



# ProDy

Protein Dynamics & Sequence Analysis

*ProDy*

Ahmet Bakan

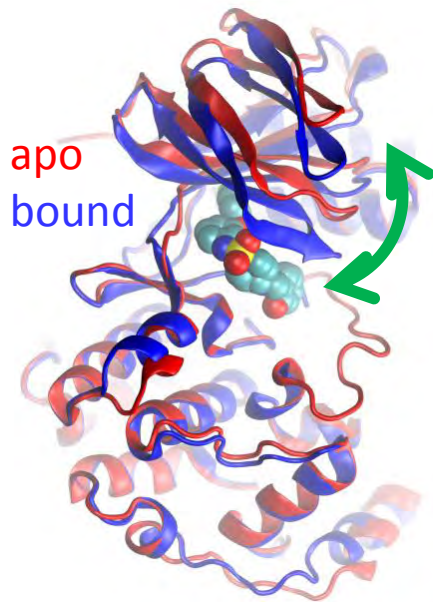


# ProDy

Protein Dynamics & Sequence Analysis

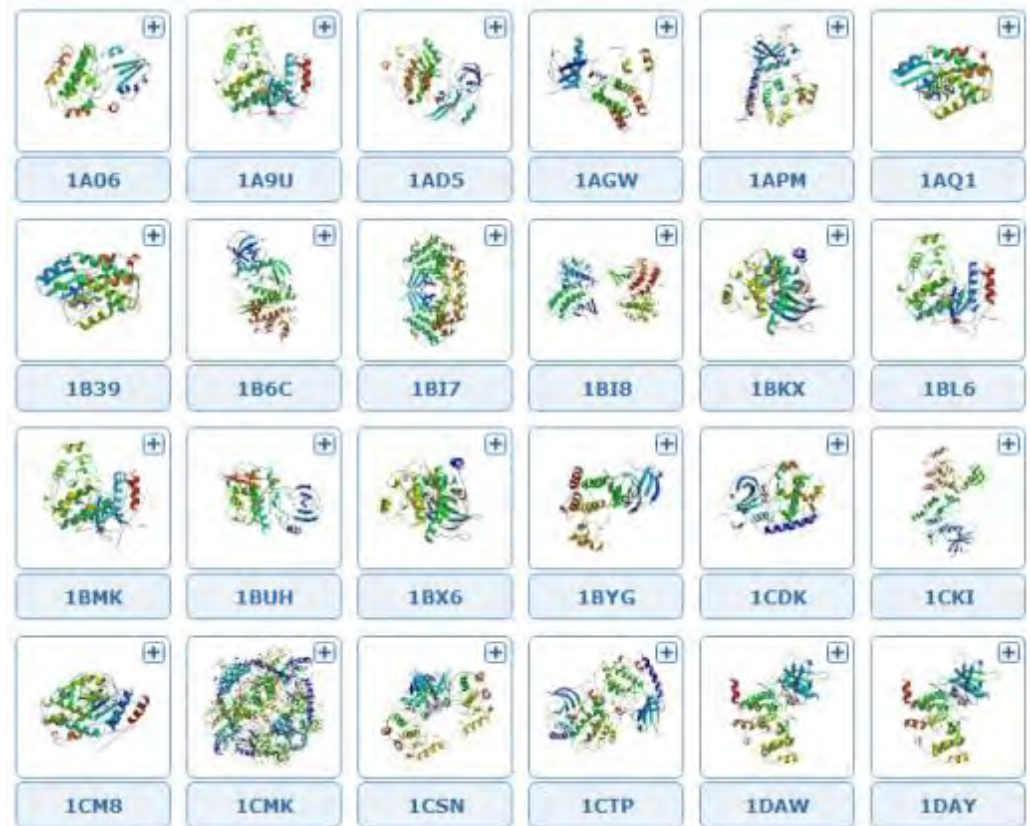
## Structure Analysis

Analyzing a **pair of structures**



is easy using VMD,  
Chimera, PyMol, or etc.

Analyzing a **hundreds of structures**



there is **ProDy** for that

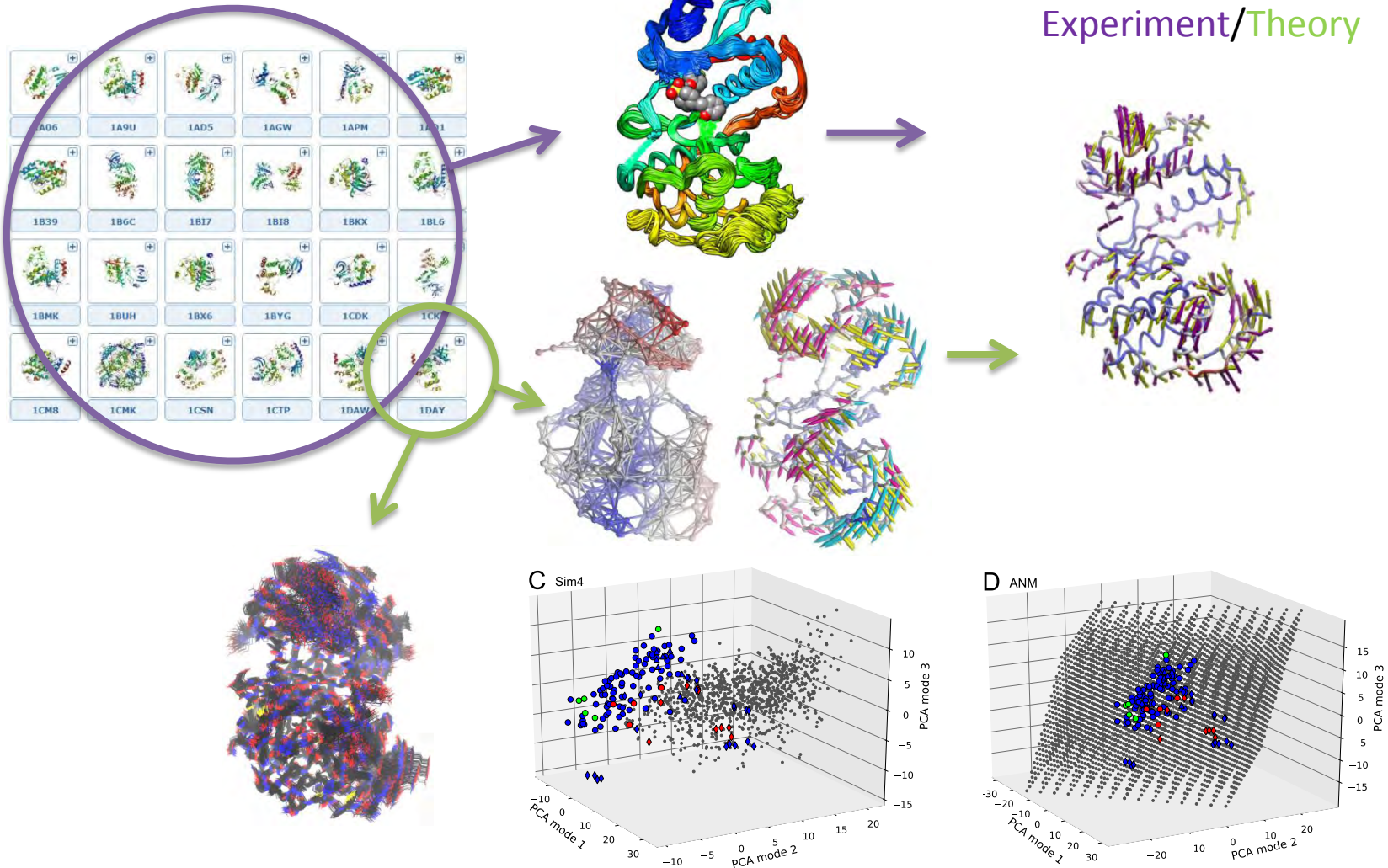


# ProDy

Protein Dynamics & Sequence Analysis

## Experiment & Theory

Experiment/Theory





What is *ProDy*?

*ProDy* is an **Application Programming Interface (API)**  
or a **toolkit**

designed for

analysis of **large** and **heterogeneous** structural datasets,

normal mode analysis of **elastic network models**,

**essential dynamics analysis** of trajectories,

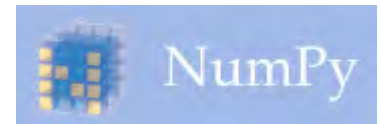
and, **application development**



# ProDy

Protein Dynamics & Sequence Analysis

## Interactive Usage



# IP[y]: IPython Interactive Computing

```

shakan@orko:~$
578
RuntimeError: Coordinates are not set. Call select_residues method.

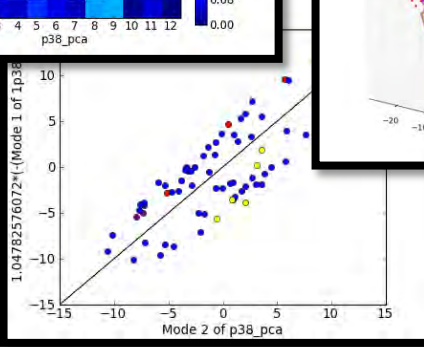
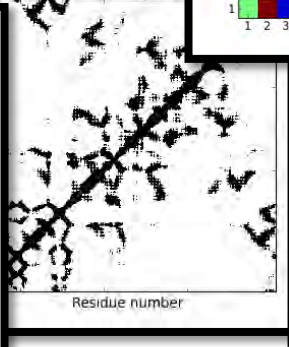
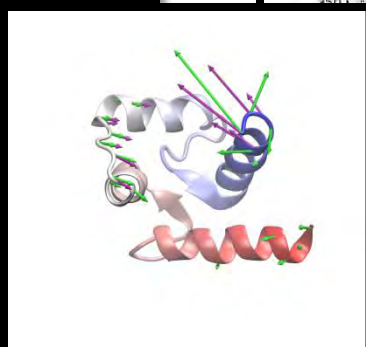
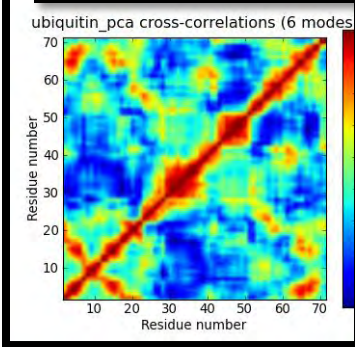
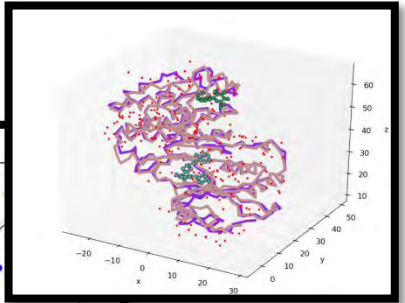
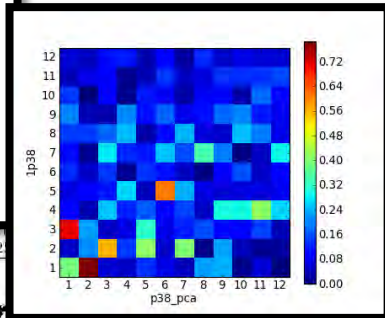
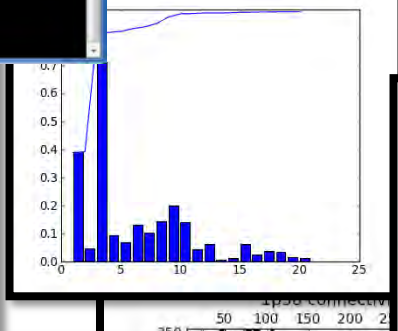
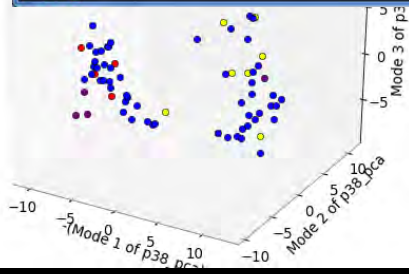
In [4]: ann.se
ann.secondary          ann.set_logger
ann.select_residues    ann.set_res_str_assignments
ann.set_coordinates    ann.set_workDir
ann.set_hessian

In [4]: ann.select_residues(**)
Which chains and residues do you want to use from lmkp:
Chain A length: 144 (Residue ids range from 204 to 347)
You have entered: -
Selection result:
144 residues from chain A

In [5]: ann.perform_analysis ()
?> Hessian matrix is being calculated.
?> Parameter: cutoff = 15
?> Parameter: gamma = 1
?> Hessian is calculated in 0.67s.
?> Normal mode calculation has started.
?> 20 modes will be calculated.
?> Normal modes are calculated in 0.12s.

In [6]:

```





# ProDy

Protein Dynamics & Sequence Analysis

## Python & *ProDy* Demo

- Python
  - numbers, strings, lists, dictionaries, slicing, indexing
  - files, loops, exception control
- NumPy
  - Arrays and efficiency
- Matplotlib
  - Plotting, showing an image
- ProDy
  - Parsers
  - AtomGroups
    - Iterations, hierarchical view, slicing, indexing
  - Plotting functions
    - Show protein
  - Ensemble Analysis



# ProDy

Protein Dynamics & Sequence Analysis

## What's next?



### ProDy

Learn how to use ProDy from the introductory ProDy tutorial or from the comprehensive API reference manual.

[Tutorial](#)

[Manual](#)



### Normal Mode Wizard

Learn how to depict normal modes and generate animations of protein dynamics along them with NMWiz.

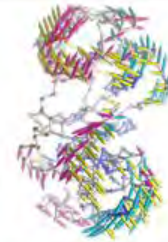
[Go to Tutorial](#)



### Evol Tutorial

Learn how to identify conserved and coevolved residues and characterizing their dynamical properties.

[Go to Tutorial](#)



### Elastic Network Models

Learn how to perform normal mode analysis and developing customized force constant functions.

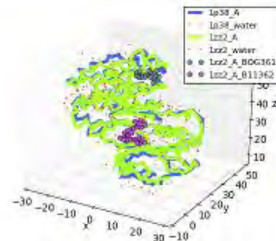
[Go to Tutorial](#)



### Ensemble Analysis

Learn how to analyze large and heterogeneous ensembles of protein structures to infer dynamical properties.

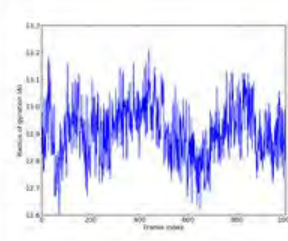
[Go to Tutorial](#)



### Structure Analysis

Learn how to compare and align structures, identify ligand contacts, and extract ligands from PDB files.

[Go to Tutorial](#)



### Trajectory Analysis

Learn how to analyze simulation trajectories, in particular handling large trajectory files that don't fit in memory.

[Go to Tutorial](#)



### Conformational Sampling

Learn how to generate alternate protein conformations along ANM modes and to refine them using NAMM.

[Go to Tutorial](#)



# ProDy

Protein Dynamics & Sequence Analysis

## Online Documentation

- Prefer the online HTML documentation
  - Frequently revised and improved
  - Easy to copy and past code into a Python shell

```
In [15]: showProtein(p38);
```

In [15]


```
showProtein(p38);
```

In [15]

- Show code in a page for copying

### Toolbox

Show code snippets for copying:

 Show code and output

Enter search terms or a module, class or function name:

### Code Snippets

```
# This code was copied from ProDy documentation.
# Title: ProDy Basics - ProDy
# URL: http://www.csb.pitt.edu/ProDy/tutorials/prody_tutorial/basics.html

from prody import *
from pylab import *
ion()

parse<TAB>

p38 = parsePDB('1p38') # returns an AtomGroup object
p38 # typing in variable name will give some information

p38.num<TAB>

p38.numAtoms()
p38.numCoordsets() # returns number of models
p38.numResidues() # water molecules also count as residues
```

Select Code Close