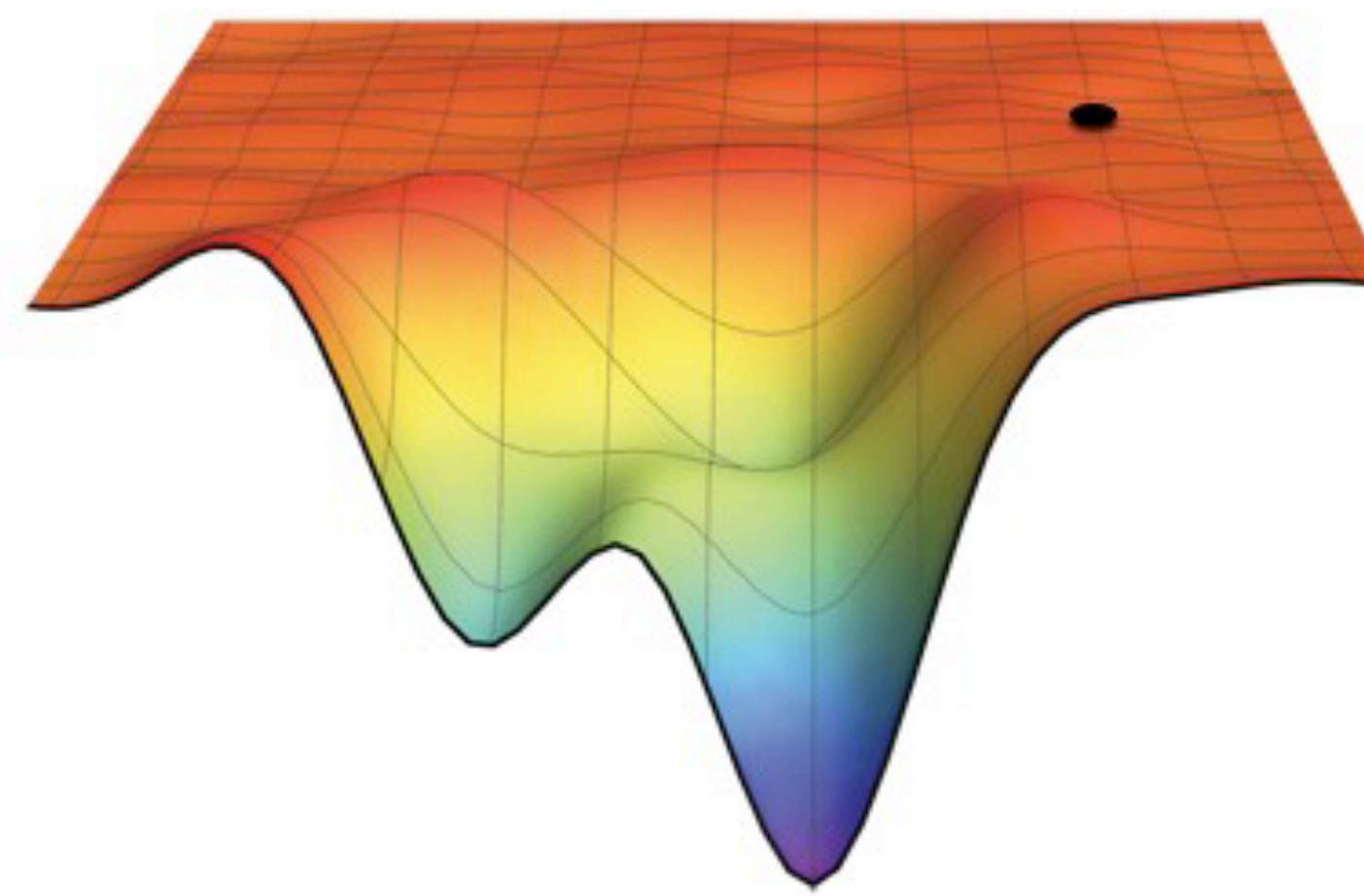
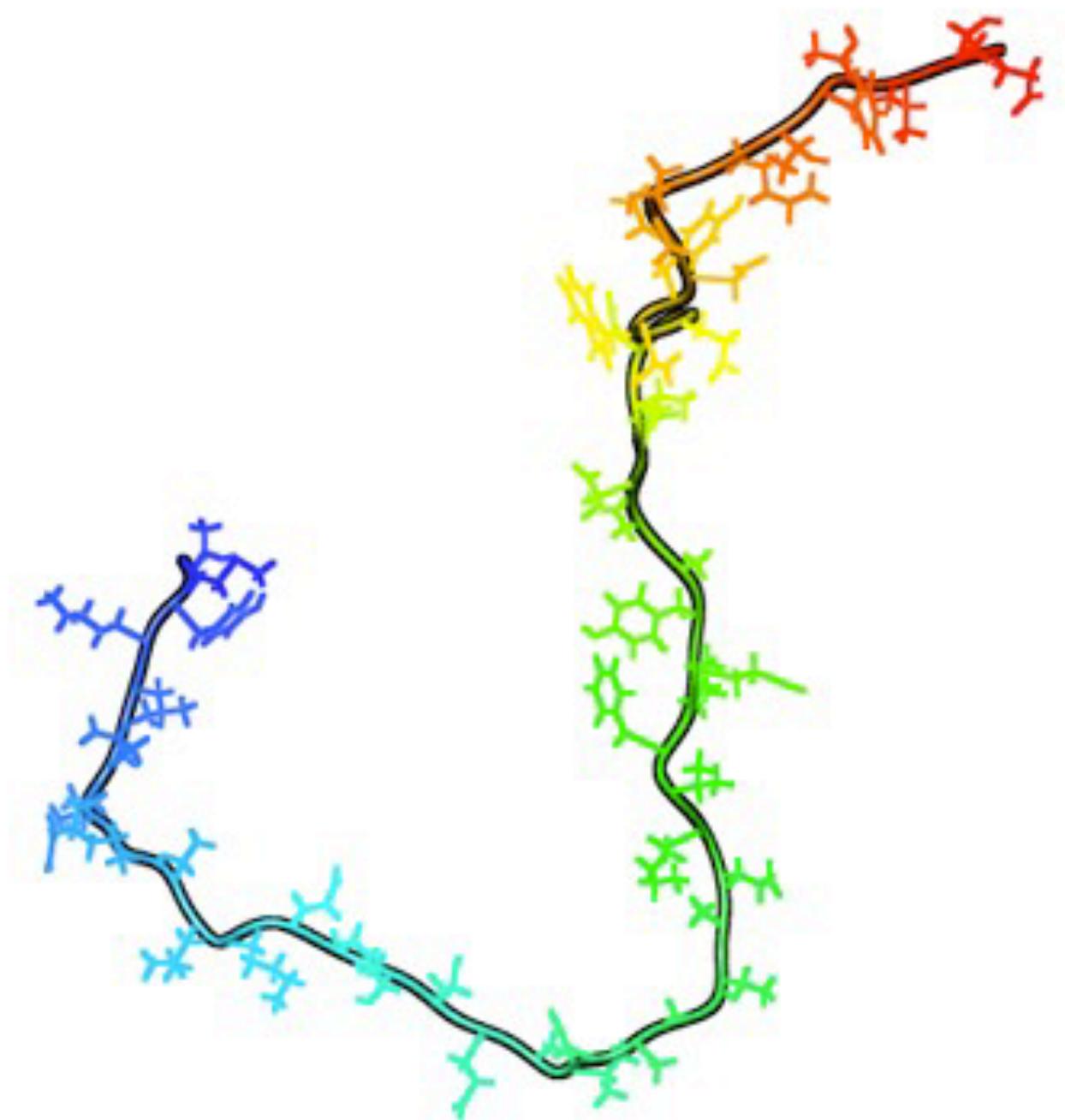


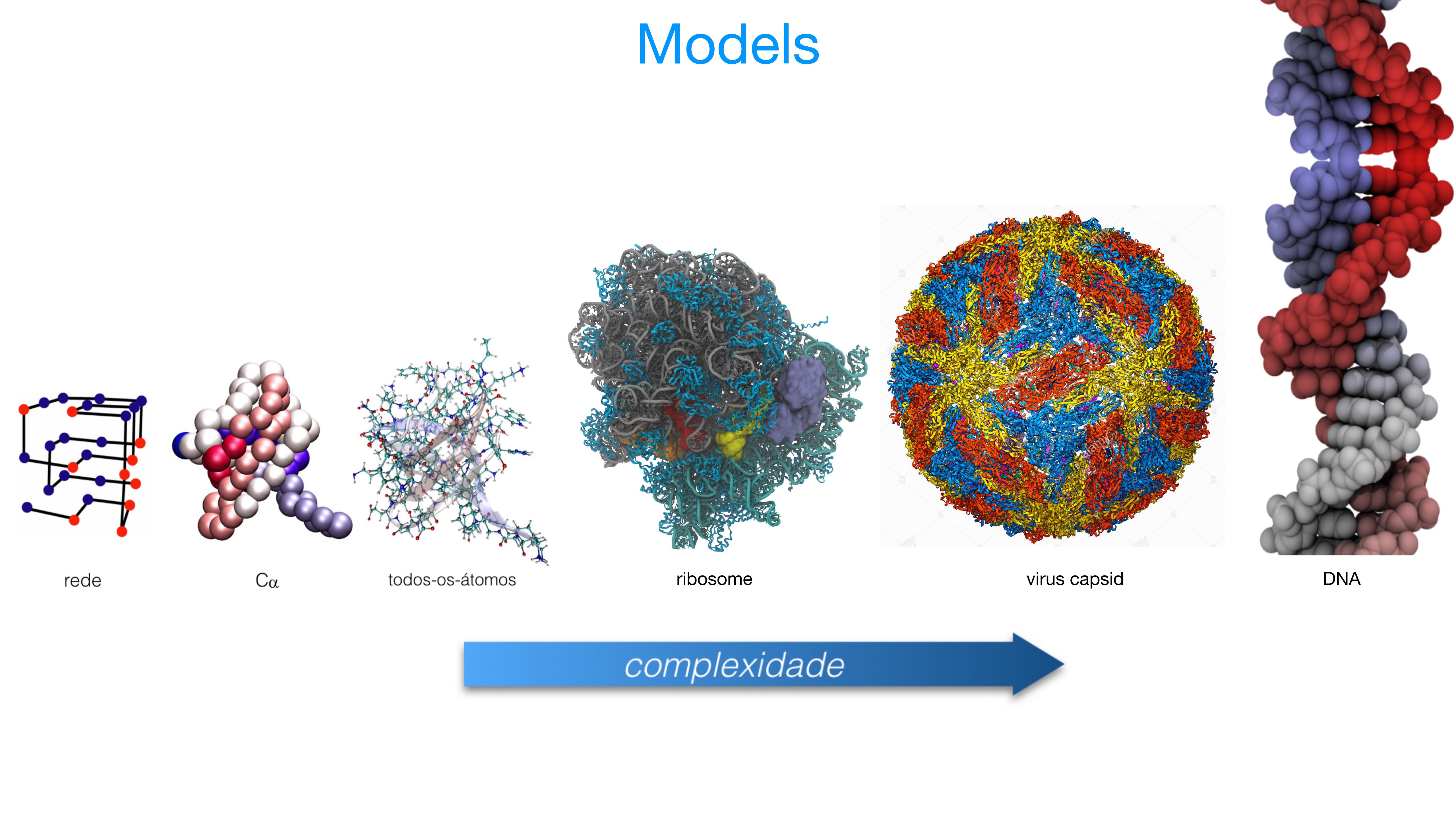
# **SMOG**

Ronaldo.Oliveira@UFTM.edu.br

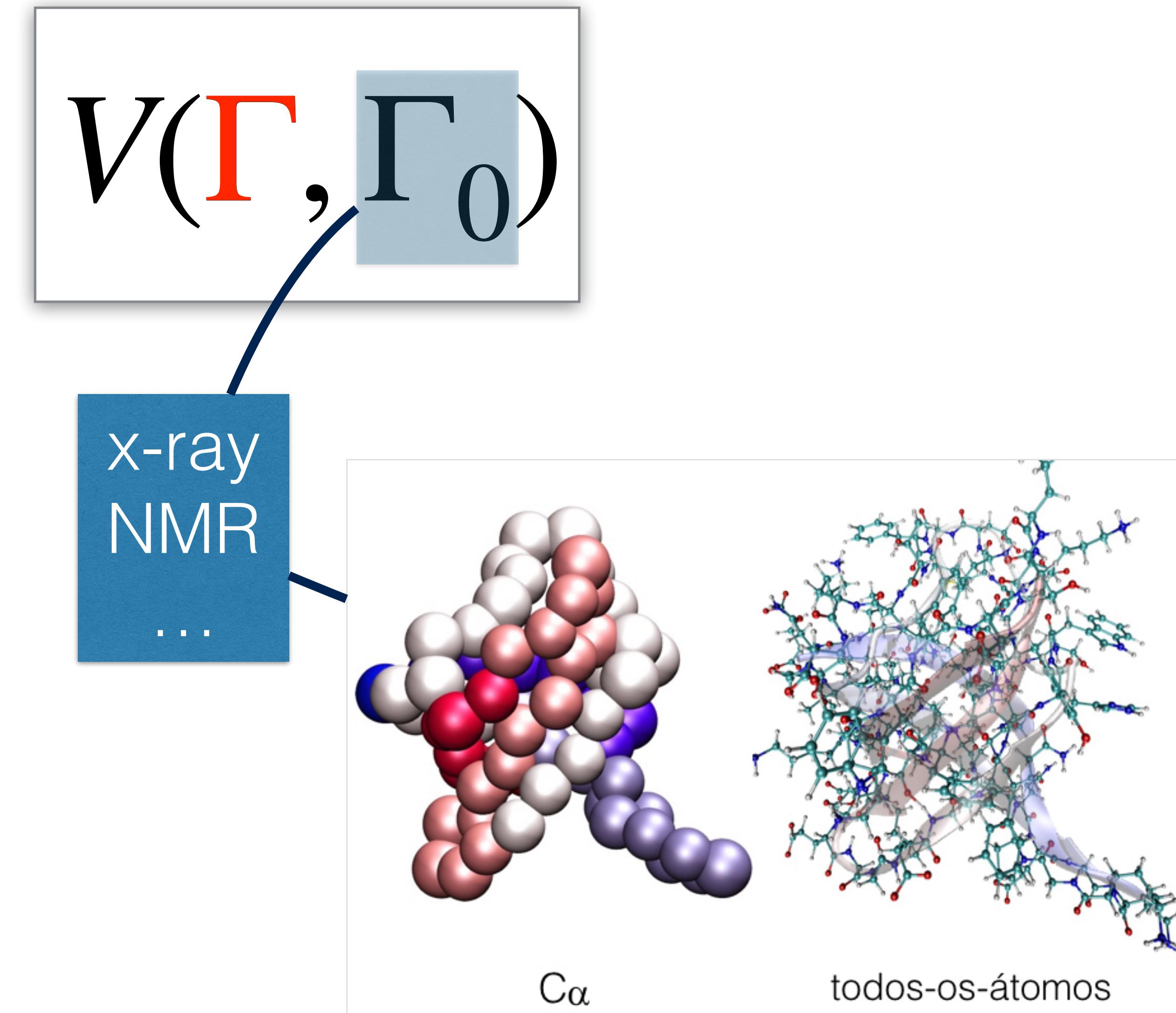
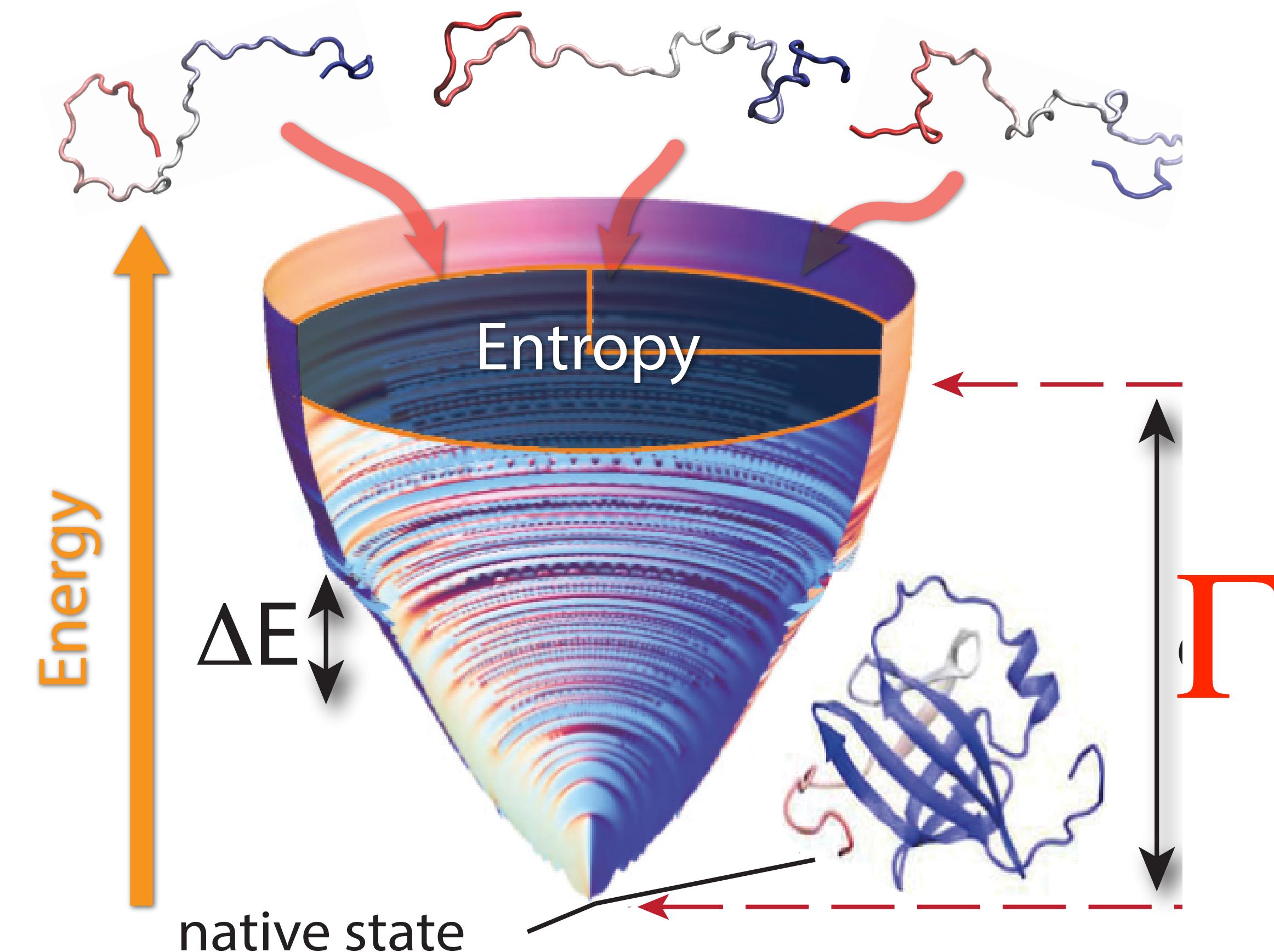
Ronaldo@RICE.edu



# Models



# Structure-Based Model (SBM)

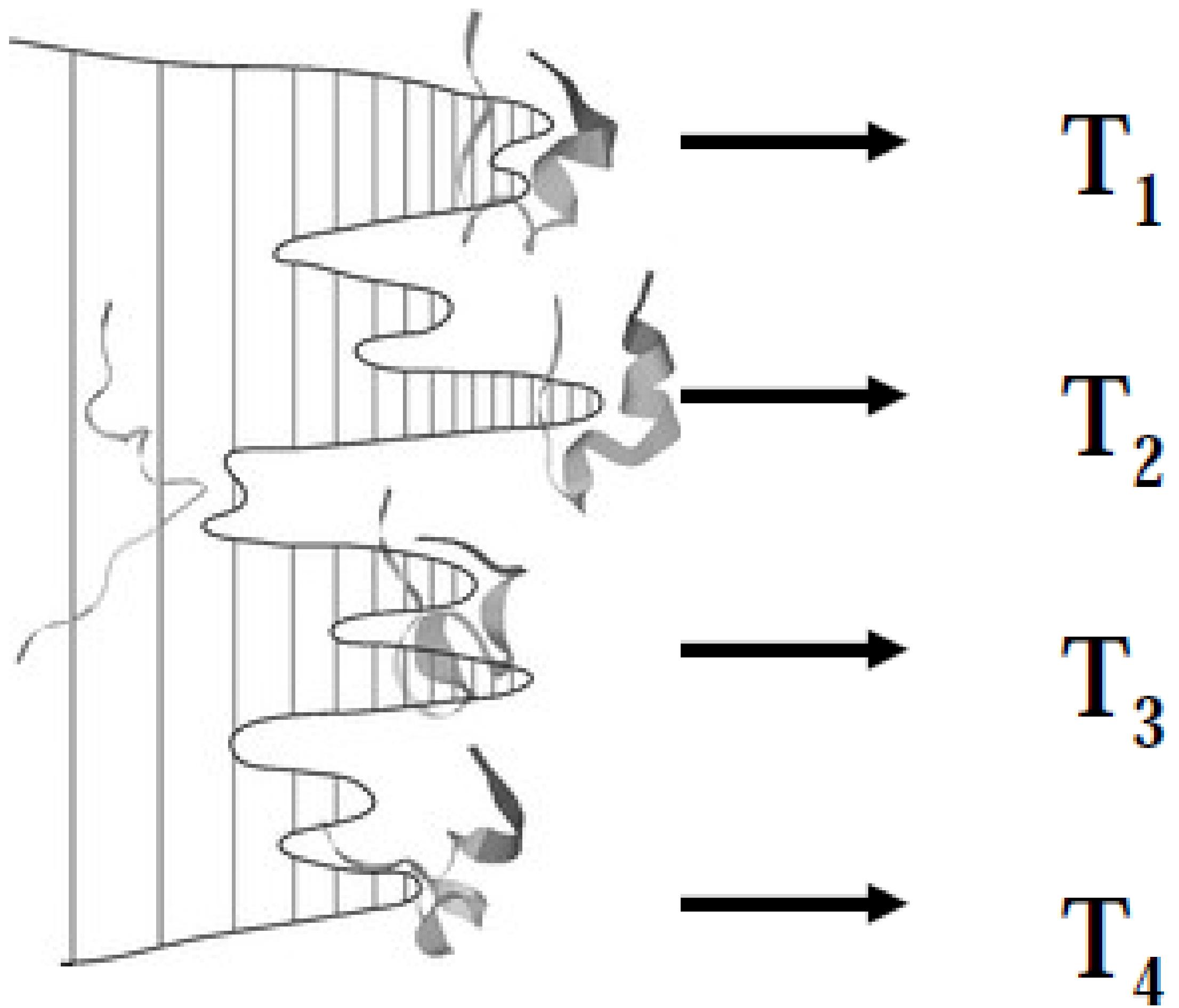


# Structure-Based Model (SBM)

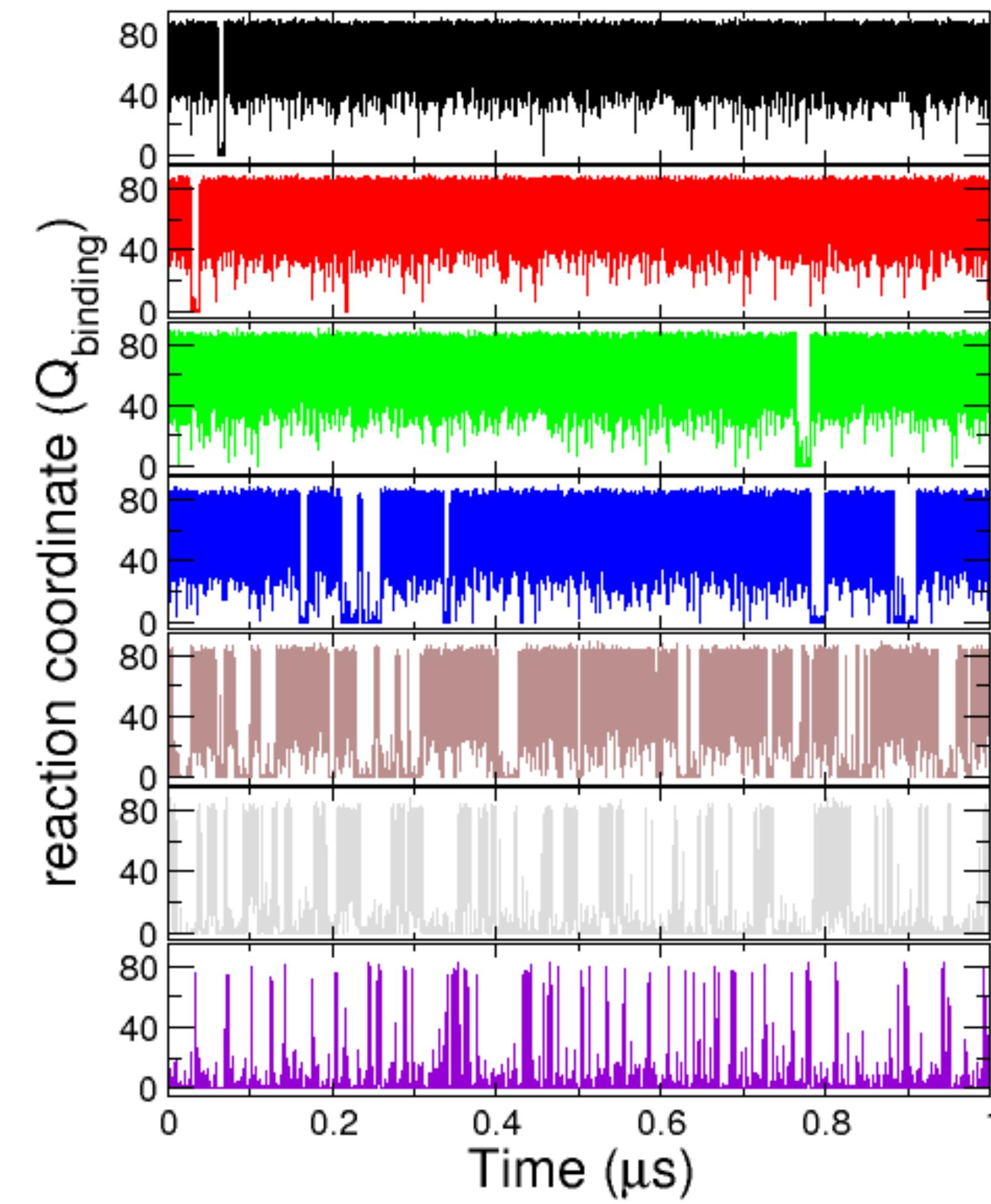
$$V(\Gamma, \Gamma_o) = \sum_{bonds} \epsilon_r (r - r_o)^2 + \sum_{angles} \epsilon_\theta (\theta - \theta_o)^2 + \sum_{dihedrals} \epsilon_\phi \left\{ [1 - \cos(\phi - \phi_o)] + \frac{1}{2} [1 - \cos(3(\phi - \phi_o))] \right\} + \sum_{contacts} \epsilon_C \left[ 5 \left( \frac{d_{ij}}{r_{ij}} \right)^{12} - 6 \left( \frac{d_{ij}}{r_{ij}} \right)^{10} \right] + \sum_{non-contacts} \epsilon_{NC} \left( \frac{\sigma_{NC}}{r_{ij}} \right)^{12}$$

- Native interactions govern folding
- Describes the folding or biological mechanism of interest

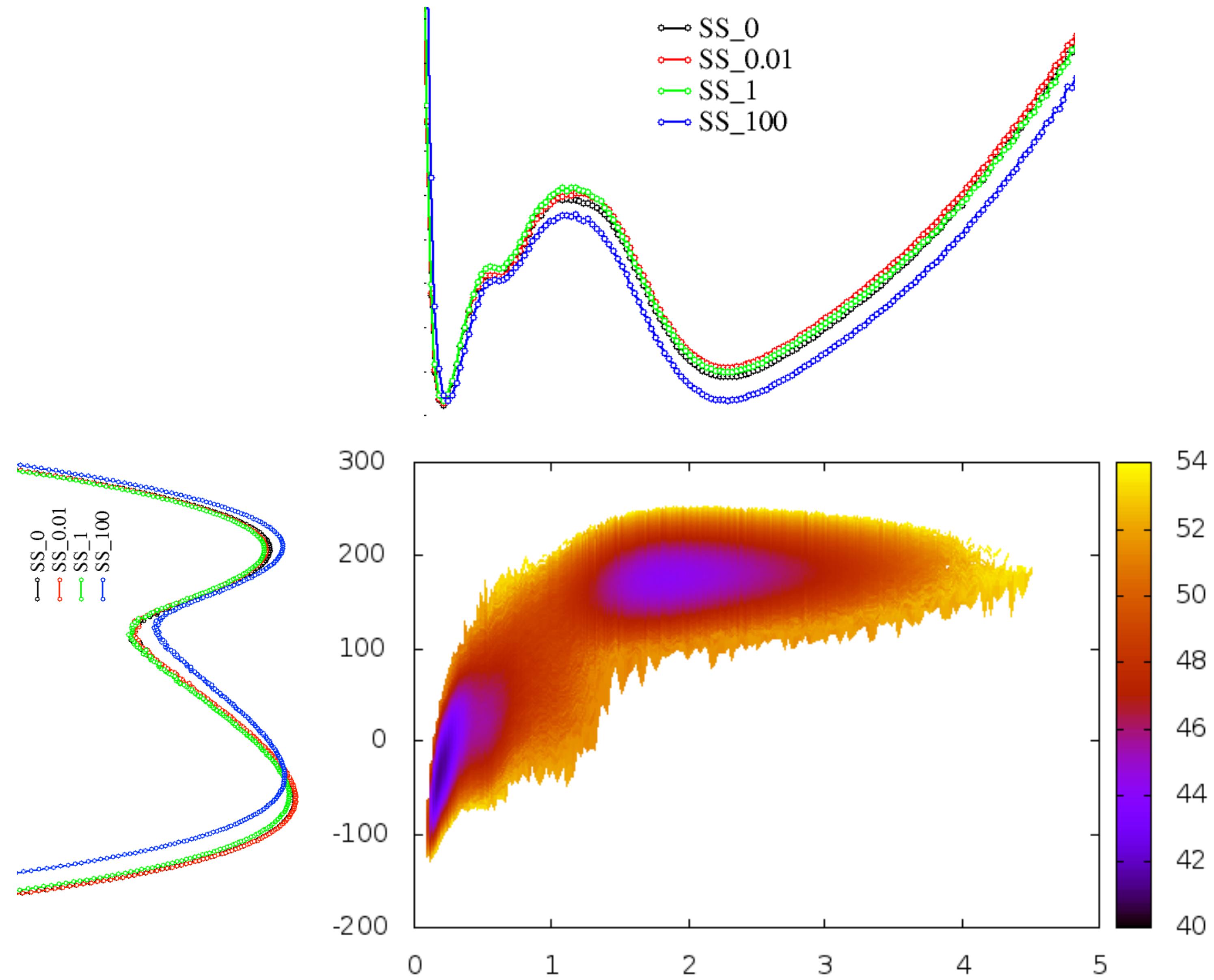
# Stochastic Dynamics at $\neq$ Temperatures



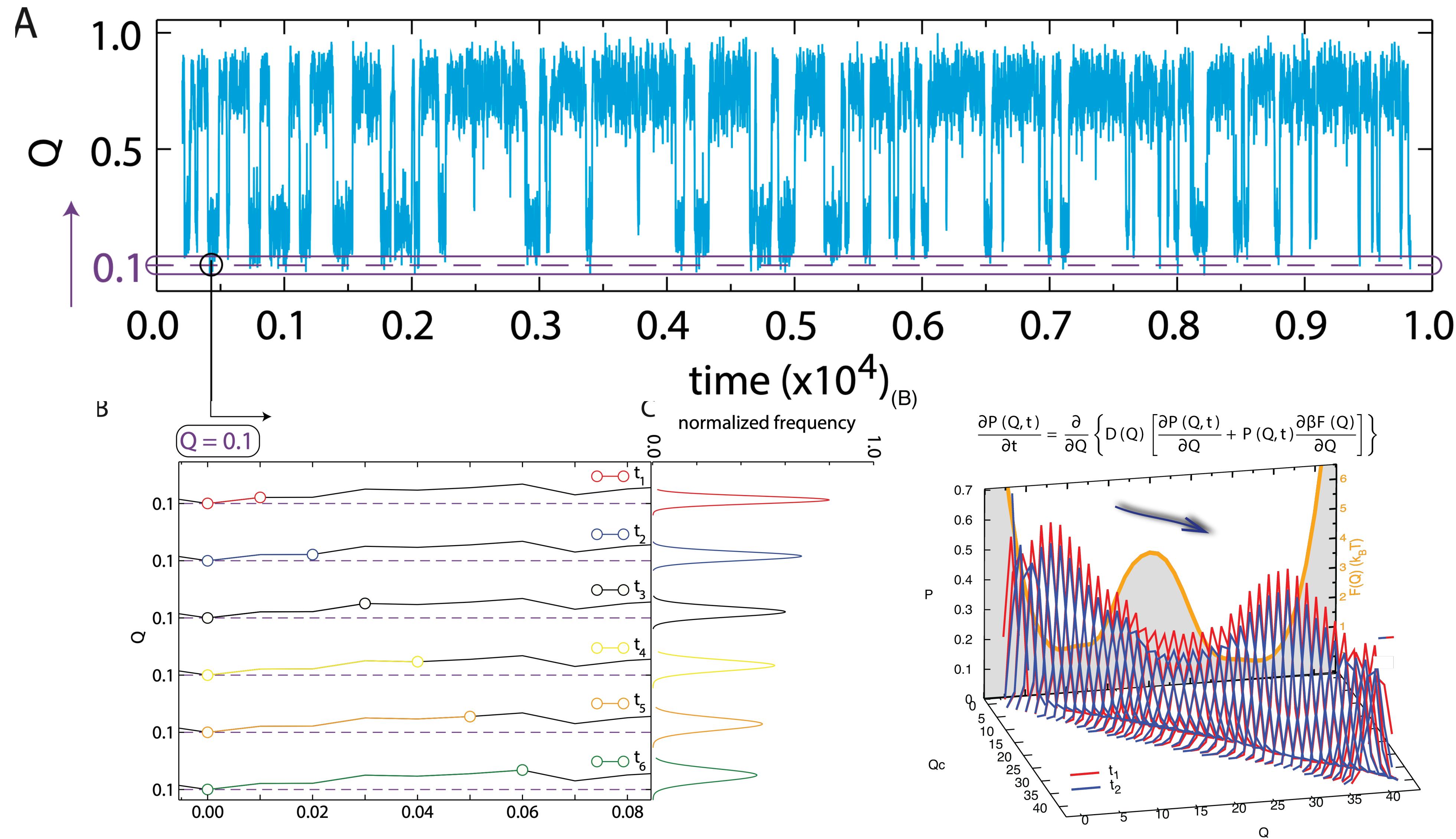
$T_1$   
 $T_2$   
 $T_3$   
 $T_4$



# Thermodynamic analysis



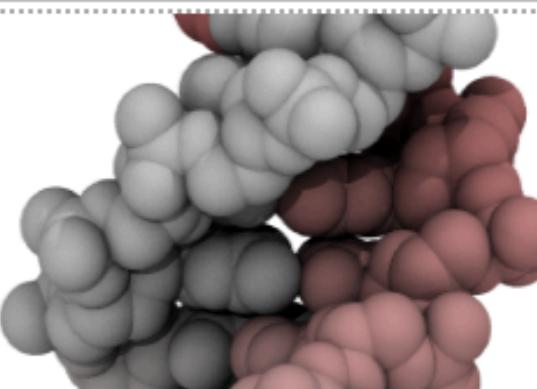
# Kinetic analysis





Northeastern

Introduction
Prepare a Simulation
Run a Simulation
Additional Tools
Tutorials and Examples
The Contact Map
Source Code Extensions
FAQs
News
Download SMOG (v2)
- Source Code
- Docker Container
- Singularity Container
OpenSMOG
Version History
Getting Help
Citing SMOG
Acknowledgements



# SMOG: Structure-based Models for Biomolecules

## A very brief introduction to structure-based modeling

While originally developed for the study of protein folding, over the last 15 years we have worked to extend the applicability of structure-based models to investigate a broad range of biomolecular dynamics, including domain rearrangements in proteins, folding and ligand binding in RNA and large-scale rearrangements in ribonucleoprotein assemblies (e.g. the [ribosome](#)). In its simplest form, a structure-based model defines a particular structure (typically obtained from X-ray, cryo-EM or NMR) as the global energetic minimum. Oftentimes, this single-basin description can provide rich information about complex processes. When this simple representation is insufficient to describe the process of interest, these models can be easily extended to include multiple basins of attraction, as well as describe non-specific effects (e.g. electrostatic or solvation effects). While coarse-grained variants of this model have had considerable success in expanding our understanding of protein folding, all-atom structure-based models are now widely used to study all types of biomolecules (complete descriptions can be found [elsewhere](#)). Additionally, with the computational simplicity of the models, they are proving to be very useful for structural modeling purposes, including atomic modeling of cryo-EM, SAXS and biochemical data.

To increase the accessibility of these modeling techniques, we provide the smog-server webtool, as well as the downloadable [SMOG 2](#) software, which are able to generate SMOG models for use with Gromacs, NAMD, LAMMPS and OpenMM. While the webtool provides support for proteins, nucleic acids, and some ligands, SMOG 2 enables the application of structure-based models to any type of polymer-ligand system. For in-depth descriptions of the underlying models, and for examples of how to use the models, we refer you to the [original publications](#) and recent [review articles](#).

## How to prepare and simulate a SMOG model for a specific biomolecule

While we provide a range of examples on the [Tutorials and Examples](#) page, here are the basic steps for using a SMOG model:

**Step 1:** [Prepare the simulation](#). You first need to generate a force field for your molecular system. This may be accomplished in two ways.

- [Use the Webtool](#) We have made some SMOG models available through our smog-server web interface. All you need to do is provide a PDB structure file, specify which parameters you would like to use, and the webtool will generate the input files that will define the model.
- [Use SMOG 2](#) You may alternately use the standalone version, called SMOG 2, which you can download and run on your machine. SMOG 2 provides far more comprehensive support for SMOG models. It allows you to define your own styles of SMOG models, include additional types of interactions (e.g. electrostatics) and define new residues (e.g. post-transcriptional modifications). For large-scale systems and maximal OpenMM compatibility, it is strongly recommended that you use SMOG 2.

**Step 2:** [Run the simulation](#). Due to some technical differences between structure-based models and many other commonly used models, there are a few additional considerations that will ensure maximal performance of your calculations. Click [here](#) for details on how to run a SMOG model in Gromacs, NAMD, OpenMM, or LAMMPS. It is highly recommended that you read this information, even if you are a simulation expert.

**Step 3:** [Analyze the results](#). Once your simulations have finished, there are many forms of post-analysis that may be of interest. While there are analysis tools available with the Gromacs distribution, and VMD has a powerful Tcl scripting interface, we have also made some scripts and programs available [here](#). If you would like to contribute analysis scripts, please send us a copy and we will make them available.

This resource is provided by the Center for Theoretical Biological Physics.

Please direct questions and comments to [info@smog-server.org](mailto:info@smog-server.org).

Page created and maintained by [Jeff Noel](#) and [Paul Whitford](#)

# Proteins G/L

## Exploring the Folding Mechanism of Small Proteins GB1 and LB1

Qianyi Cheng,<sup>†,||<sup>ID</sup></sup> InSuk Joung,<sup>‡,||</sup> Juyong Lee,<sup>‡</sup> Kunihiro Kuwajima,<sup>¶||</sup> and Jooyoung Lee<sup>\*,§,||<sup>ID</sup></sup>

<sup>†</sup>Department of Chemistry, University of Memphis, Memphis, Tennessee 38152, United States

<sup>‡</sup>Department of Chemistry, Kangwon National University, Chuncheon 24341, South Korea

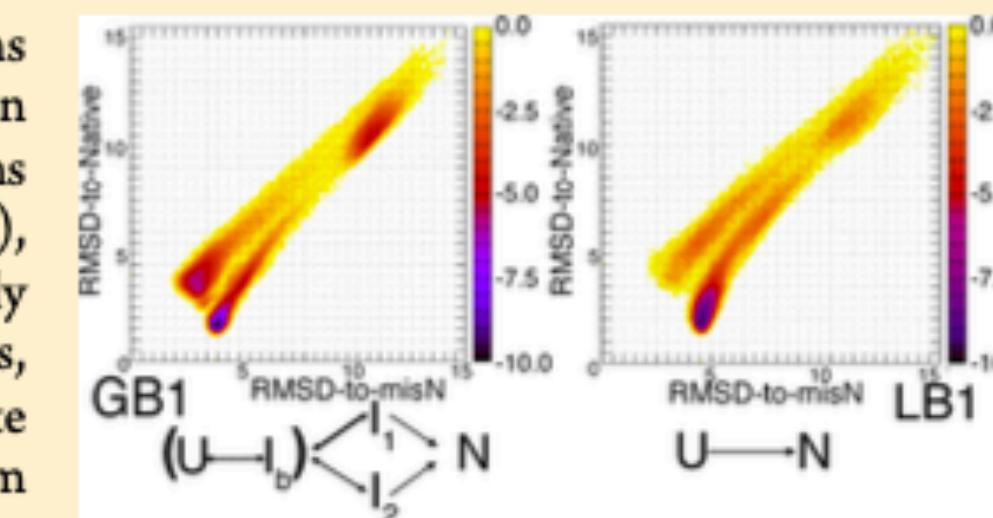
<sup>¶</sup>Department of Physics, University of Tokyo, Tokyo 113-0033, Japan

<sup>§</sup>Center for In Silico Protein Science, Korea Institute for Advanced Study, Seoul 02455, South Korea

<sup>||</sup>School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, South Korea

### Supporting Information

**ABSTRACT:** The computational atomistic description of the folding reactions of the B1 domains, GB1 and LB1, of protein G and protein L, respectively, is an important challenge in current protein folding studies. Although the two proteins have overall very similar backbone structures ( $\beta$ -hairpin– $\alpha$ -helix– $\beta$ -hairpin), their apparent folding behaviors observed experimentally were remarkably different. LB1 folds in a two-state manner with the single-exponential kinetics, whereas GB1 folds in a more complex manner with an early stage intermediate that may exist on the folding pathway. Here, we used a new method of all-atom molecular dynamics simulations to investigate the folding mechanisms of GB1 and LB1. With the Lorentzian energy term derived from the native structure, we successfully observed frequent folding and unfolding events in the simulations at a high temperature (414 K for GB1 or 393 K for LB1) for both the proteins. Three and two transition-state structures were predicted for the GB1 and LB1 folding, respectively, at the high temperature. Two of the three transition-state structures of GB1 have a better formed second  $\beta$ -hairpin. One of the LB1 transition states has a better formed first hairpin, and the other has both hairpins equally formed. The structural features of these transition states are in good agreement with experimental transition-state analysis. At 300 K, more complex folding processes were observed in the simulations for both the proteins. Several intermediate structures were predicted for the two proteins, which led to the conclusion that both the proteins folded through similar mechanisms. However, the intermediate state accumulated in a sufficient amount only in the GB1 folding, which led to the double-exponential feature of its folding kinetics. On the other hand, the LB1 folding kinetics were well fitted by a single-exponential function. These results are fully consistent with those previously observed experimentally.



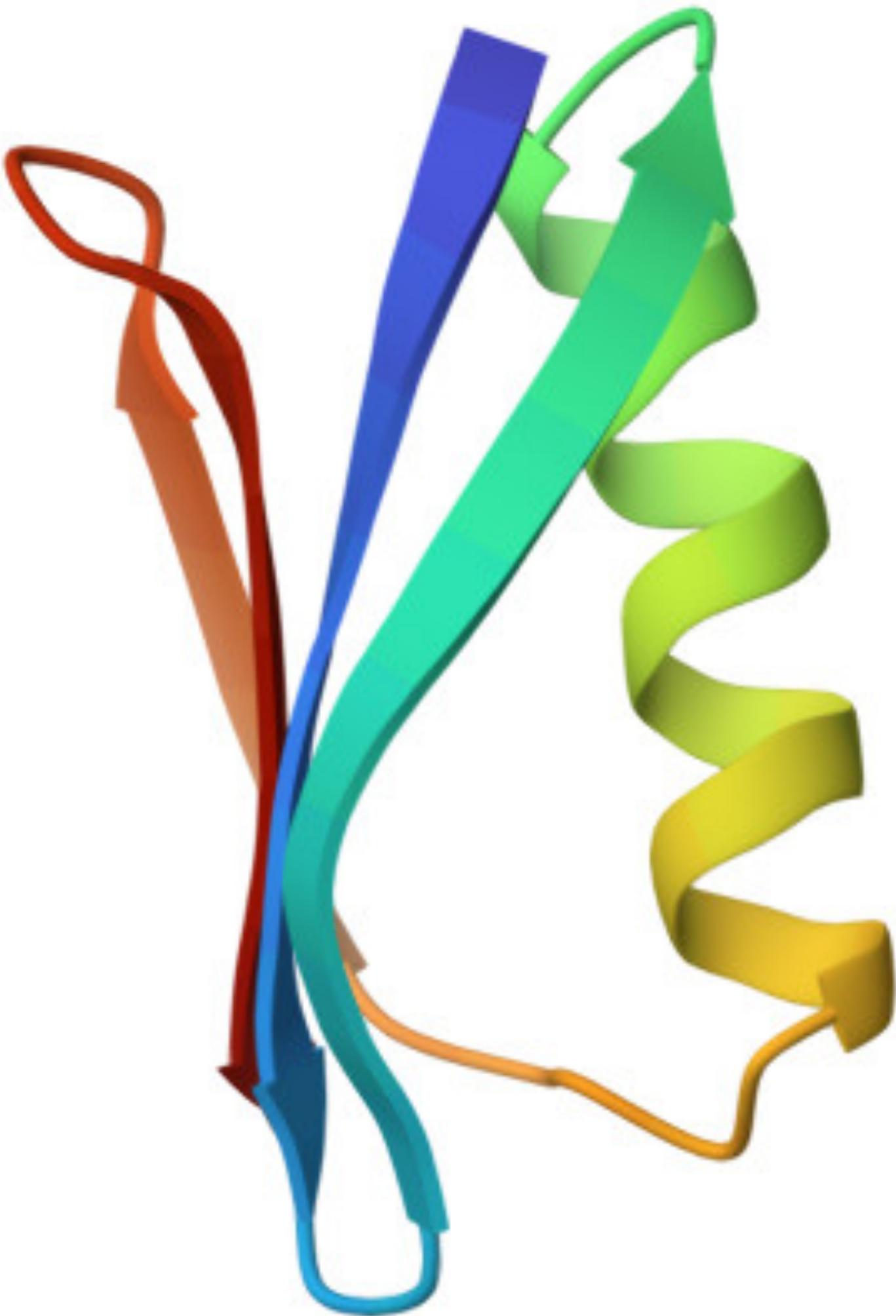
# GB1

Download

- PDB file: 1PGB

B1 IMMUNOGLOBULIN-BINDING DOMAIN OF  
STREPTOCOCCAL PROTEIN G AND COMPARISON WITH NMR

residues 1–56



# Setup

## 1# Pull container with smog

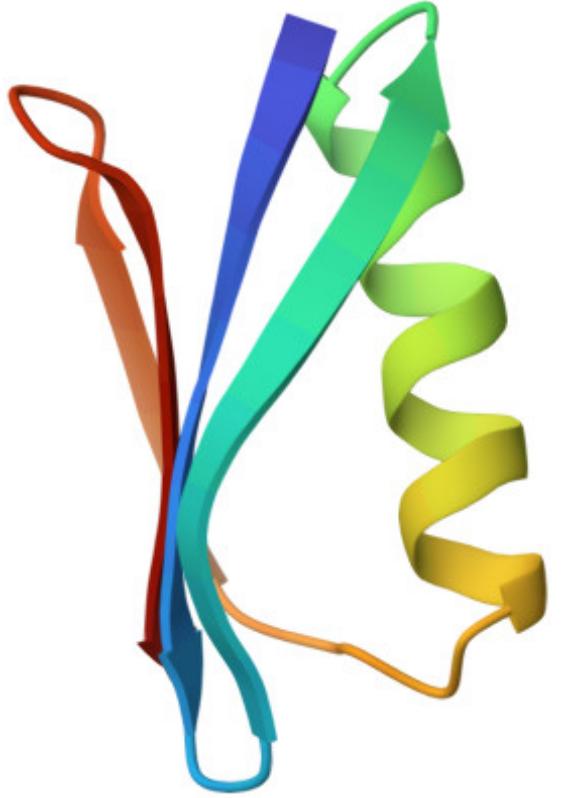
```
docker pull smogserver/smogopensmog:s2.4.5-os1.1.1
```

## 2# Launch container

```
docker run -it --rm -v $(pwd) :/workdir smogserver/  
smogopensmog:s2.4.5-os1.1.1
```

```
cd /workdir/
```

# GB1



- Run the code to prepare the pdb file

```
smog_adjustPDB -h
```

```
smog_adjustPDB -i 1pgb_original.pdb -removeH -removewater  
-o gb1_adjusted.pdb
```

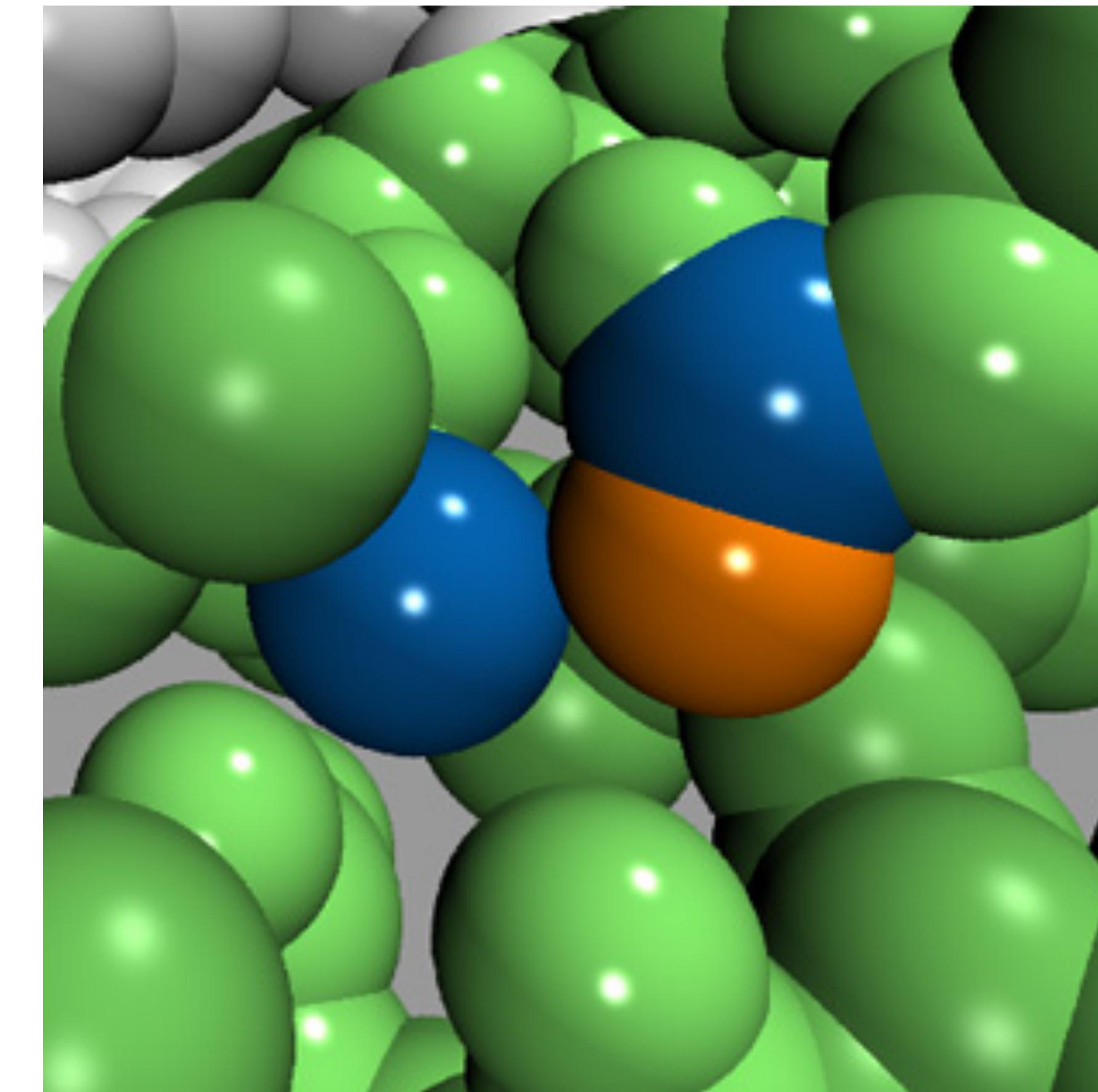
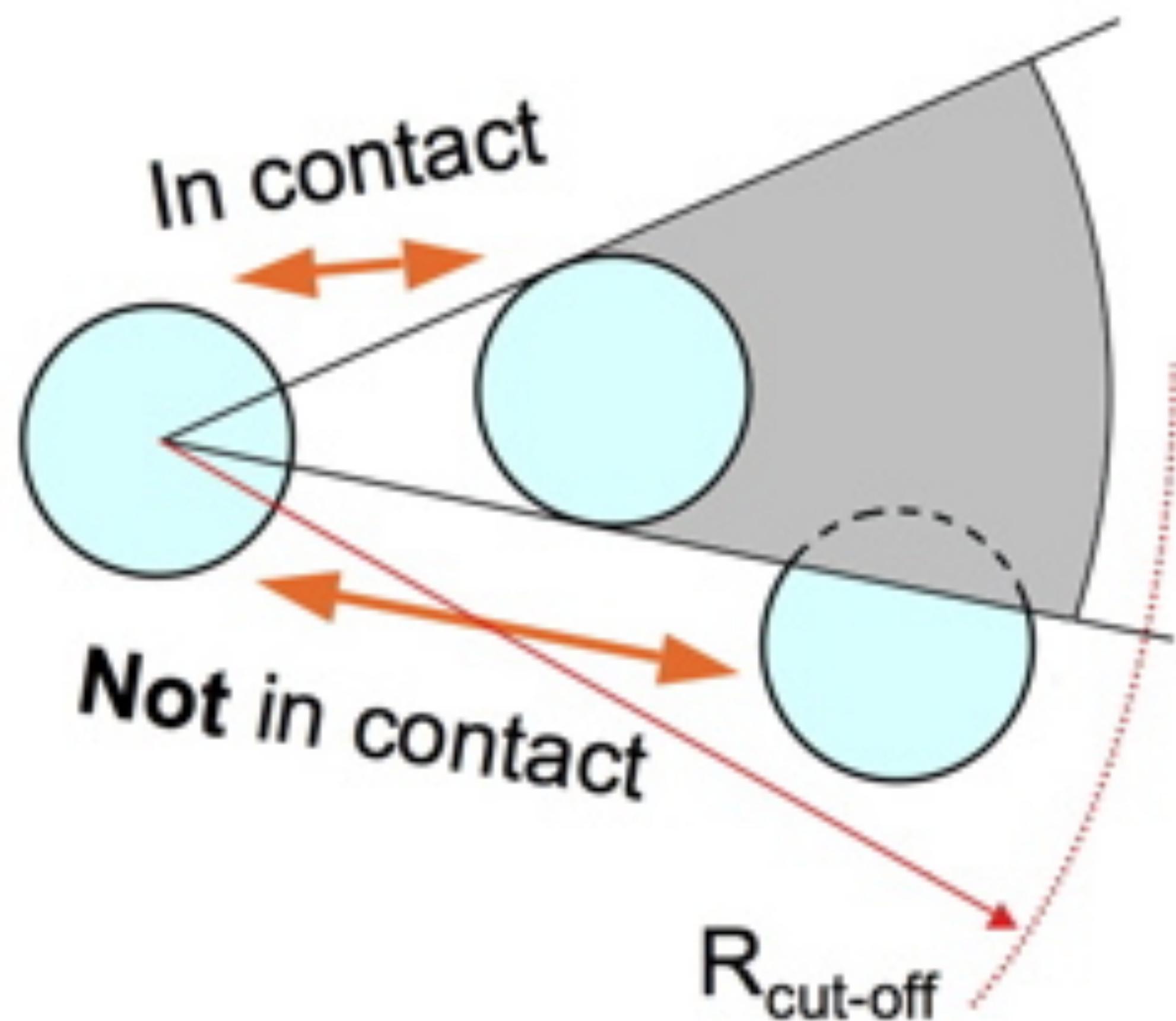
- Now, prepare for smog

```
smog2 -h
```

- Calpha first

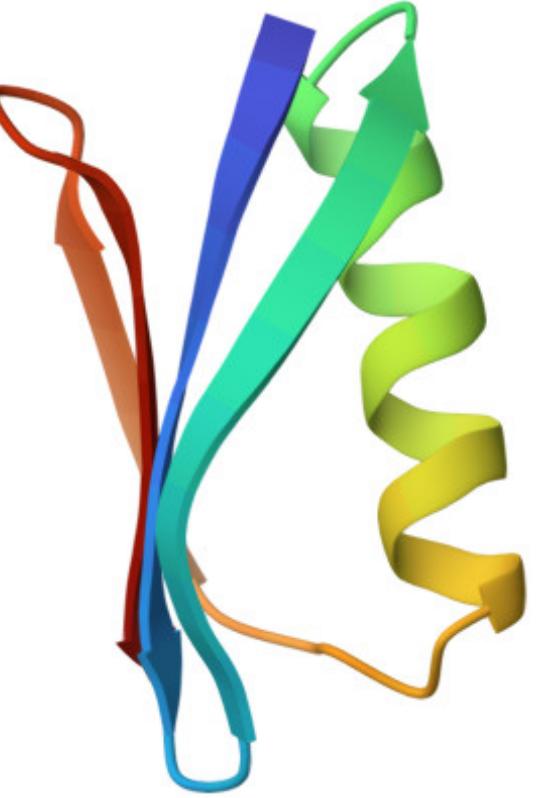
```
smog2 -i gb1_adjusted.pdb -CA -dname gb1.CA
```

# Shadow Contact Map



4 Å cut-off map of CI2 (chymotrypsin inhibitor 2)  
 $S > 0.5 \text{ \AA}$  shadows the orange

# GB1



- Calpha on openSMOG

```
smog2 -i gb1_adjusted.pdb -CA -dname gb1.CA -openSMOG
```

# run.CA.py

```
from OpenSMOG import SBM

#Choose some basic runtime settings. We will call our system gb1.CA
SMOGRun = SBM(name='gb1.CA', time_step=0.0005, collision_rate=1.0, r_cutoff=1.1, temperature=0.5)

#Select a platform and GPU IDs (if needed)
SMOGRun.setup_openmm(platform='cpu',GPUindex='default')

#Decide where to save your data (here or output_folder)
SMOGRun.saveFolder('.')

#You may optionally set some input file names to variables
SMOG_grofile = 'gb1.CA.gro'
SMOG_topfile = 'gb1.CA.top'
SMOG_xmlfile = 'gb1.CA.xml'

#Load your force field data
SMOGRun.loadSystem(Grofile=SMOG_grofile, Topfile=SMOG_topfile, Xmlfile=SMOG_xmlfile)

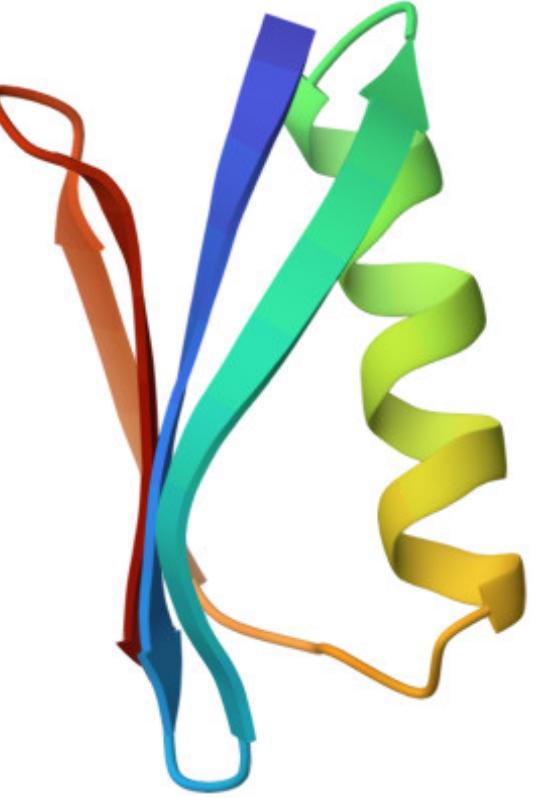
#Create the context, and prepare the simulation to run
SMOGRun.createSimulation()

#Perform energy minimization
SMOGRun.minimize(tolerance=1)

#Decide how frequently to save data
SMOGRun.createReporters(trajectory=True, energies=True, energy_components=True, interval=10**2)

#Launch the simulation
SMOGRun.run(nsteps=10**4, report=True, interval=10**2)
```

# GB1



- All-atoms on openSMOG  
smog2 -i gb1\_adjusted.pdb -CA -dname gb1.CA -openSMOG

# run\_AA.py

```
from OpenSMOG import SBM

#Choose some basic runtime settings. We will call our system gb1.CA
SMOGrun = SBM(name='gb1.AA', time_step=0.002, collision_rate=1.0, r_cutoff=0.65, temperature=0.5)

#Select a platform and GPU IDs (if needed)
SMOGrun.setup_openmm(platform='cpu',GPUindex='default')

#Decide where to save your data (here or output_folder)
SMOGrun.saveFolder('.')

#You may optionally set some input file names to variables
SMOG_grofile = 'gb1.AA.gro'
SMOG_topfile = 'gb1.AA.top'
SMOG_xmlfile = 'gb1.AA.xml'

#Load your force field data
SMOGrun.loadSystem(Grofile=SMOG_grofile, Topfile=SMOG_topfile, Xmlfile=SMOG_xmlfile)

#Create the context, and prepare the simulation to run
SMOGrun.createSimulation()

#Perform energy minimization
SMOGrun.minimize(tolerance=1)

#Decide how frequently to save data
SMOGrun.createReporters(trajectory=True, energies=True, energy_components=True, interval=50)

#Launch the simulation
SMOGrun.run(nsteps=2*10**3, report=True, interval=50)
```



# LB1

- PDB file: 1HZ6

B1 DOMAIN OF PROTEIN L FROM PEPTOSTREPTOCOCCUS MAGNUS WITH A TYROSINE TO TRYPTOPHAN SUBSTITUTION

residues 4–64

=> Compare with GB1



**Thanks a lot!**