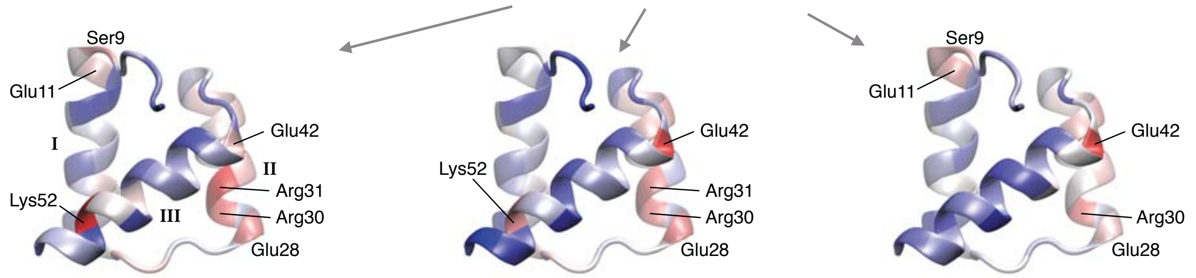
This contact matrix is symmetric and has a banded structure. It can be projected onto either axis, and then mapped onto the 3D chain:



Outlook to Timescapes 1.4 (Summer 2016): Advantages of Nonlinear MI vs. Linear CC

We looked at a heat-induced unfolding trajectory of EnHD protein (Daggett lab, UW). CC profiles for graph-based activities were weak, while the MI results were more consistent with analysis based on RMS fluctuation activity:





contact residue MI based on RMS fluctuation

contact residue MI based on cutoff graph

contact residue MI based on GMD graph

Overview of the *TimeScapes* package

TimeScapes is a bundle of 10 Python programs. We will demonstrate the use the following four:

- **`agility.py`**: computes the RMS fluctuation activity
- **`terrain.py`**: computes cutoff-based and GMD-based activities
- **`tagging.py`**: performs pairwise-residue contact-distance analysis, using the activity functions from `agility` or `terrain`
- **`turning.py`**: performs pivot-angle analysis, using the activity functions from `agility` or `terrain`

The last two programs also output PDB files containing the ranking coefficients in the B-factor field, for visualization purposes.

`agility.py`

The basic usage is:

```
agility.py infile1 infile2 delta outname
```

where:

`infile1`: PDB file, used for mass assignment and optional least-squares fit.

`infile2`: Trajectory file.

`delta`: Full Width at Half Maximum (FWHM) of Gaussian-weighted window.

$$\text{FWHM} = 2\sqrt{2\ln 2}\sigma\,,$$ where σ is the standard deviation.

`outname`: Basaname prefix for output files.

terrain.py

The basic usage is:

```
terrain.py infile1 infile2 cut1 cut2 delta gtype outname
```

where:

`infile1`: PDB file, used for coarse-model assignment.

`infile2`: Trajectory file.

`cut1`: Values up to `cut1` are considered true contacts.

`cut2`: Values between `cut1` and `cut2` define the buffer zone in the recrossing filter, while values above `cut2` are considered non-contacts.

`delta`: Smoothing parameter for the Parzen window Gaussian function.

`gtype`: Graph type: either `GMD` or `Cutoff`.

`outname`: Basaname prefix for output files.

tagging.py

The basic usage is:

```
tagging.py infile1 infile2 infile3 outname
```

where:

`infile1`: PDB file, used for coarse-model assignment.

`infile2`: Trajectory file.

`infile3`: Time series data file containing the activity function (typically the `xxx_segmentation.dat` output from `agility` or `terrain`)

`outname`: Basaname prefix for output files.

`turning.py`

The basic usage is the same as for `tagging`:

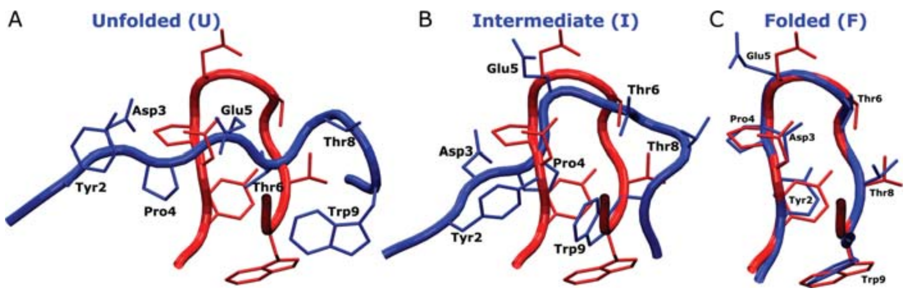
```
turning.py infile1 infile2 infile3 outname
```

where:

`infile1`:	PDB file, used for coarse-model assignment.
`infile2`:	Trajectory file.
`infile3`:	Time series data file containing the activity function (typically the `xxx_segmentation.dat` output from `agility` or `terrain`)
`outname`:	Basaname prefix for output files.

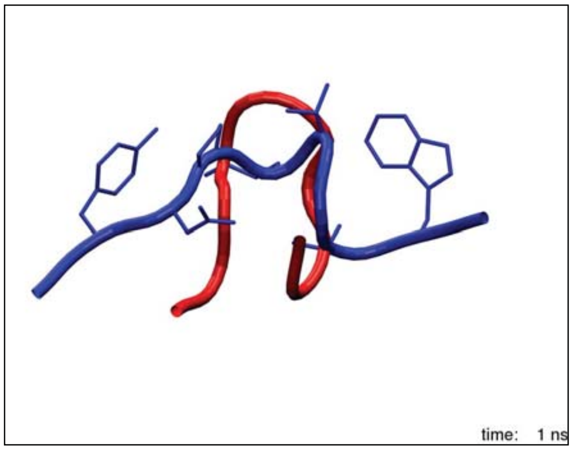
Running *TimeScapes* on the 'chignolin' trajectory

This is a 300 ns folding trajectory of a 10-residue peptide, using GaMD.



Miao et al. "Gaussian Accelerated Molecular Dynamics," *J. Chem. Theory Comput.* 2015.

Running *TimeScapes* on the ‘chignolin’ trajectory



300 ns folding trajectory, showing convergence to the native PDB structure (red)

Running *TimeScapes* on the ‘chignolin’ trajectory

Go to the ‘Extras’ folder. It contains a bash script and data files.

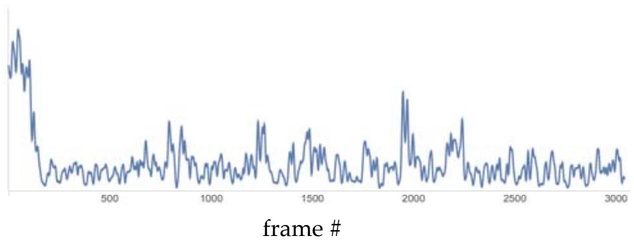
The bash script looks like:

```
#!/bin/bash
# Demo of TimeScapes 1.3 based on the chignolin trajectory
# Copyright Biomachina.org, 2016.

set -e
set -x

# adjust the version of gcc if needed:
: ${comp_version:=gcc-5.3.0}

for i in "$@"; do
  [[ $i == *=* ]] && eval $i
done

# adjust location of TimeScapes if needed, and uncomment your OS line:
# Mac:
prefix=$(greadlink -f ~/TimeScapes_1.3/objs/Darwin/x86_64/${comp_version})
# Cygwin:
# prefix=$(readlink -f ~/TimeScapes_1.3/objs/CYGWIN_NT-6.1/x86_64/${comp_version})
# Linux:
# prefix=$(readlink -f ~/TimeScapes_1.3/objs/Linux/x86_64/${comp_version})

PATH=${prefix}/bin:$PATH

mkdir -p timescapes_output
cd timescapes_output

agility.py ../demo.pdb ../demo.dcd 10 agility_10
terrain.py ../demo.pdb ../demo.dcd 2 3 10 GMD terrain_GMD_2_3_10
terrain.py ../demo.pdb ../demo.dcd 6 7 10 Cutoff terrain_Cut_6_7_10

tagging.py ../demo.pdb ../demo.dcd agility_10_segmentation.dat tagging_agil 2 0
tagging.py ../demo.pdb ../demo.dcd terrain_GMD_2_3_10_segmentation.dat tagging_terr_GMD 2 0
tagging.py ../demo.pdb ../demo.dcd terrain_Cut_6_7_10_segmentation.dat tagging_terr_Cut 2 0

turning.py ../demo.pdb ../demo.dcd agility_10_segmentation.dat turning_agil
turning.py ../demo.pdb ../demo.dcd terrain_GMD_2_3_10_segmentation.dat turning_terr_GMD
turning.py ../demo.pdb ../demo.dcd terrain_Cut_6_7_10_segmentation.dat turning_terr_Cut

set +x
```

`agility` output files

```
agility_10_minima.log
agility_10_segmentation.dat
agility_10_transitions.log
```

RMS fluctuation activity file, used by `tagging` and `turning`



```
frame #     value        derivative

       0 3.164843   0.000000 0
       1 3.135847  -0.028997 0
       2 3.103421  -0.032426 0
       3 3.069534  -0.033887 0
       4 3.036775  -0.032759 0
       5 3.006153  -0.030622 0
       6 2.978774  -0.027379 0
       7 2.956830  -0.021944 0
       8 2.940008  -0.016822 0
       9 2.929659  -0.010348 0
      10 2.929443  -0.000216 0
      11 2.944859   0.015416 0
      12 2.984017   0.039159 0
      . . . . . . . . . . . .
```

`terrain` output files

```
terrain_Cut_6_7_10_activity.dat
terrain_Cut_6_7_10_events.log
terrain_Cut_6_7_10_minima.log
terrain_Cut_6_7_10_segmentation.dat
terrain_Cut_6_7_10_transitions.log
```

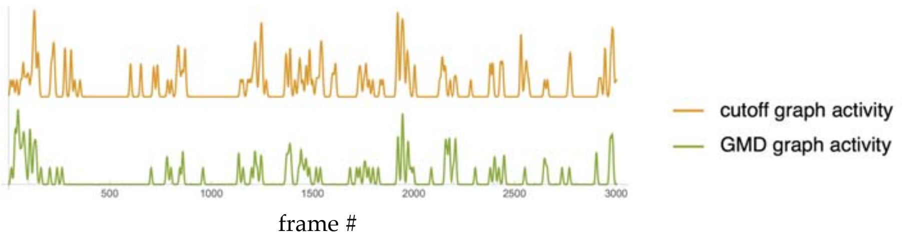
total, contact-forming, and -breaking activities (cutoff)

total activity and derivative (cutoff)

```
terrain_GMD_2_3_10_activity.dat
terrain_GMD_2_3_10_events.log
terrain_GMD_2_3_10_minima.log
terrain_GMD_2_3_10_segmentation.dat
terrain_GMD_2_3_10_transitions.log
```

total, contact-forming, and -breaking activities (GMD)

total activity and derivative (GMD)



`tagging` output files

```
tagging_agil.log
tagging_agil_dump.dat
tagging_agil_pairwise.dat
tagging_agil_pairwise_resname.dat
tagging_agil_pairwise_resname_count.dat
tagging_agil_pairwise_resname_max.dat
tagging_agil_pairwise_resname_normalized.dat
tagging_agil_projected.dat
tagging_agil_projected.pdb
tagging_terr_Cut.log
tagging_terr_Cut_dump.dat
tagging_terr_Cut_pairwise.dat
tagging_terr_Cut_pairwise_resname.dat
tagging_terr_Cut_pairwise_resname_count.dat
tagging_terr_Cut_pairwise_resname_max.dat
tagging_terr_Cut_pairwise_resname_normalized.dat
tagging_terr_Cut_projected.dat
tagging_terr_Cut_projected.pdb
tagging_terr_GMD.log
tagging_terr_GMD_dump.dat
tagging_terr_GMD_pairwise.dat
tagging_terr_GMD_pairwise_resname.dat
tagging_terr_GMD_pairwise_resname_count.dat
tagging_terr_GMD_pairwise_resname_max.dat
tagging_terr_GMD_pairwise_resname_normalized.dat
tagging_terr_GMD_projected.dat
tagging_terr_GMD_projected.pdb
```

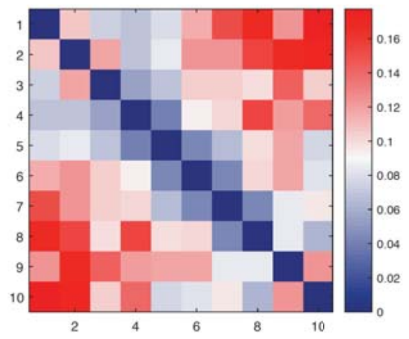
residue pairwise Pearson correlation matrix

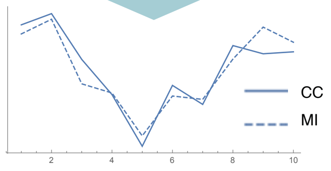
correlations projected to sequence

PDB file with the projected correlations in the B-factor field. See in VMD.

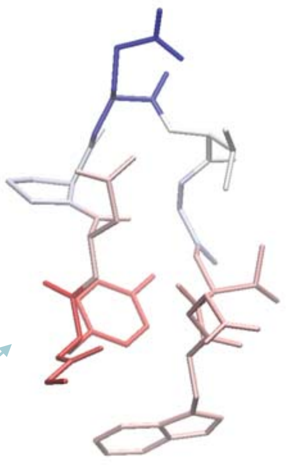
`tagging`: visualizing the output

`tagging_agil_pairwise.dat`





`tagging_agil_projected.dat`



`tagging_agil_projected.pdb`

turning: output files

```
turning_agil.log
turning_agil_differentials.dat
turning_agil_dihedrals.dat
turning_agil_dump.dat
turning_agil_pairwise_resname.dat
turning_agil_pairwise_resname_count.dat
turning_agil_pairwise_resname_max.dat
turning_agil_pairwise_resname_normalized.dat
turning_agil_turning.dat
turning_agil_turning.pdb
turning_terr_Cut.log
turning_terr_Cut_differentials.dat
turning_terr_Cut_dihedrals.dat
turning_terr_Cut_dump.dat
turning_terr_Cut_pairwise_resname.dat
turning_terr_Cut_pairwise_resname_count.dat
turning_terr_Cut_pairwise_resname_max.dat
turning_terr_Cut_pairwise_resname_normalized.dat
turning_terr_Cut_turning.dat
turning_terr_Cut_turning.pdb
turning_terr_GMD.log
turning_terr_GMD_differentials.dat
turning_terr_GMD_dihedrals.dat
turning_terr_GMD_dump.dat
turning_terr_GMD_pairwise_resname.dat
turning_terr_GMD_pairwise_resname_count.dat
turning_terr_GMD_pairwise_resname_max.dat
turning_terr_GMD_pairwise_resname_normalized.dat
turning_terr_GMD_turning.dat
turning_terr_GMD_turning.pdb
```

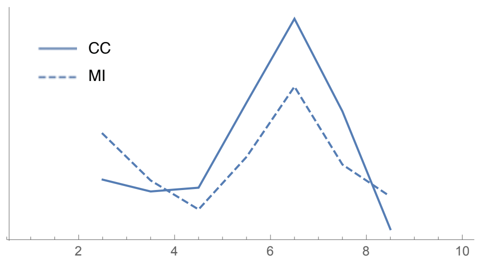
timeseries of dihedral angles and their abs. diff.

correlations of pivot angles projected to sequence

PDB file with the projected correlations in the B-factor field. See in VMD.

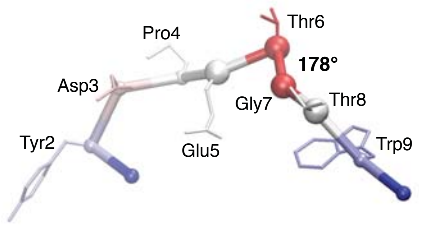
turning: visualizing the output

`turning_agil_turning.dat`



`turning_agil_turning.pdb`

frame: $t = 3.2$ ns



Pro4
Thr6
178°
Asp3
Gly7
Thr8
Tyr2
Glu5
Trp9