



blog.tice.de

PHY432 next year

Computational Methods in Physics

PHY 494



Oliver Beckstein, DPhil

Associate Professor

Department of Physics

<https://becksteinlab.physics.asu.edu>



oliver.beckstein@asu.edu

(include “**PHY 494**” in the subject!!)

room: PSF 348

office hours: Tu 10:00am–11:30am
(or by appointment)

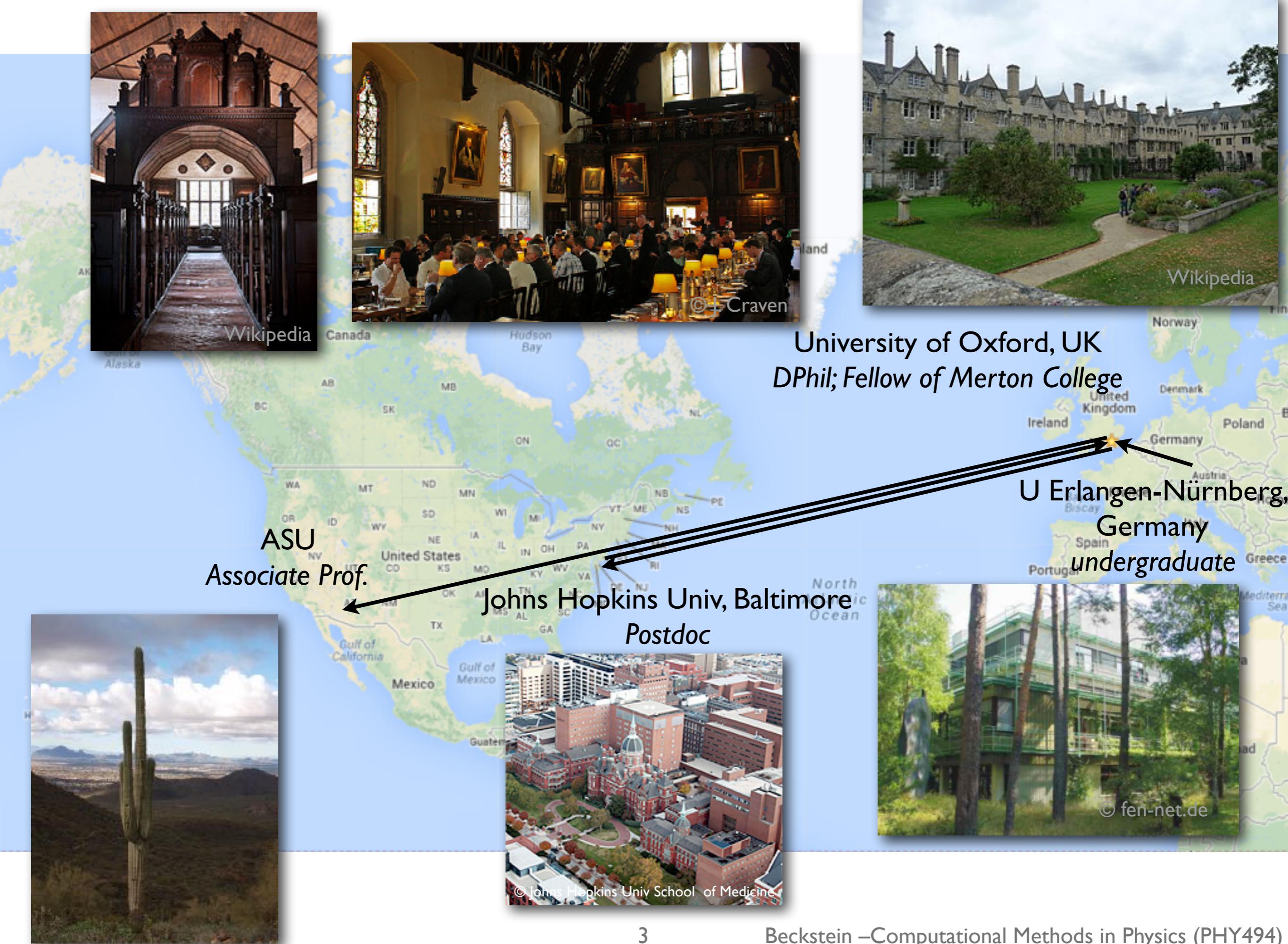
TA:

Chenou Zhang

czhan178@asu.edu

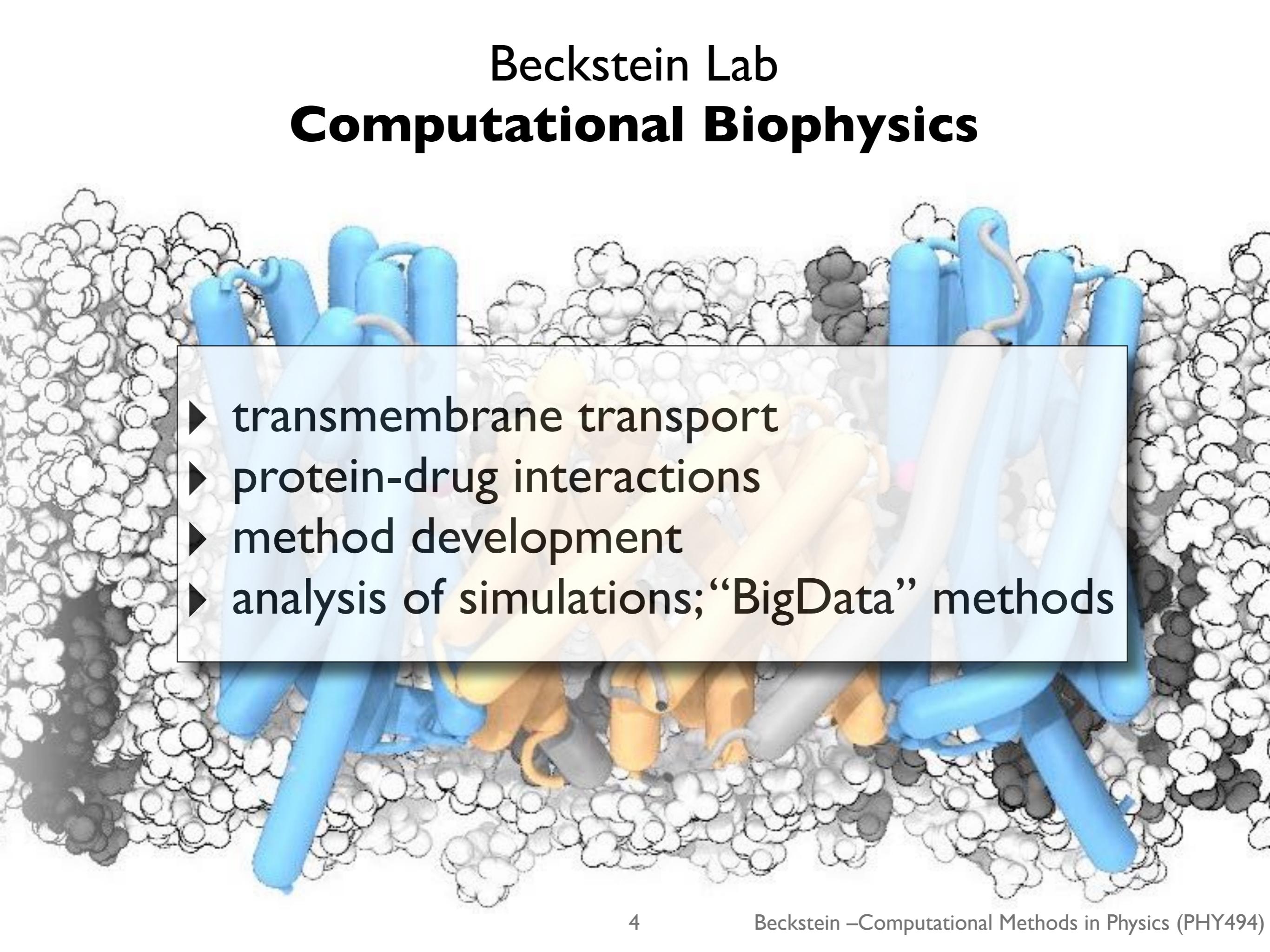
office: PSF 186

office hours: M/W 11am–12 noon



Beckstein Lab

Computational Biophysics

- 
- ▶ transmembrane transport
 - ▶ protein-drug interactions
 - ▶ method development
 - ▶ analysis of simulations; “BigData” methods

science

Theory

Experiment

Computation/
Simulation

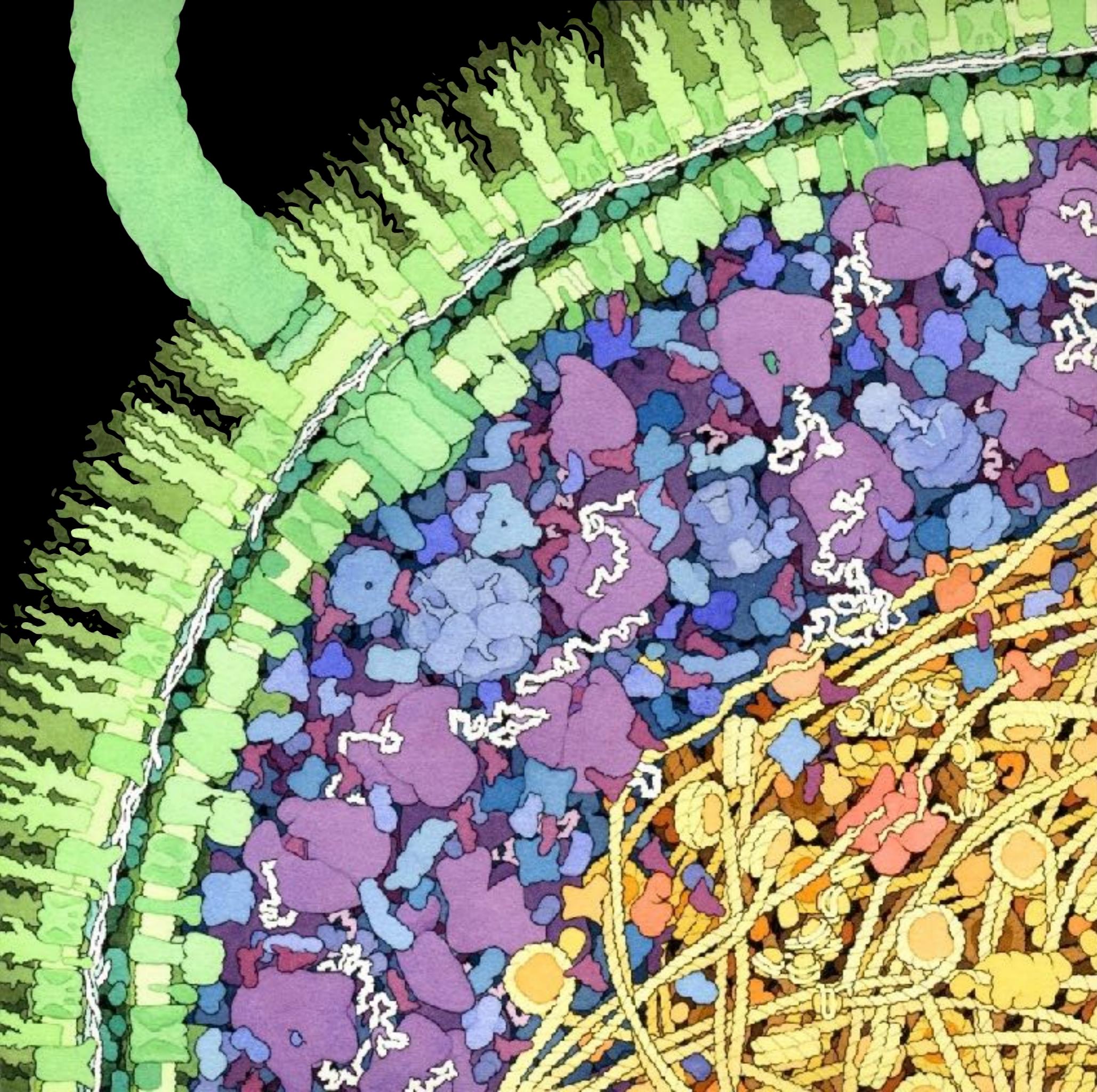
nature

LIFE

cells

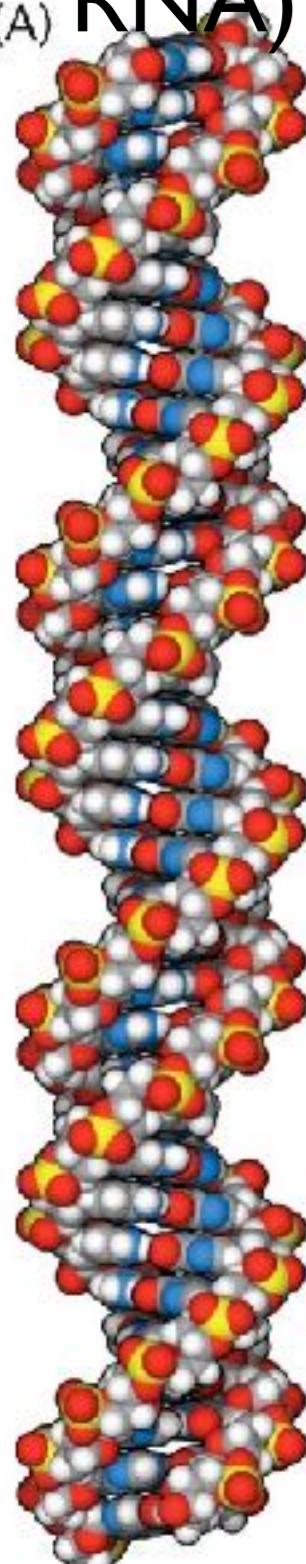
molecules

atoms



Escherichia coli (© 1999 David S. Goodsell, the Scripps Research Institute)

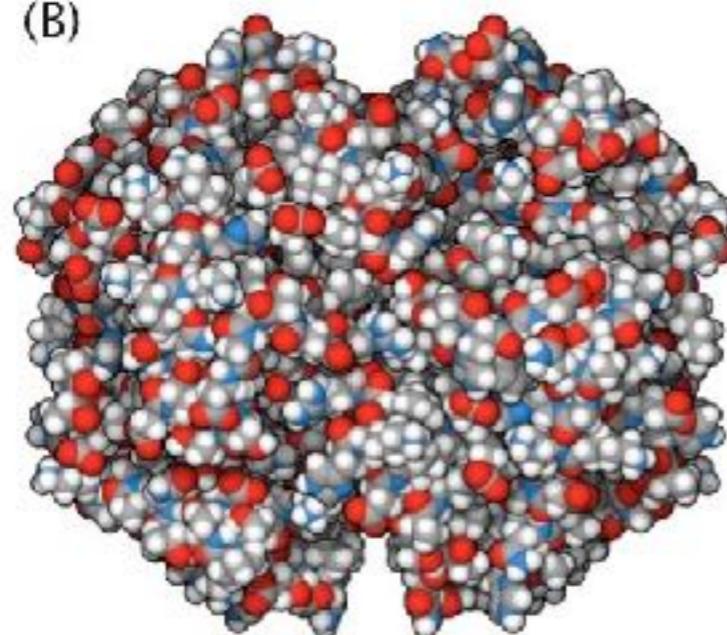
nucleic
acids (DNA,
RNA)



Biomolecules

proteins

(B)

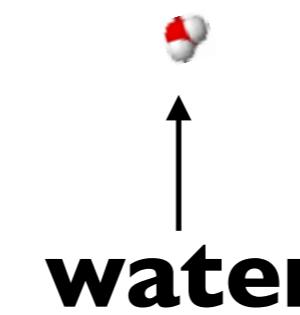


protein
representations

(C)

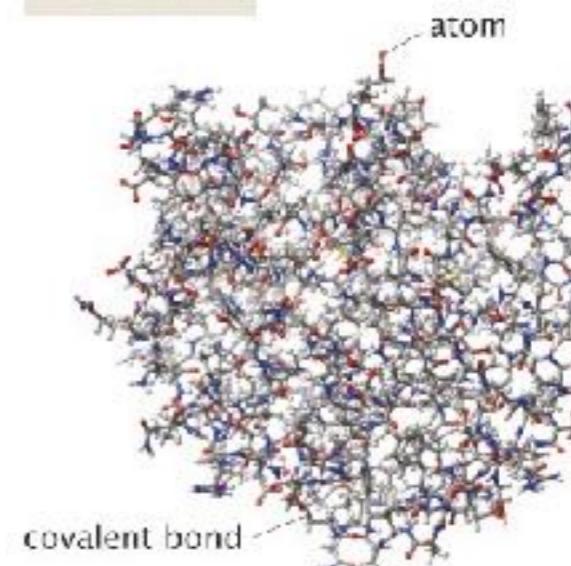


lipids

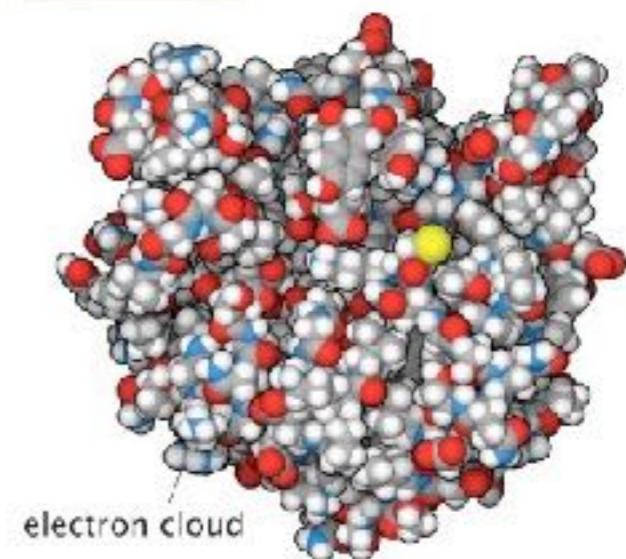


water

ball and stick



space-filling



ribbon

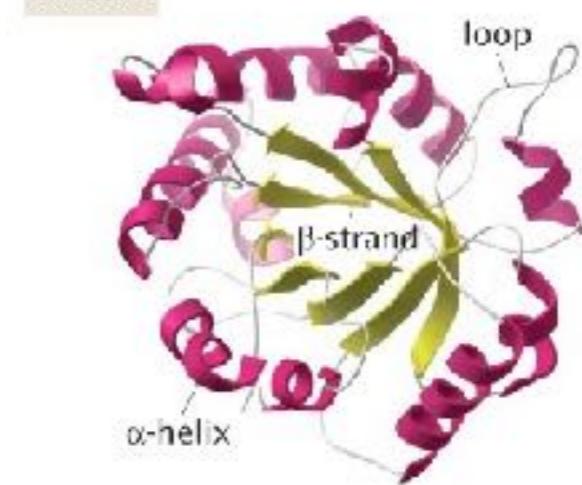
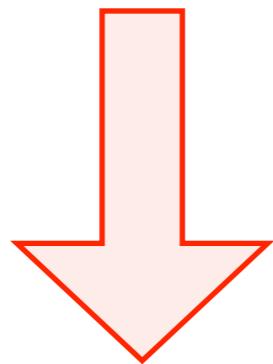


Figure 1.1 Physical Biology of the Cell, 2ed. (© Garland Science 2013)

Figure 2.32 Physical Biology of the Cell, 2ed.

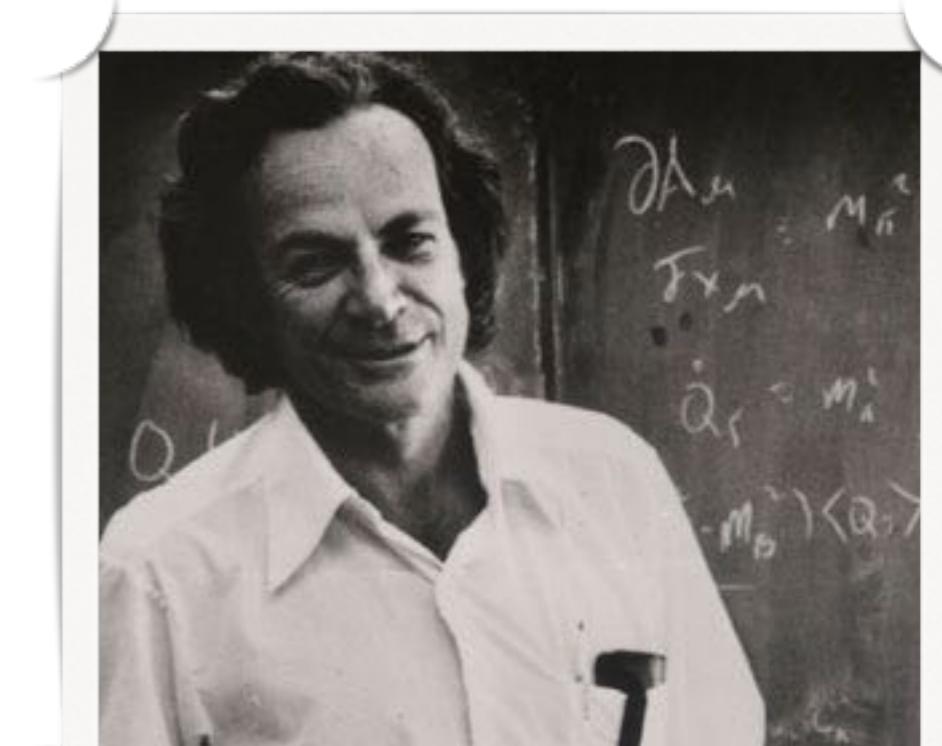
*“Everything that living things do can be understood
in terms of the jiggling and wiggling of atoms.”* —

Richard Feynman*



$$(\mathbf{r}_1(t), \dots, \mathbf{r}_N(t))$$

positions of all N atoms over time



Richard P. Feynman

Wikimedia Commons

Molecular Dynamics Computer Simulations



Molecular Dynamics (MD) Simulations (classical)

$$U(\mathbf{r}_1, \dots, \mathbf{r}_N) = U_{\text{bonded}}(\mathbf{r}_1, \dots) + U_{\text{non-bonded}}(\mathbf{r}_1, \dots, \mathbf{r}_N)$$

energy func

$$\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{r}_i} U(\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_N)$$

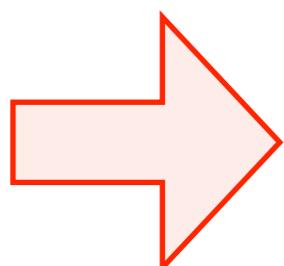
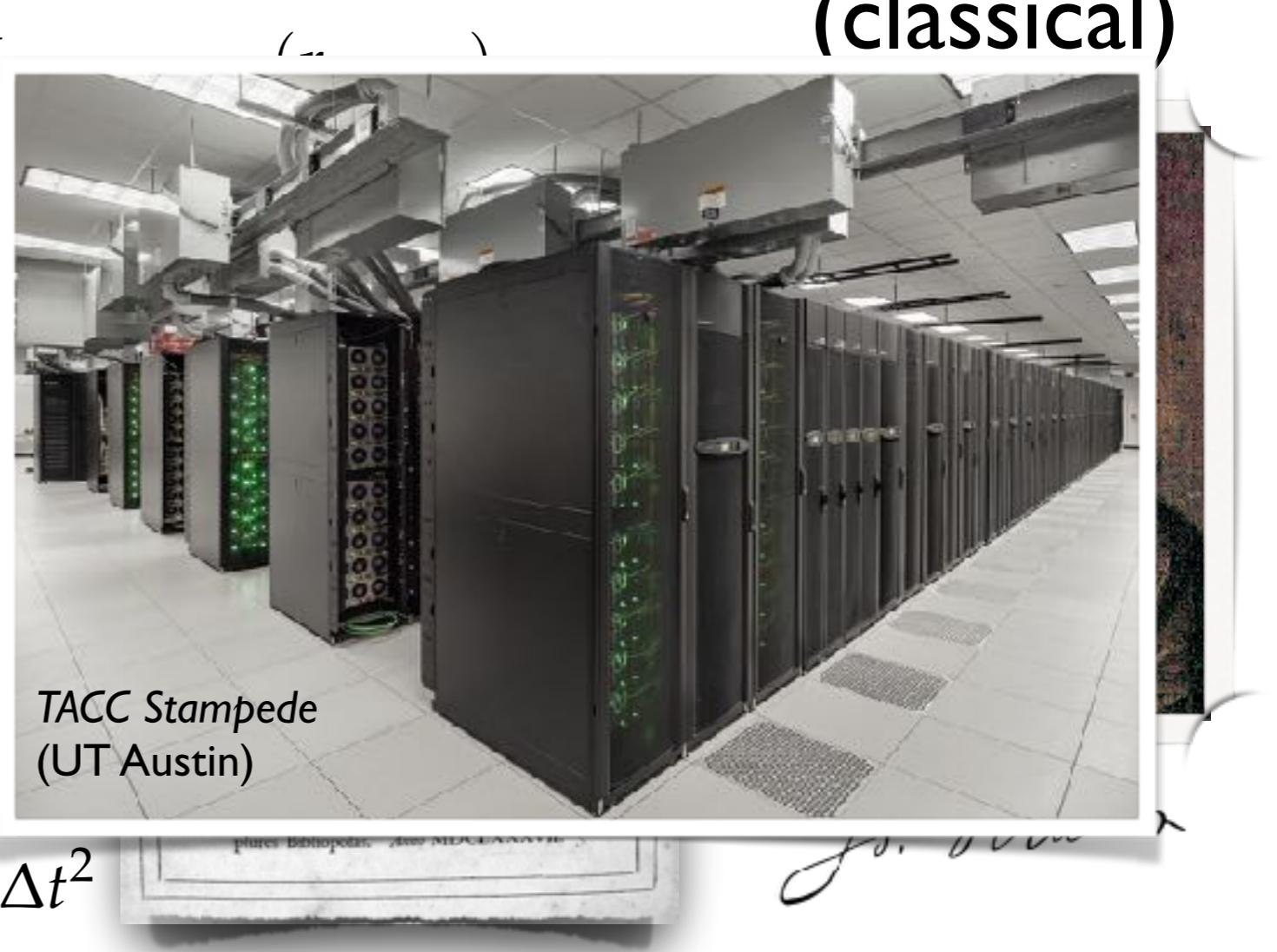
$$\mathbf{F}_i = m_i \mathbf{a}_i$$

Newton's
2nd law

$$\frac{d^2 \mathbf{r}_i}{dt^2} = \frac{\mathbf{F}_i}{m_i}$$

integrator

$$\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{\mathbf{F}_i}{m_i} \Delta t^2$$



- $\mathbf{r}_1(0), \dots, \mathbf{r}_N(0)$
- $\mathbf{r}_1(\Delta t), \dots, \mathbf{r}_N(\Delta t)$
- $\mathbf{r}_1(2\Delta t), \dots, \mathbf{r}_N(2\Delta t)$
- $\mathbf{r}_1(3\Delta t), \dots, \mathbf{r}_N(3\Delta t)$

⋮

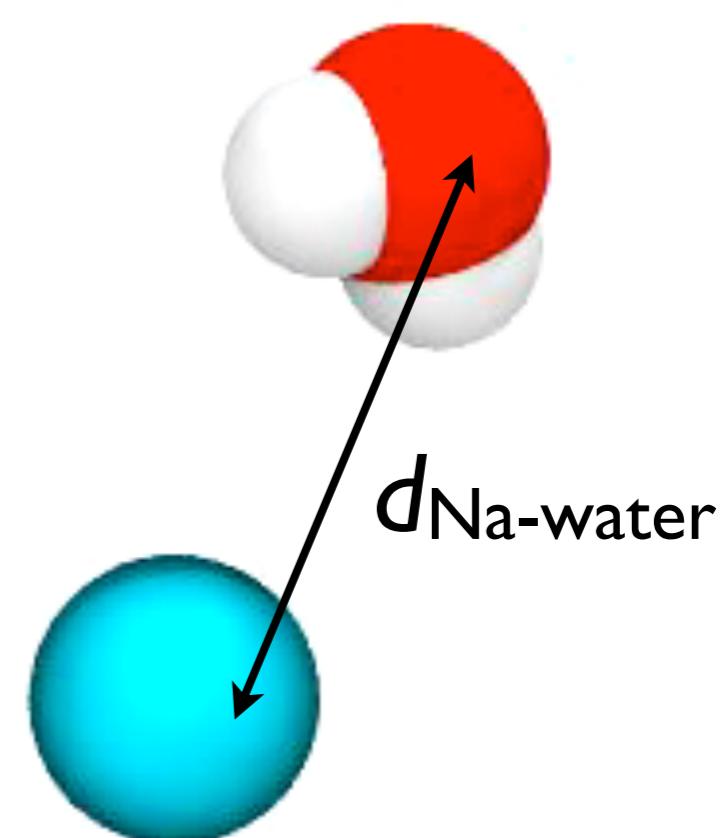
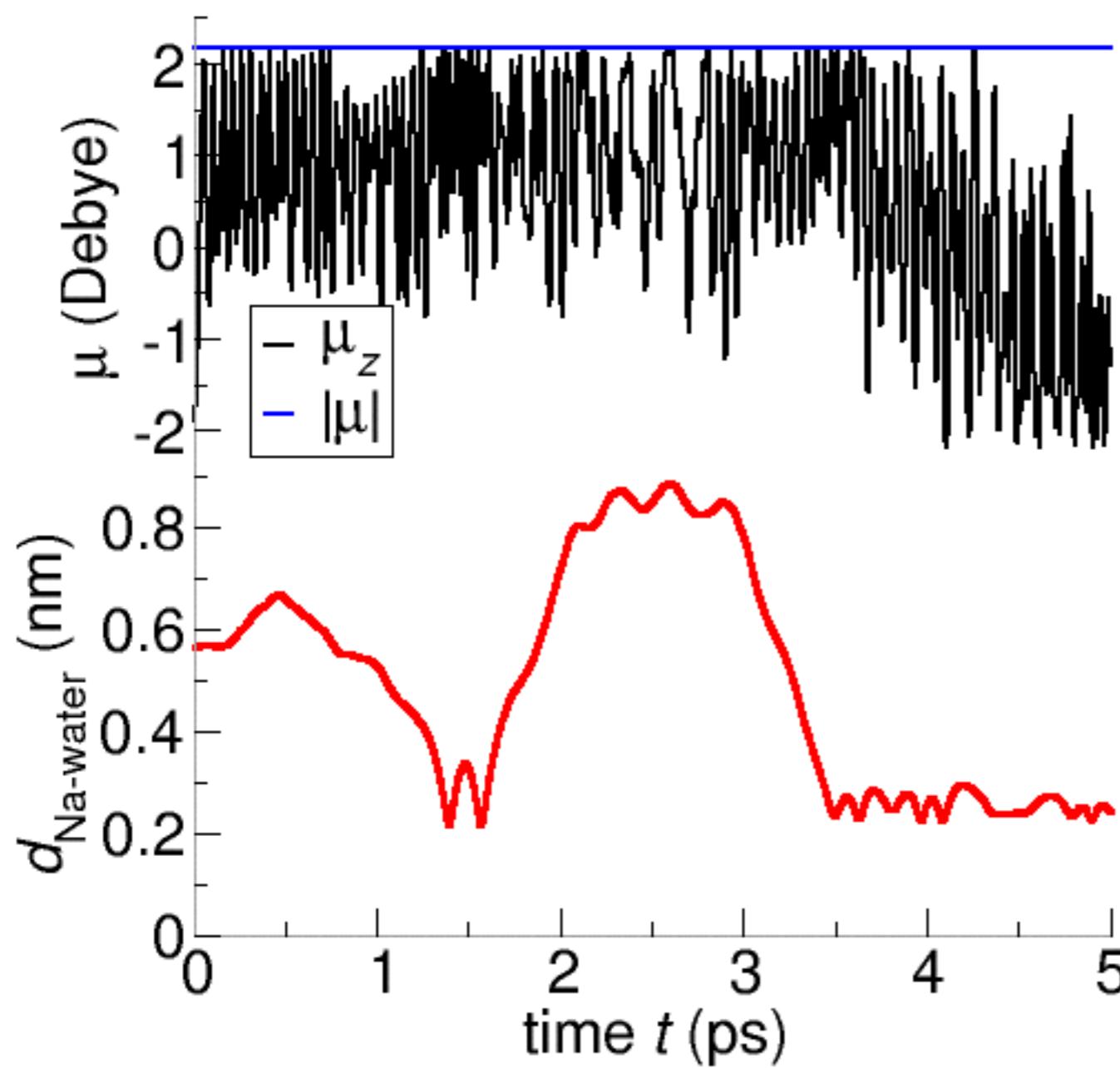
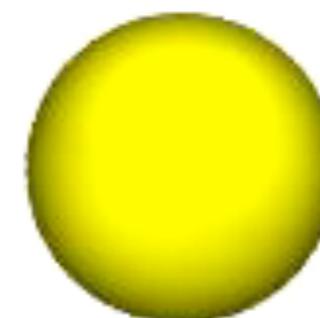
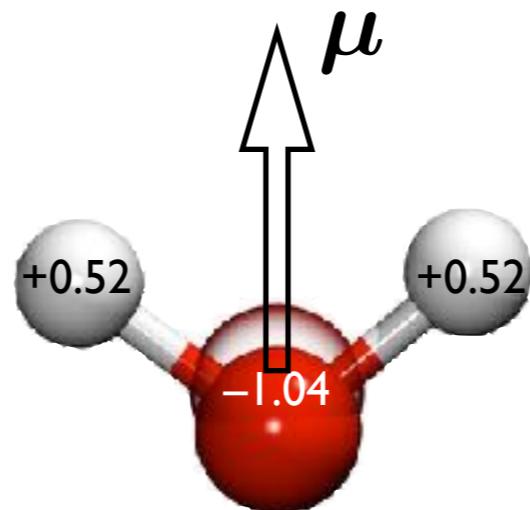
trajectory

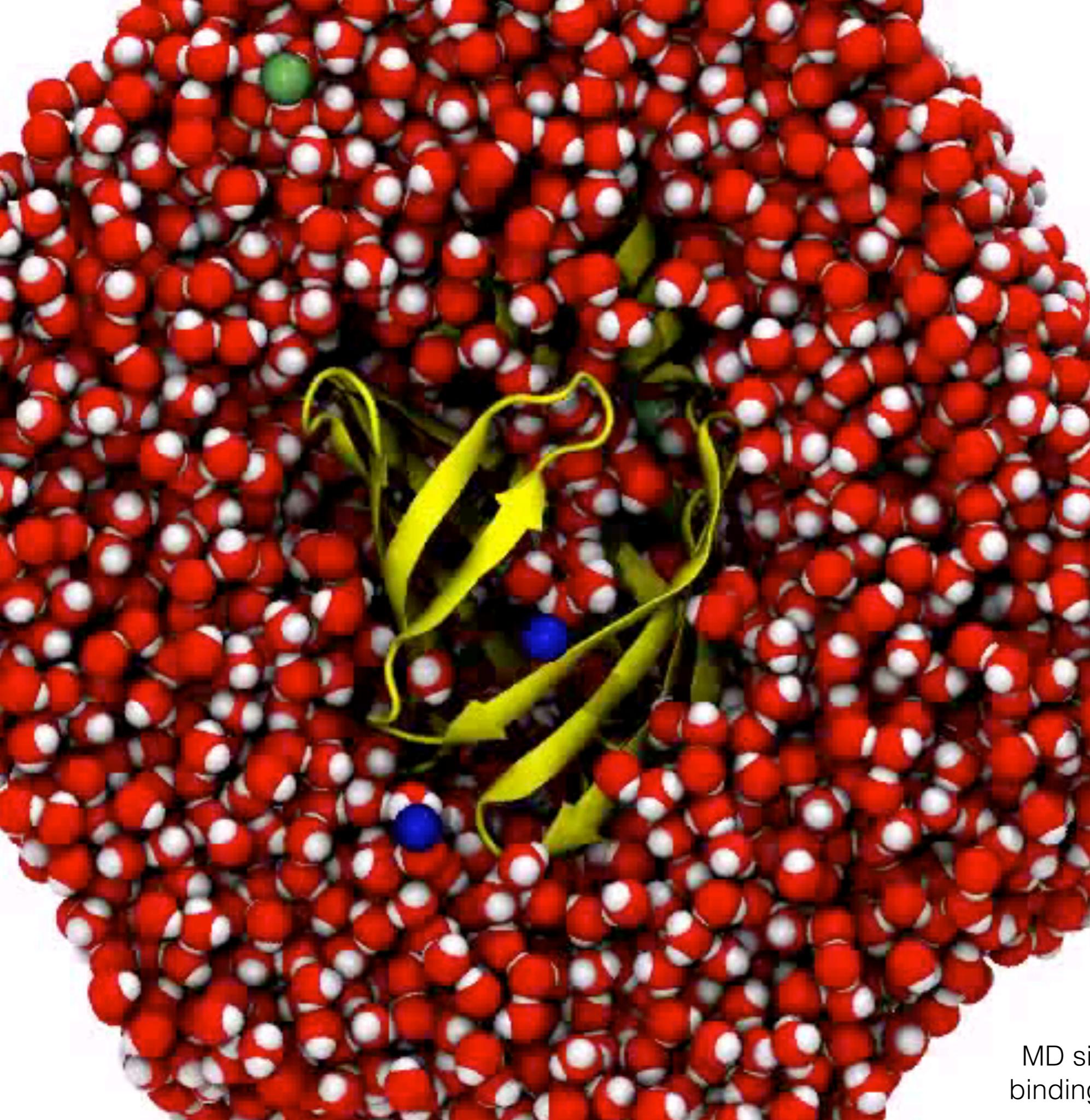
$$(\mathbf{r}_1(t), \dots, \mathbf{r}_N(t))$$

$$0 \leq t \leq \tau$$

$$\mu = q_H(\mathbf{r}_{HW_1} + \mathbf{r}_{HW_2} - 2\mathbf{r}_{MW})$$

$$d_{\text{Na-water}} = |\mathbf{r}_{\text{Na}} - \mathbf{r}_{\text{water}}|$$

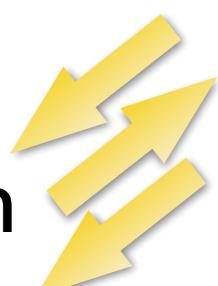




trajectory

$$(\mathbf{r}_1(t), \dots, \mathbf{r}_N(t))$$

$$0 \leq t \leq \tau$$

1 protein 

12 ions 

3432 waters 

2113 atoms

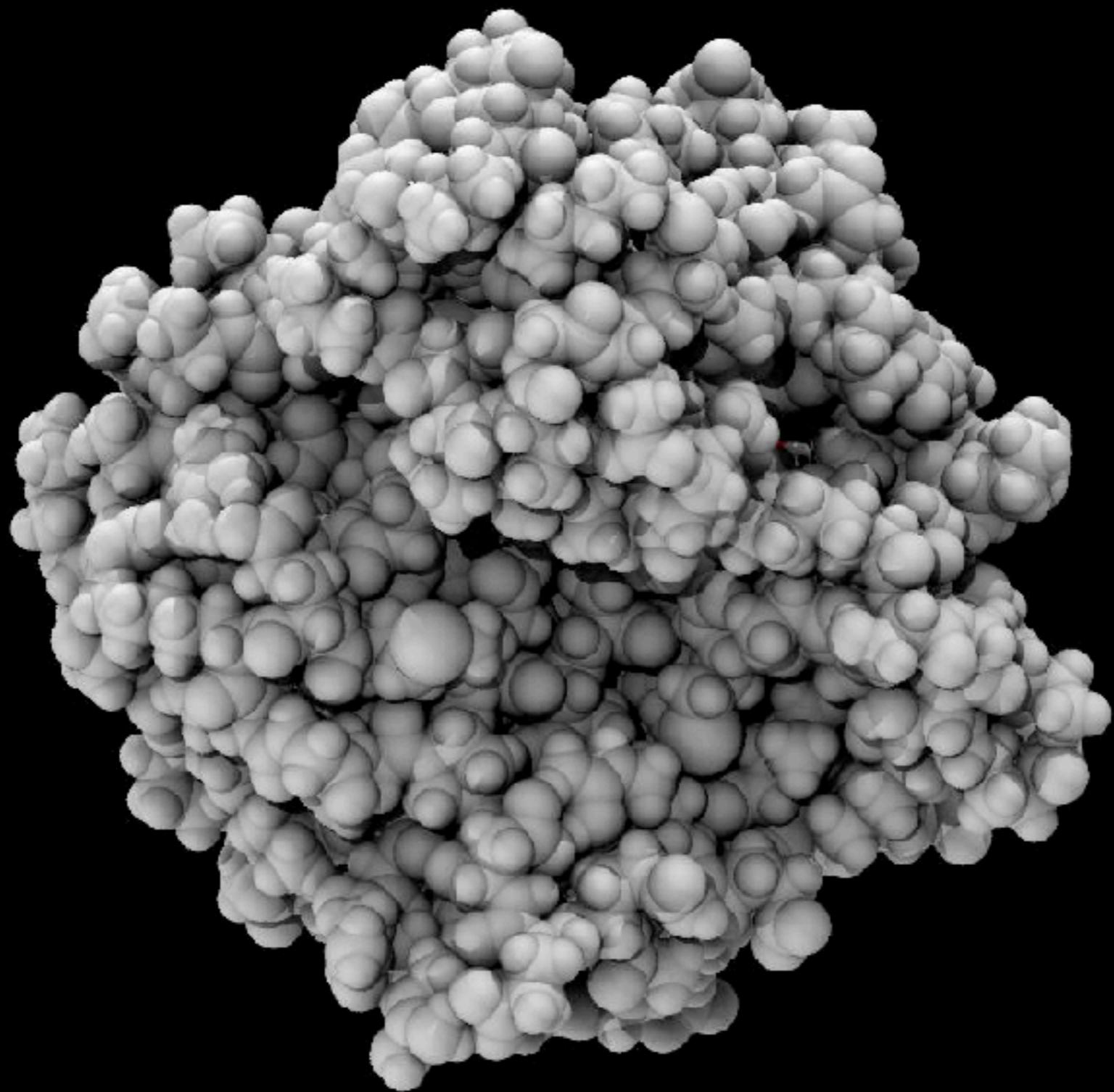
12 atoms

10296 atoms

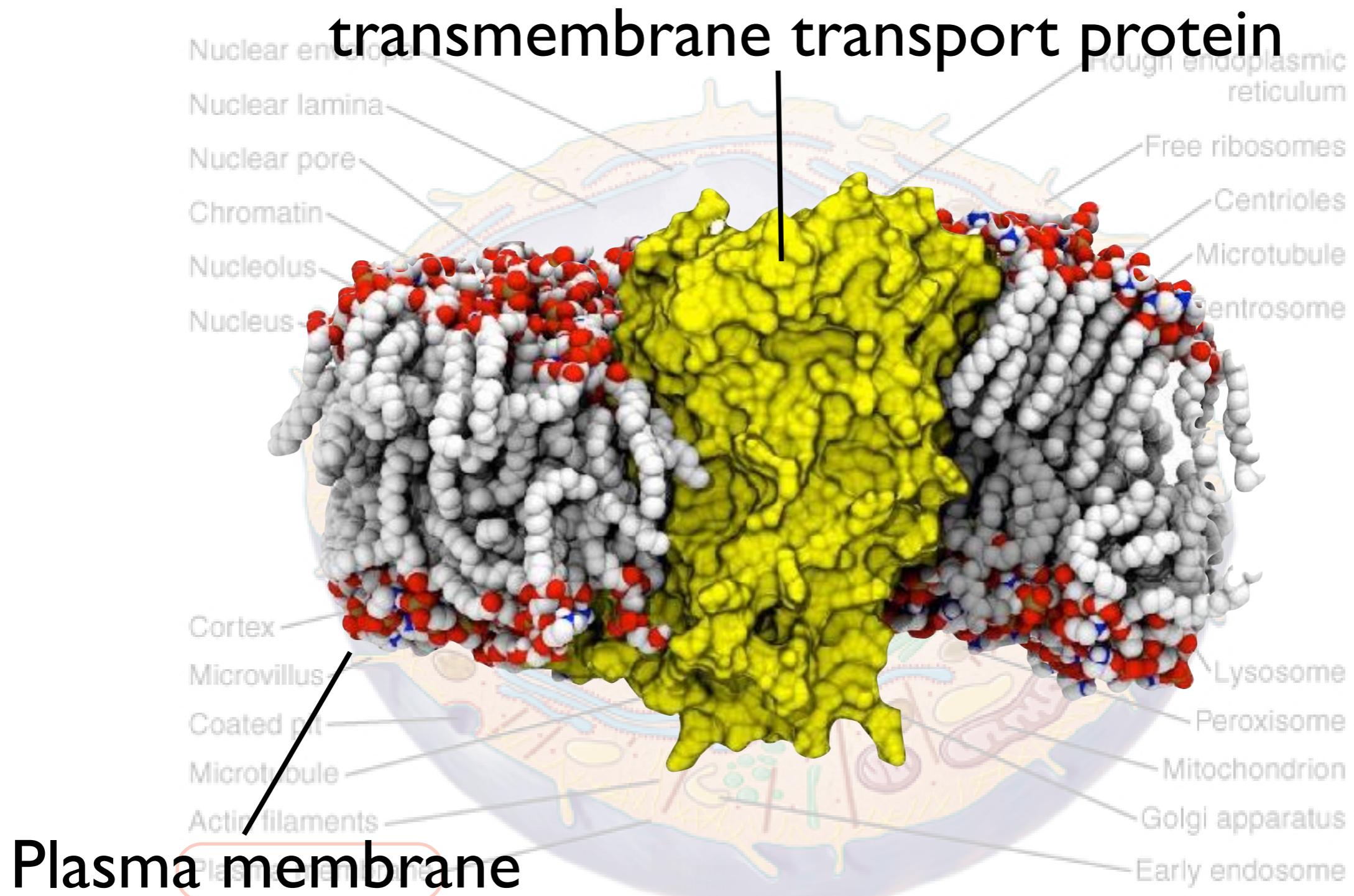
12421 atoms

MD simulation of intestinal fatty acid binding protein. Rendered with VMD.

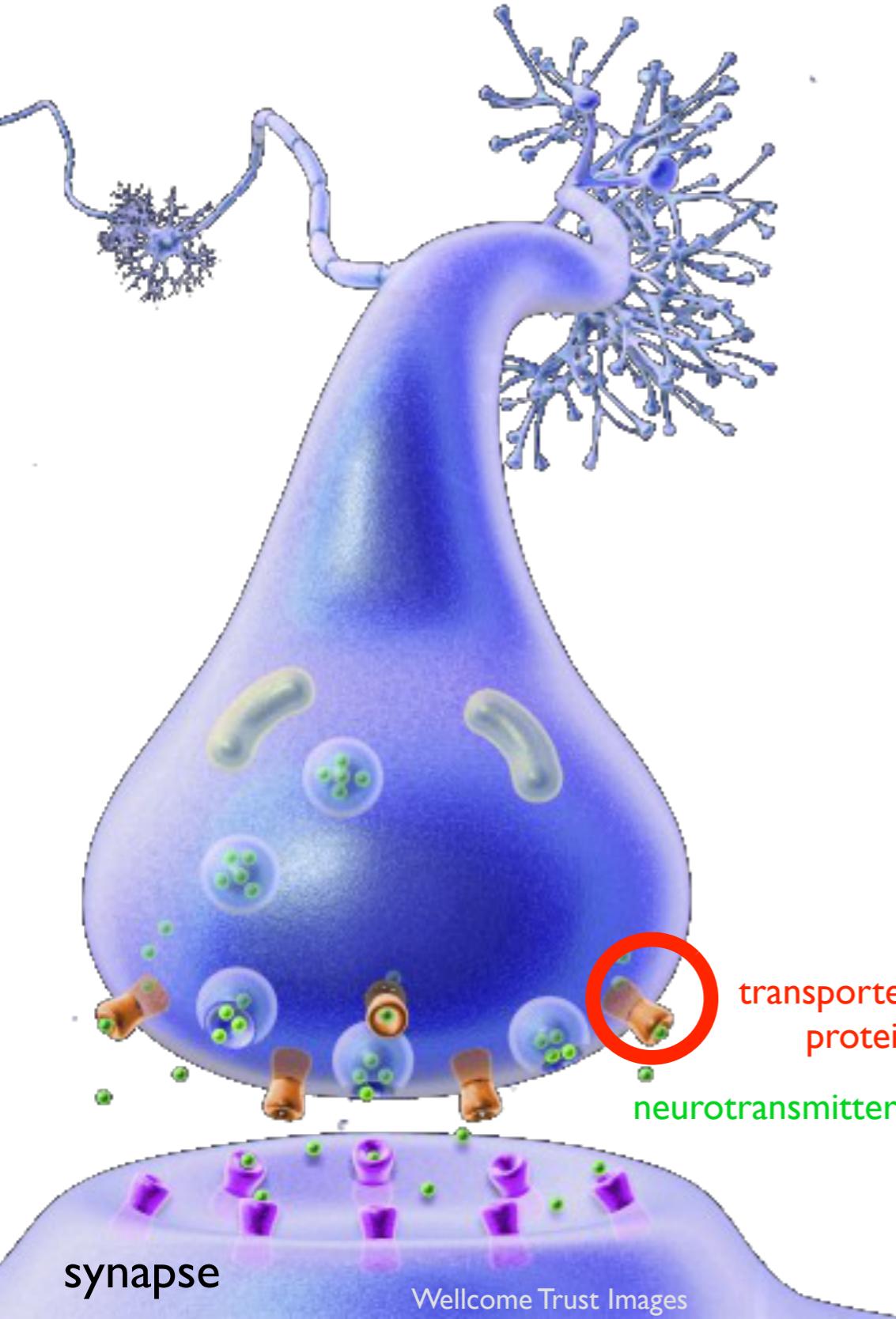
Enzyme dynamics: adenylate kinase



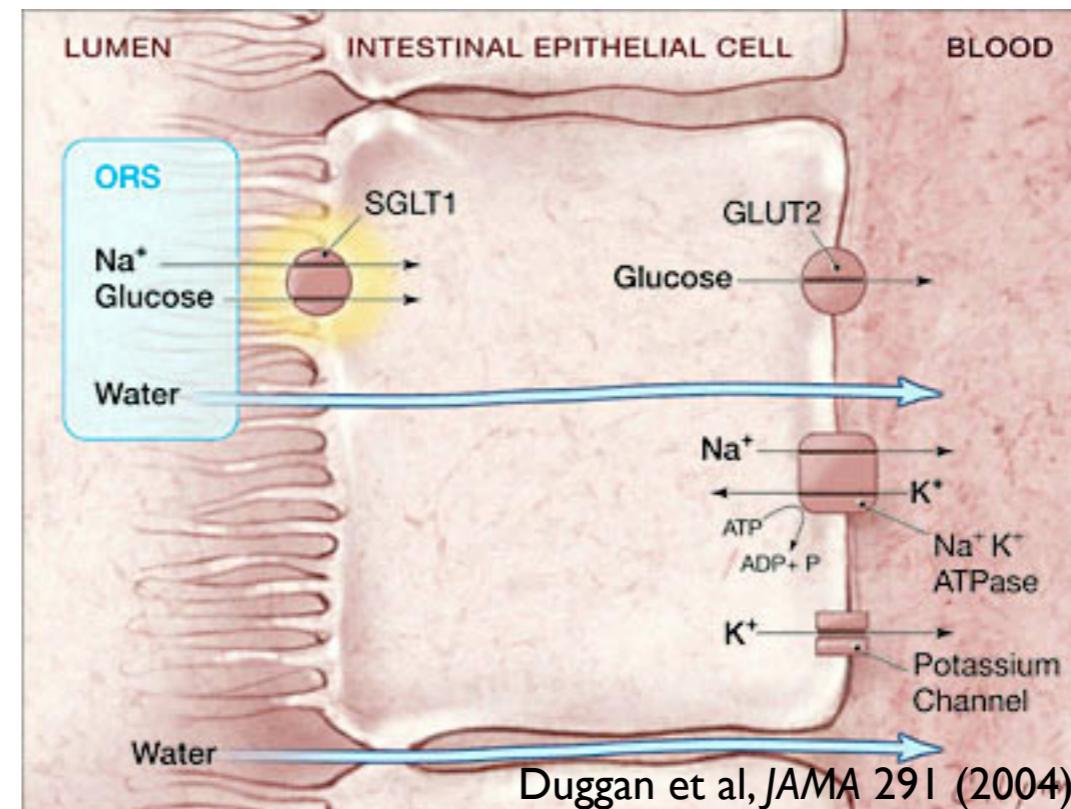
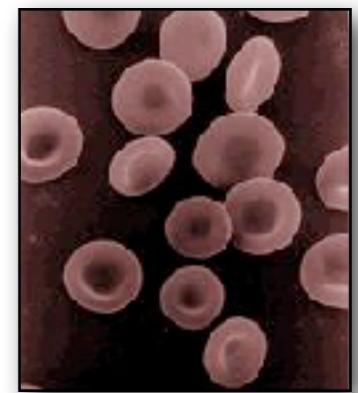
How do molecules enter and exit the cell?

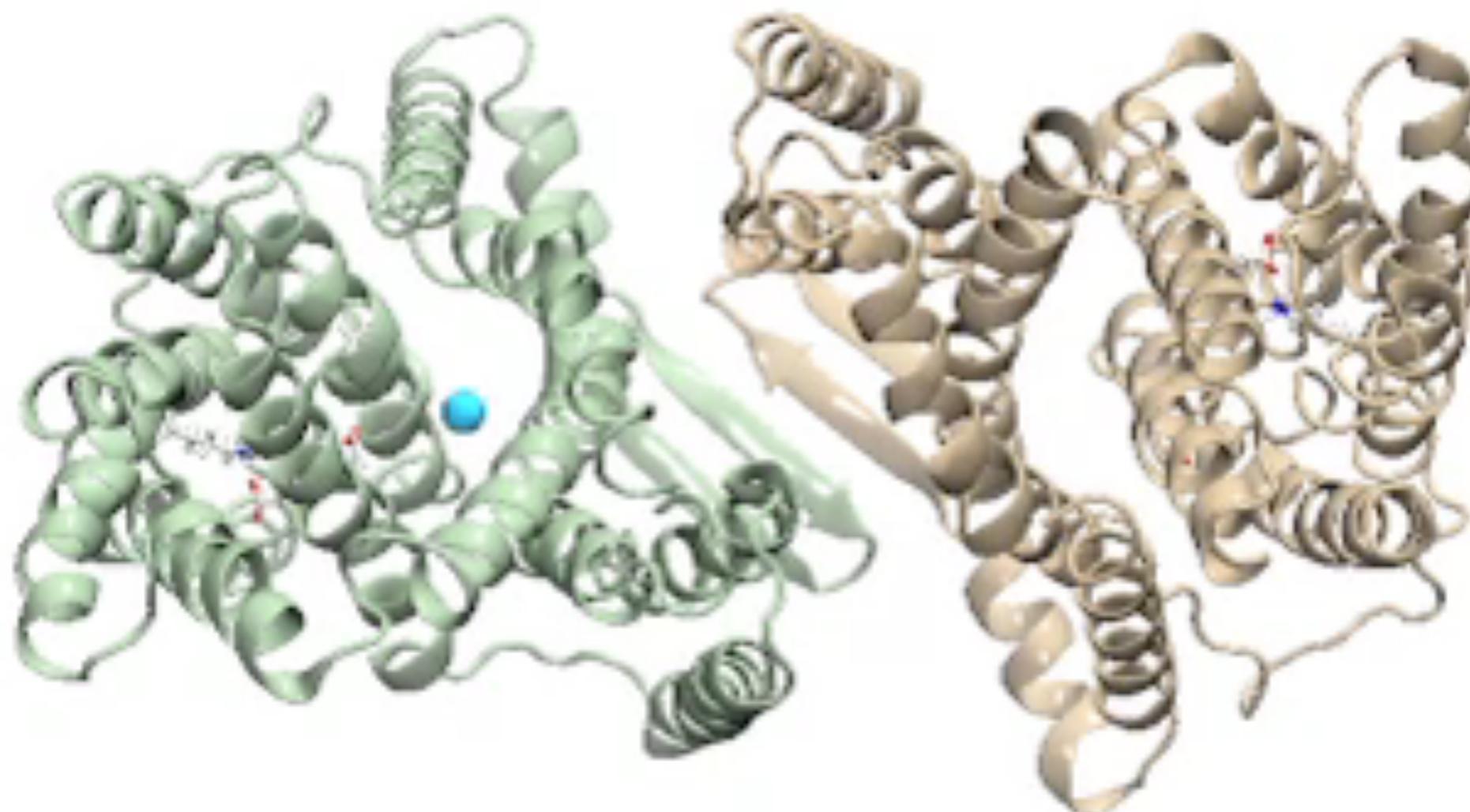


Transmembrane Transport



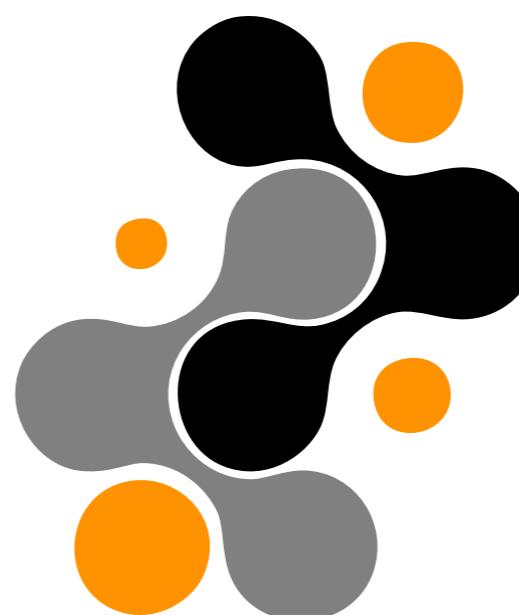
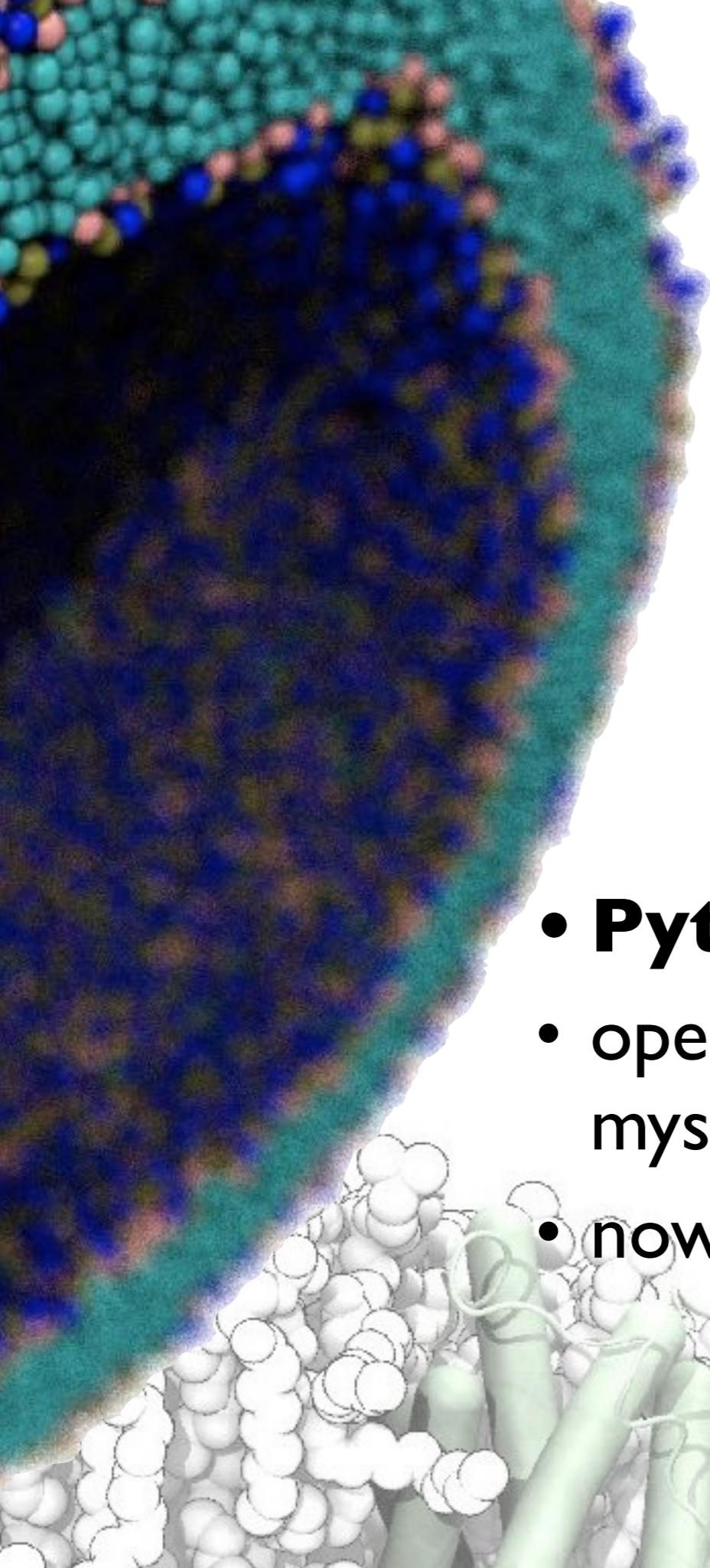
- synapse (neurotransmitter re-uptake)
- intestine (sugar, amino acid uptake)
- blood cells, kidney (pH)
- everywhere





M. Coincon, P. Uzdavins, E. Nji, D. L. Dotson, I. Winkelmann, S. Abdul-Hussein, A. D. Cameron, O. Beckstein, and D. Drew. **Crystal structures reveal the molecular basis of ion-translocation in sodium proton antiporters.** *Nature Struct. Mol. Biol.*, 23(3):248–255, 2016.
Y. Huang, W. Chen, D. L. Dotson, O. Beckstein, and J. Shen. **Mechanism of pH-dependent activation of the sodium-proton antiporter NhaA.** *Nature Communications*, 7:12940, 10 2016.
C. Lee, S. Yashiro, D. L. Dotson, P. Uzdavins, S. Iwata, M. S. P. Sansom, C. von Ballmoos, O. Beckstein, D. Drew, and A. D. Cameron. **Crystal structure of the sodium-proton antiporter NhaA dimer and new mechanistic insights.** *J Gen Physiol*, 144(6):529–544, 2014.
C. Lee, H. J. Kang, C. von Ballmoos, S. Newstead, P. Uzdavins, D. L. Dotson, S. Iwata, O. Beckstein, A. D. Cameron, and D. Drew. **A two-domain elevator mechanism for sodium/proton antiport.** *Nature*, 501 (7468):573–577, 09 2013.

NhaA proton-sodium transporter



MD ANALYSIS

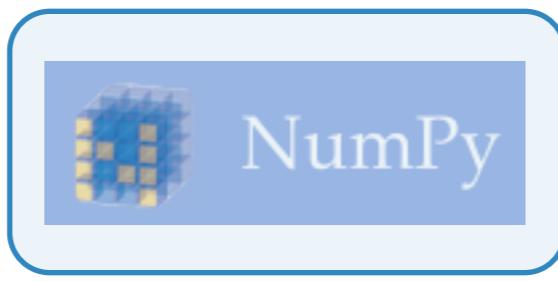
<https://mdanalysis.org>

- **Python** library for analysis of MD simulations
 - open sourced in 2007 by two grad students and myself (at Johns Hopkins U)
 - now: medium-size active open source project
- 

simulation trajectory

dcd, xtc, trr,
ncdf, trj, pdb,
pqr, gro, crd,
dms, trz, mol2,
xyz, config,
history, gms, ...

psf, tpr,
prmtop, dms,
mol2, hoomd
xml, ...



python™

“accessible”
structured
data

analysis
algorithm

processed
data

tables
images

graphs

C_α RMSF

RESULTS!

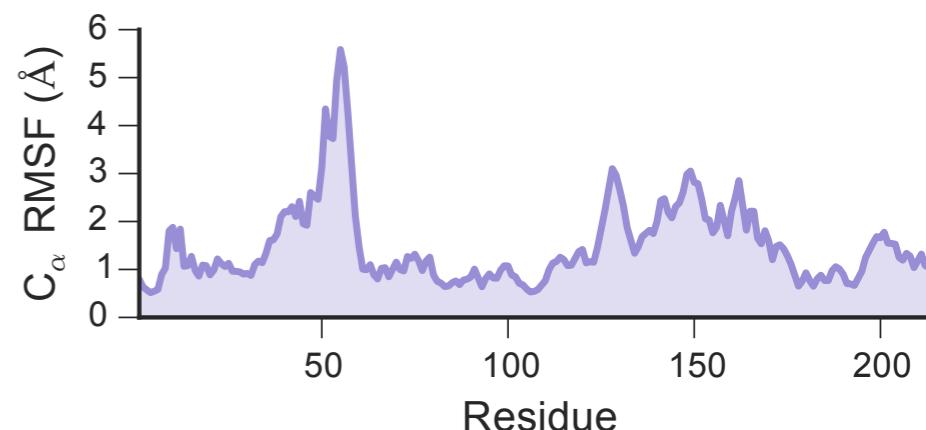
```
import numpy as np
import MDAnalysis as mda

u = mda.Universe("topol.tpr", "trj.xtc")
ca = u.select_atoms("name CA")

means = np.zeros((len(ca), 3))
sumsq = np.zeros_like(means)

for k, ts in enumerate(u.trajectory):
    sumsq += (k/(k+1.0)) * (ca.positions - means)**2
    means[:] = (k*means + ca.positions)/(k+1.0)
rmsf = np.sqrt(sumsq.sum(axis=1)/(k+1.0))

matplotlib.pyplot.plot(ca.residues.resids, rmsf)
```



Naveen Michaud-Agrawal, Elizabeth J. Denning, Christian Beckstein (logo), Joshua L. Adelman, Shobhit Agarwal, Irfan Alibay, Balasubramanian, Utkarsh Bansal, Jonathan Barnoud, Tone Bengtsen, Alejandro Bernardin, Mateusz Bieniek, Wouter Boomsma, Jose Borreguero, Bart Bruininks, Sébastien Buchoux, Sören von Bülow, David Caplan, Matthieu Chavent, Kathleen Clark, Ruggero Cortini, Davide Cruz, Robert Delgado, John Detlefs, Xavier Deupi, Jan Domanski, David L. Dotson, Shujie Fan, Lennard van der Feltz, Philip Fowler, Joseph Goose, Richard J. Gowers, Lukas Grossar, Abhinav Gupta, Akshay Gupta, Benjamin Hall, Eugen Hruska, Kyle J. Huston, Joe Jordan, Jon Kapla, Navya Khare, Andrew William King, Max Linke, Philip Loche, Jinju Lu, Micaela Matta, Andrew R. McCluskey, Robert McGibbon, Manuel Nuno Melo, Dominik 'Rathann' Mierzejewski, Henry Mull, Fiona B. Naughton, Alex Nesterenko, Hai Nguyen, Sang Young Noh, Nabarun Pal, Mattia F. Palermo, Danny Parton, Joshua L. Phillips, Kashish Punjani, Vedant Rathore, Tyler Reddy, Pedro Reis, Paul Rigor, Carlos Yanez S., Utkarsh Saxena, Sean L. Seyler, Paul Smith, Andy Somogyi, Caio S. Souza, Shantanu Srivastava, Lukas Stelzl, Gorman Stock, Ayush Suhane, Xiki Tempula, Matteo Tiberti, Isaac Virshup, Lily Wang, Nestor Wendt, Zhiyi Wu, Zhuyi Xue, Juan Eiros Zamora, Johannes Zeman, and Oliver Beckstein.

9 core developers

5 GSoC students

1 GSoD tech.writer

4 REU students

84 contributors



mdanalysis.org

github.com/MDAnalysis



Google

GSoC 2016–2019
GSoD 2019



>
-
PHY494

Introduction

- Lecture/Computer lab

Tuesday/Thursday 1:30–2:45pm, PSF 355

- use own laptop if possible

- Information on **Canvas**

- announcements

- syllabus

- literature

- homework submission

➡ ... check regularly!

- Information on **GitHub**
[https://github.com/ASU-
CompMethodsPhysics-
PHY494](https://github.com/ASU-CompMethodsPhysics-PHY494)
- code
- homework submission
- code submission

Textbooks

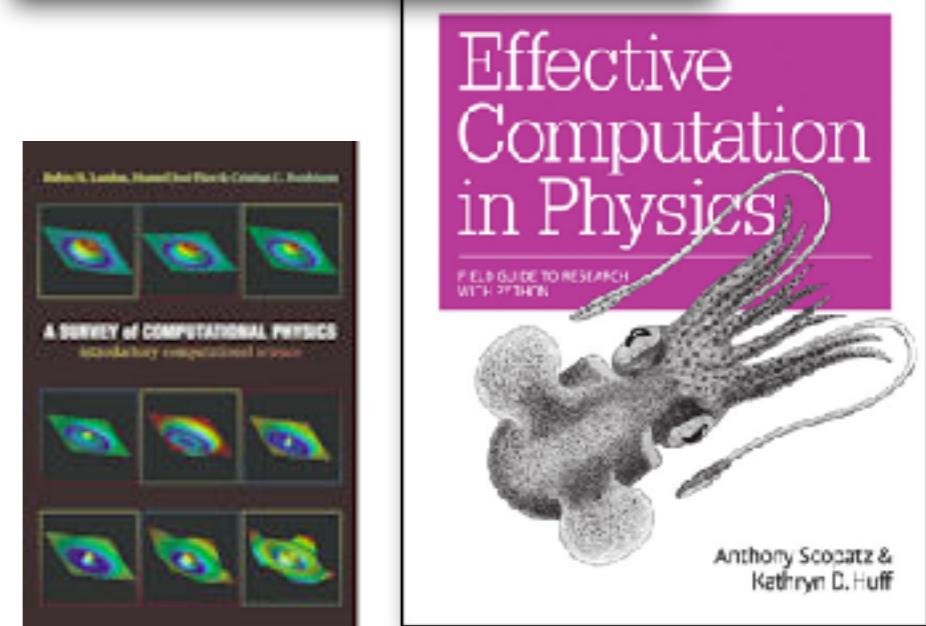
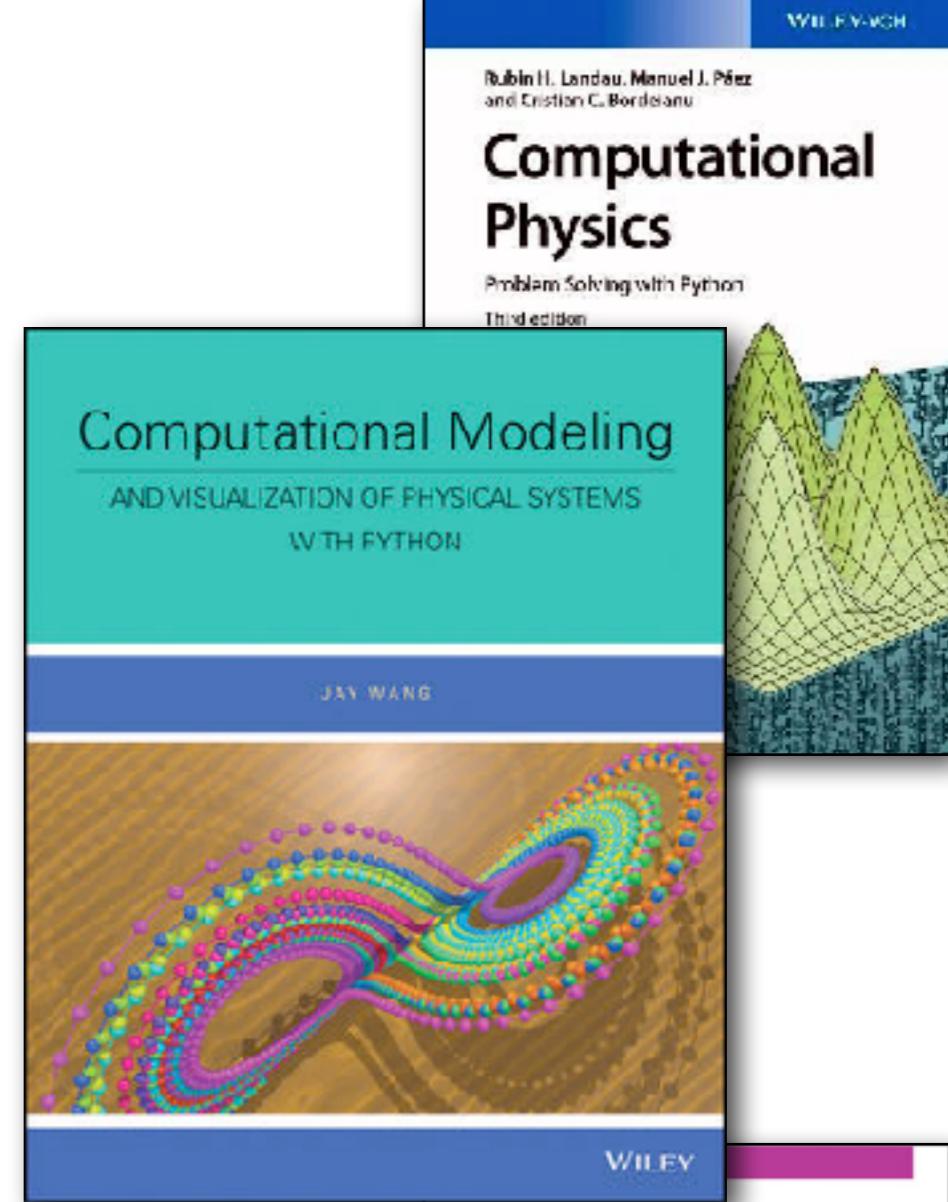
Computational Physics, R Landau et al, Wiley. 3rd edition (2015). (hard copy or eBook)

Computational Modeling and Visualization of Physical Systems with Python, J Wang, Wiley (2016)

Other books

Effective Computation in Physics, A Scopatz and KD Huff, O'Reilly (2015).

A Survey of Computational Physics. R Landau et al., Princeton U Press (2011). Free Online ComPADRE edition
<http://www.compadre.org/psrc/items/detail.cfm?ID=11578>



problem

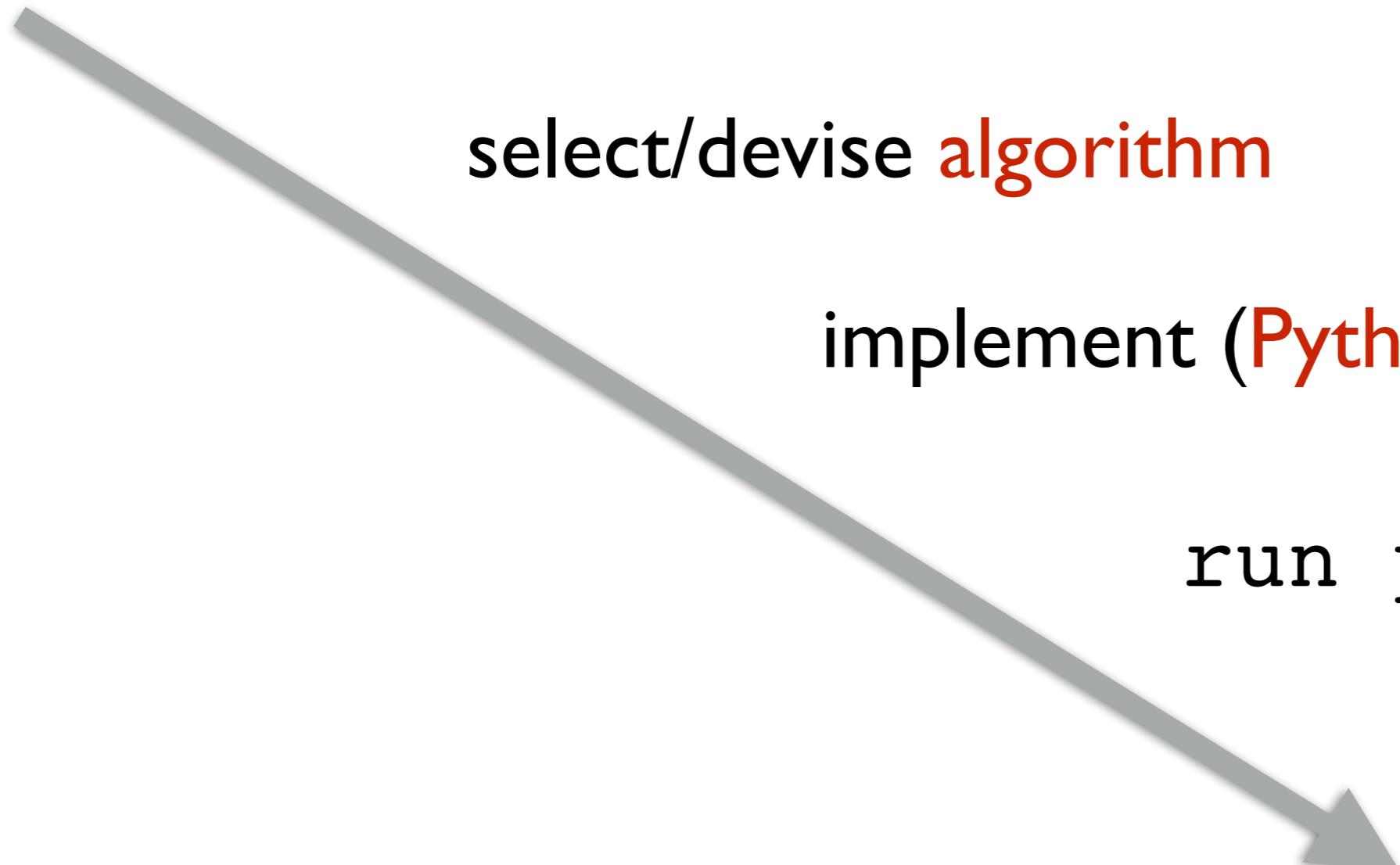
express quantitatively
(equations)

select/devise algorithm

implement (Python)

run program

analyze output



Course Content

Programming

Software engineering

Physics

Working in a scientific computing environment

- Unix shell
- version control with *git*
- programming in *Python*; use of numerical libraries; publication-quality plotting

Fundamentals of numerical approaches

- number representations and errors
- derivatives and integration
- linear algebra (matrix calculations, eigen problems)
- root-finding, minimization, data fitting
- solving ordinary differential equations (ODEs)
- partial differential equations (PDEs)
- Fourier analysis
- random numbers and Monte Carlo

Course Content

Applications to physical problems

(Not all topics will be included in the class but can be introduced through projects.)

- chaotic pendulum, non-linear dynamics
- molecular dynamics
- Ising model
- electrostatics (Laplace and Poisson equation)
- heat and wave equations
- quantum mechanics (Schrödinger equation)
- fluid dynamics
- realistic baseball throw

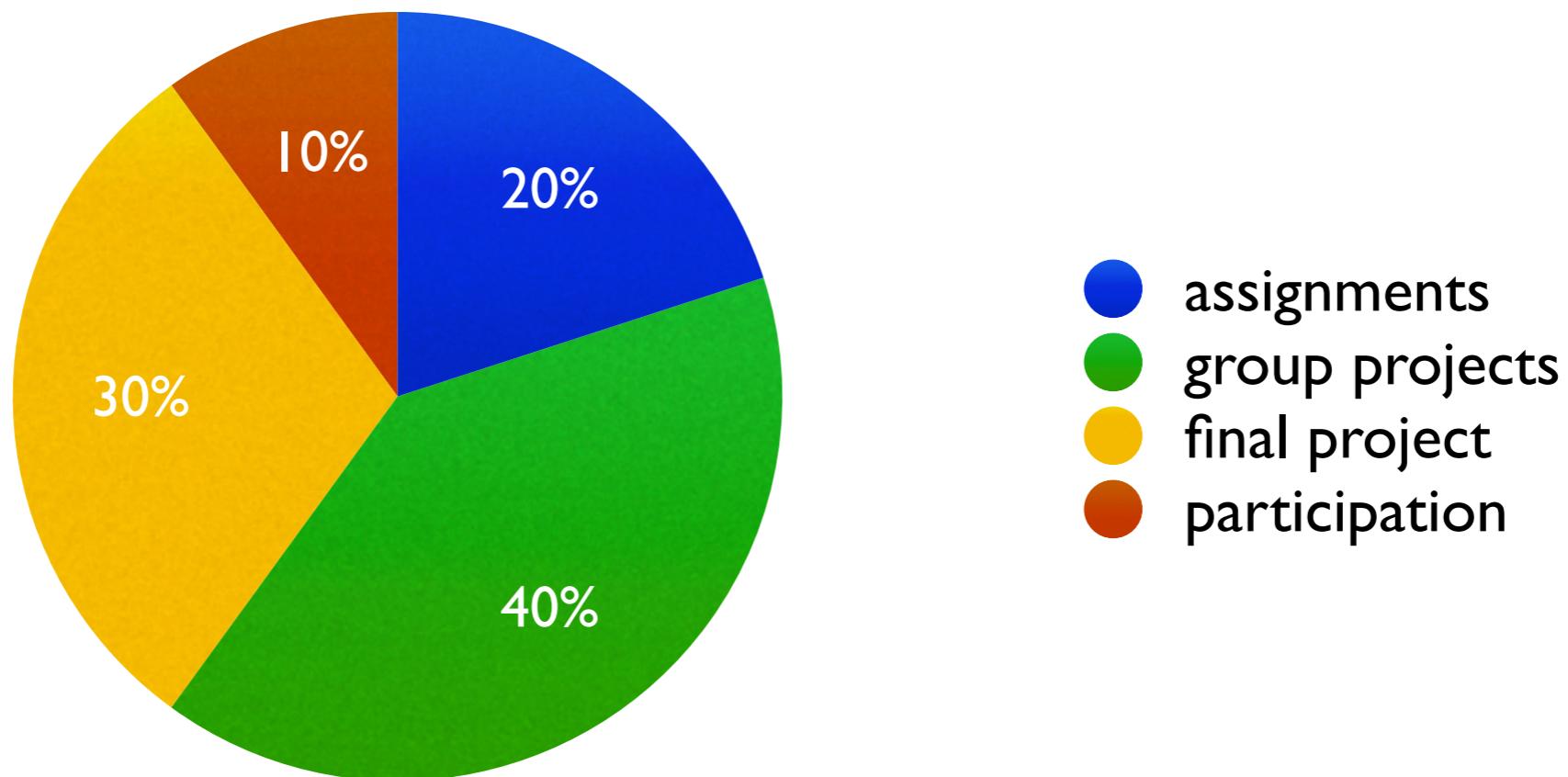
Assignments: homeworks (electronic submission via Canvas, code and output to GitHub repository)

Projects: (2), individual or groups (code developed together, reports written individually)

Final project: teams of two/three, code and poster developed together, individual Q&A at poster (during “Final Poster Session”)

Participation: attendance, etc

Grading



Grading scale*

A $\geq 90\%$

B $\geq 80\%$

C $\geq 70\%$

D $< 70\%$

\pm grades used (see Syllabus)
(A+, A, A-, ..., C+, C, D)

* may be adjusted downwards

Academic integrity

Present your own and others'
work honestly and fairly.

Academic dishonesty

- cheating
- plagiarism (includes copying code!)
- making stuff up
- unacceptable collaboration
- self-plagiarism

➡ Read <https://provost.asu.edu/academic-integrity>

Goals for the next 2 weeks

- set up working environment (bash, git, python)
- learn using the bash shell
- learn basic Python
- learn using git for version control
- learn using NumPy and matplotlib

Goals for the next 2 weeks

- set up working environment (bash, git, python)
- learn using the bash shell
- learn basic Python
- learn using git for version control
- learn using NumPy and matplotlib

Sticky notes™ alert system



help, please



done

Setting up...

<https://goo.gl/bXs7NE>

<http://asu-compmethodsphysics-phy494.github.io/ASU-PHY494/>

(Links also in Modules on Canvas)