Simulation of drug targets and simulation analysis

Ron Dror

Jan. 11, 2017

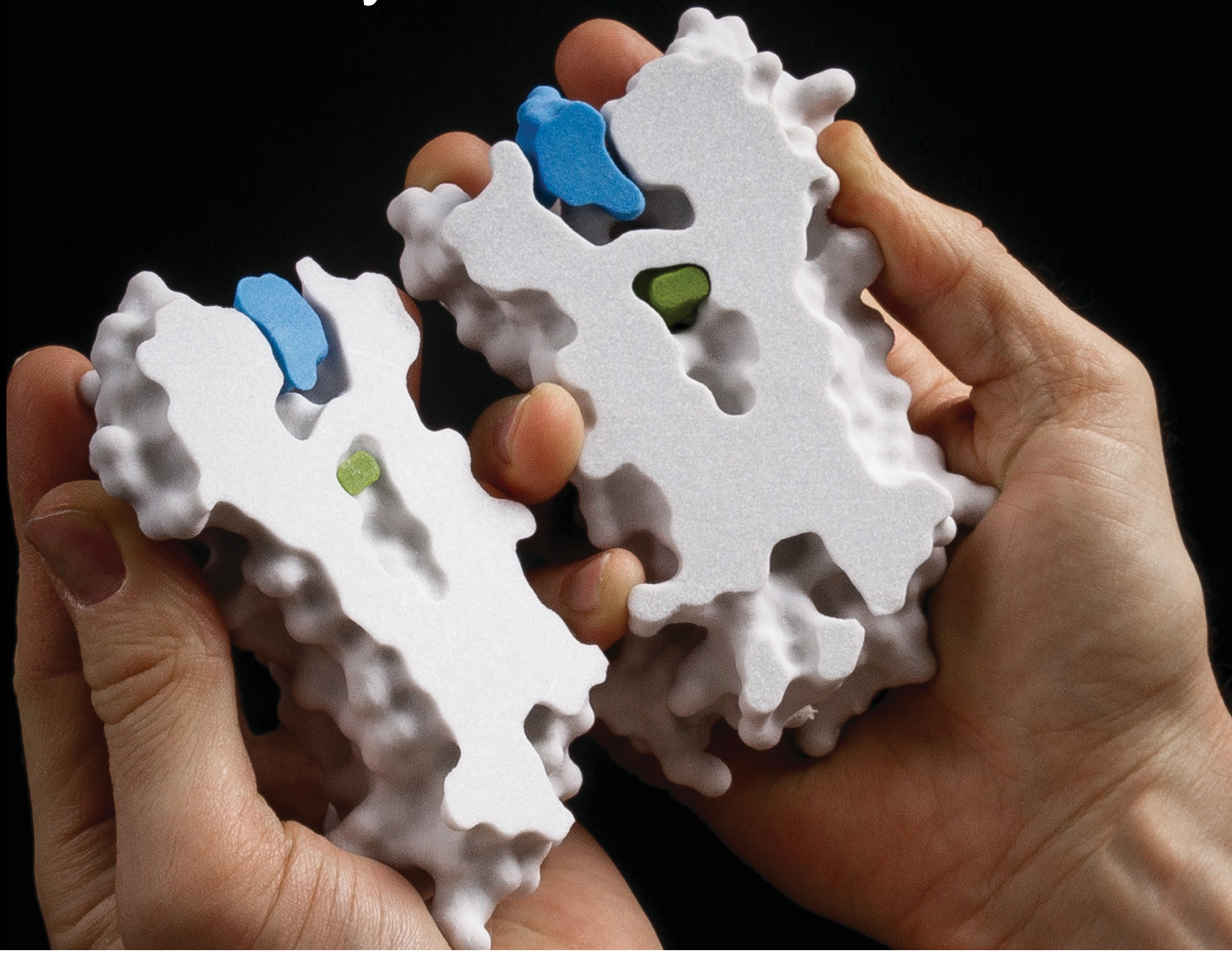


Image credit:
Ansgar Philippsen

A few logistical items

Reminders

- Monday, Jan. 16: Guest lecture by Eli Groban (Autodesk) on virtual reality for biomolecules
  - *Completely optional (MLK day).*
  - Bring a smart phone (Android or iPhone) if you have one
- Please fill out preference forms by Tuesday, Jan. 17
  - Presentations: https://goo.gl/forms/G1ZArNmkpiXXq1U53
  - Critiques: https://goo.gl/forms/fEIPmibEnZfk7s262
  - If you're not sure by then whether you're taking the class, indicate that in the "Additional notes" field
- Still taking volunteers for Jan. 25 presentations ("Modern Protein Design")

Critiques

- Each critique should be submitted (through Canvas) before the beginning of the class during which the paper is being presented
- We'll assign each student two papers to critique, based on their submitted preferences.
  - The third can be on any of the "Main Papers" listed at http://cs371.stanford.edu/schedule.html, but you need to present on one topic and critique three others
- We've posted sample critiques on the web site
- Critiques should typically be 2–3 pages long.
  - If you prefer, you can substitute two one-page critiques (on two different papers) for one "regular" critique.

Overview

- Molecular dynamics (MD) simulations predict the atomic-level motions of molecules

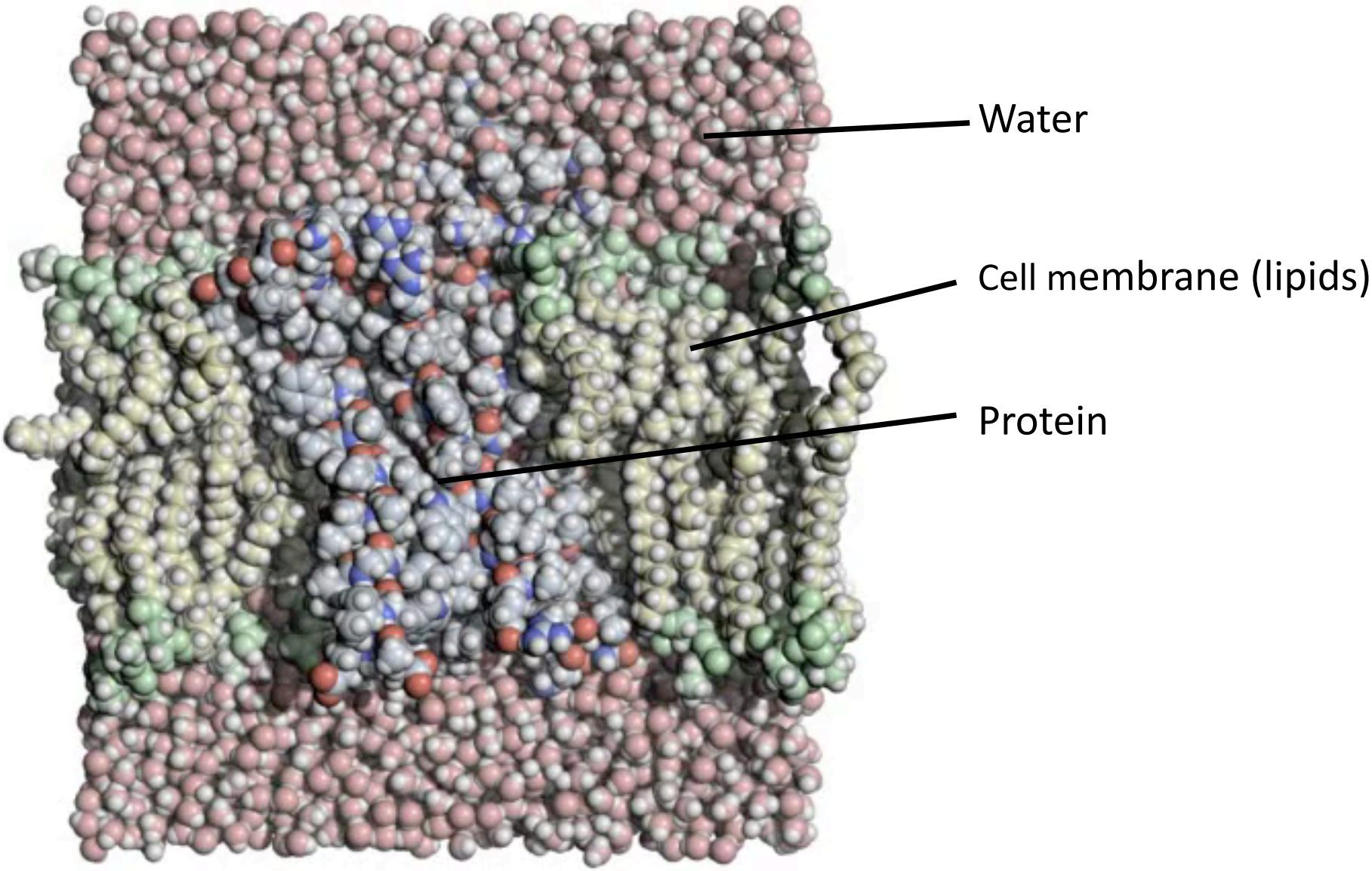
- These simulations have been around for decades, but they've become much more powerful recently, thanks to faster computers, better algorithms, and better models of the underlying physics
  - First paper illustrates application of modern MD to a problem of interest in drug discovery

- MD simulations generate a lot of data, and extracting all the important information from that data is challenging
  - Second paper presents a statistical method for detecting important "events" in MD simulations

Background:
Molecular dynamics (MD) simulations

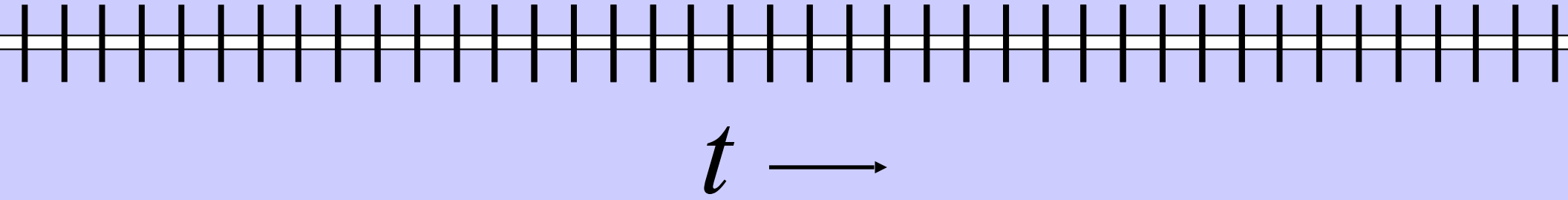
An MD simulation computes the motion of every atom in a molecular system

For example, a protein and its surroundings



Molecular Dynamics

Divide time into discrete time steps

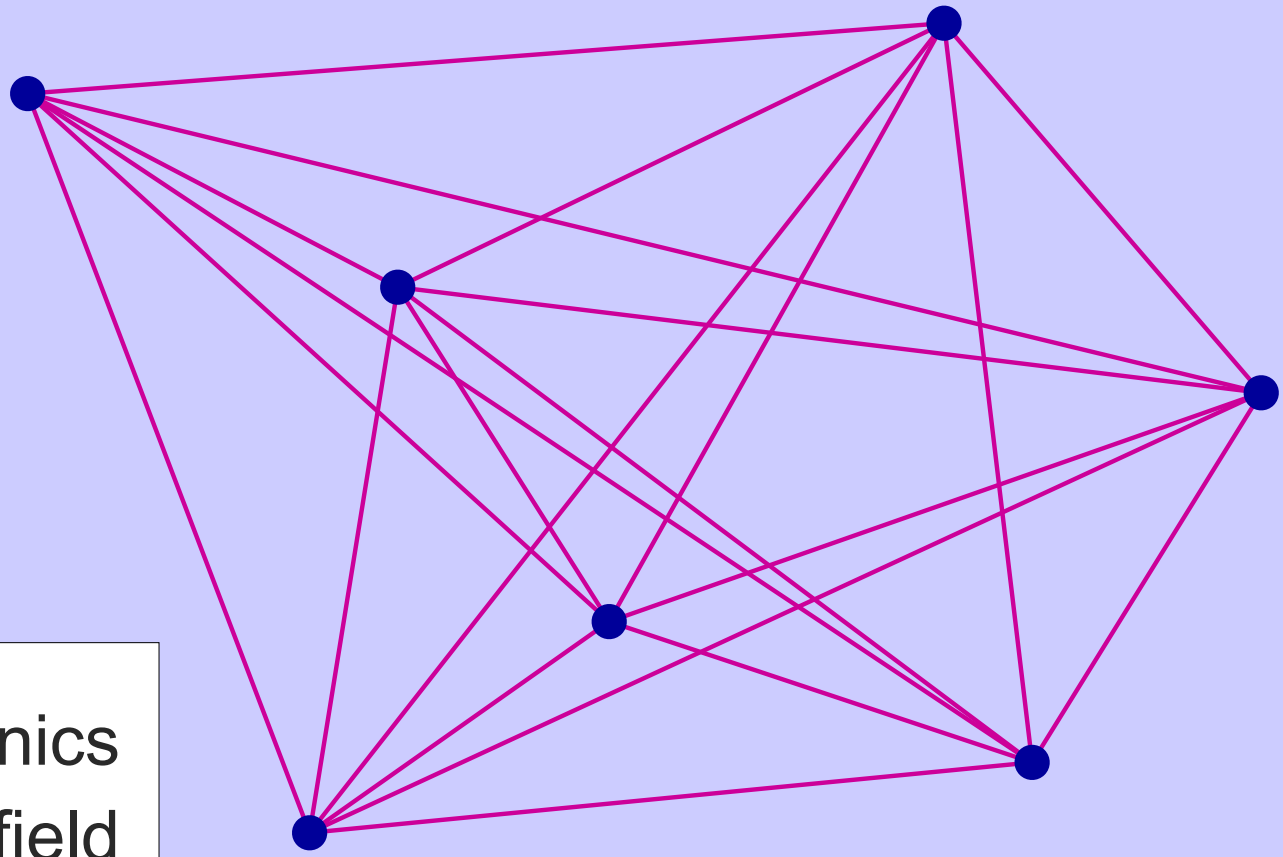


$t \longrightarrow$

~2 fs time step

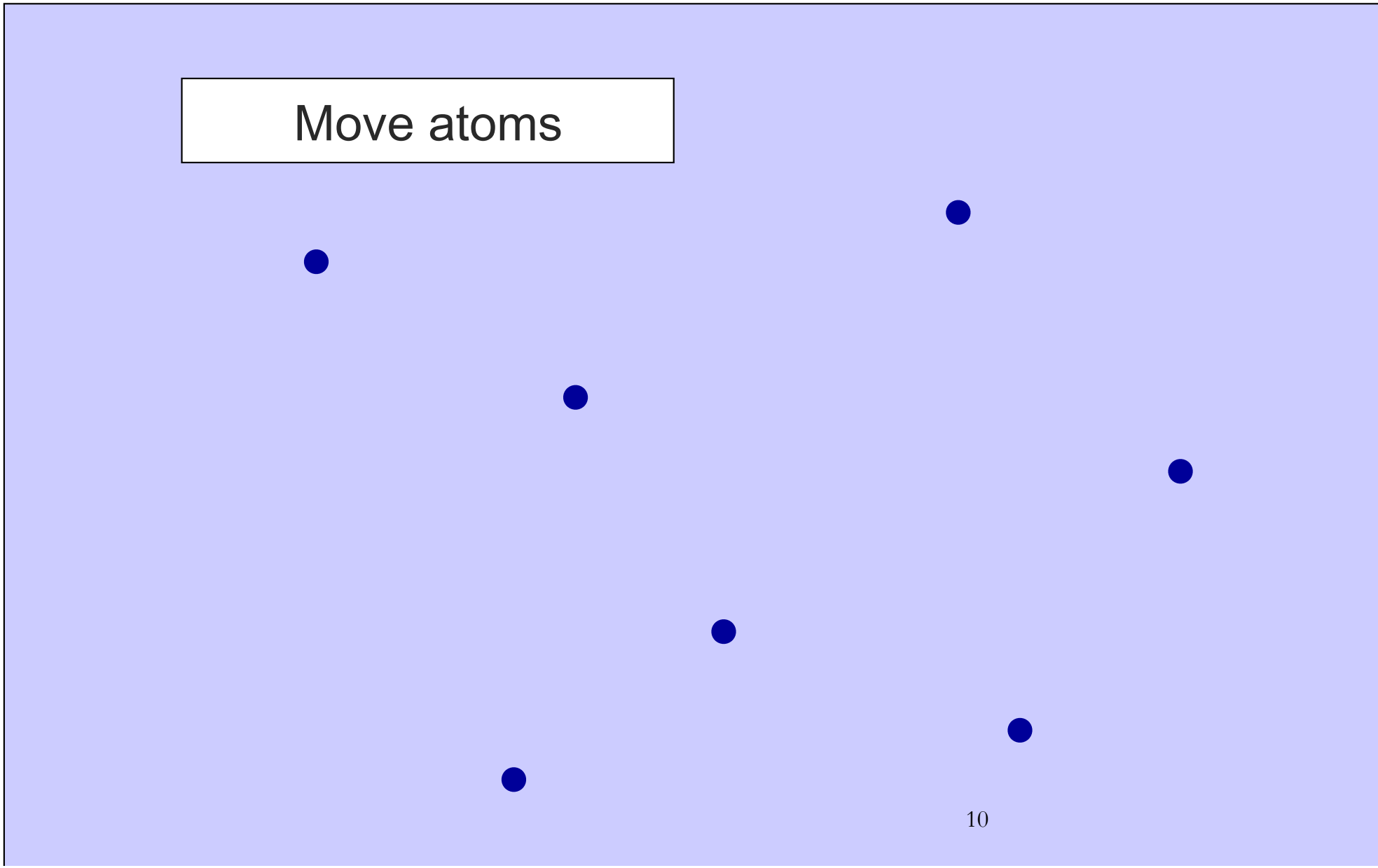
Molecular Dynamics

Calculate forces exerted by atoms on one another

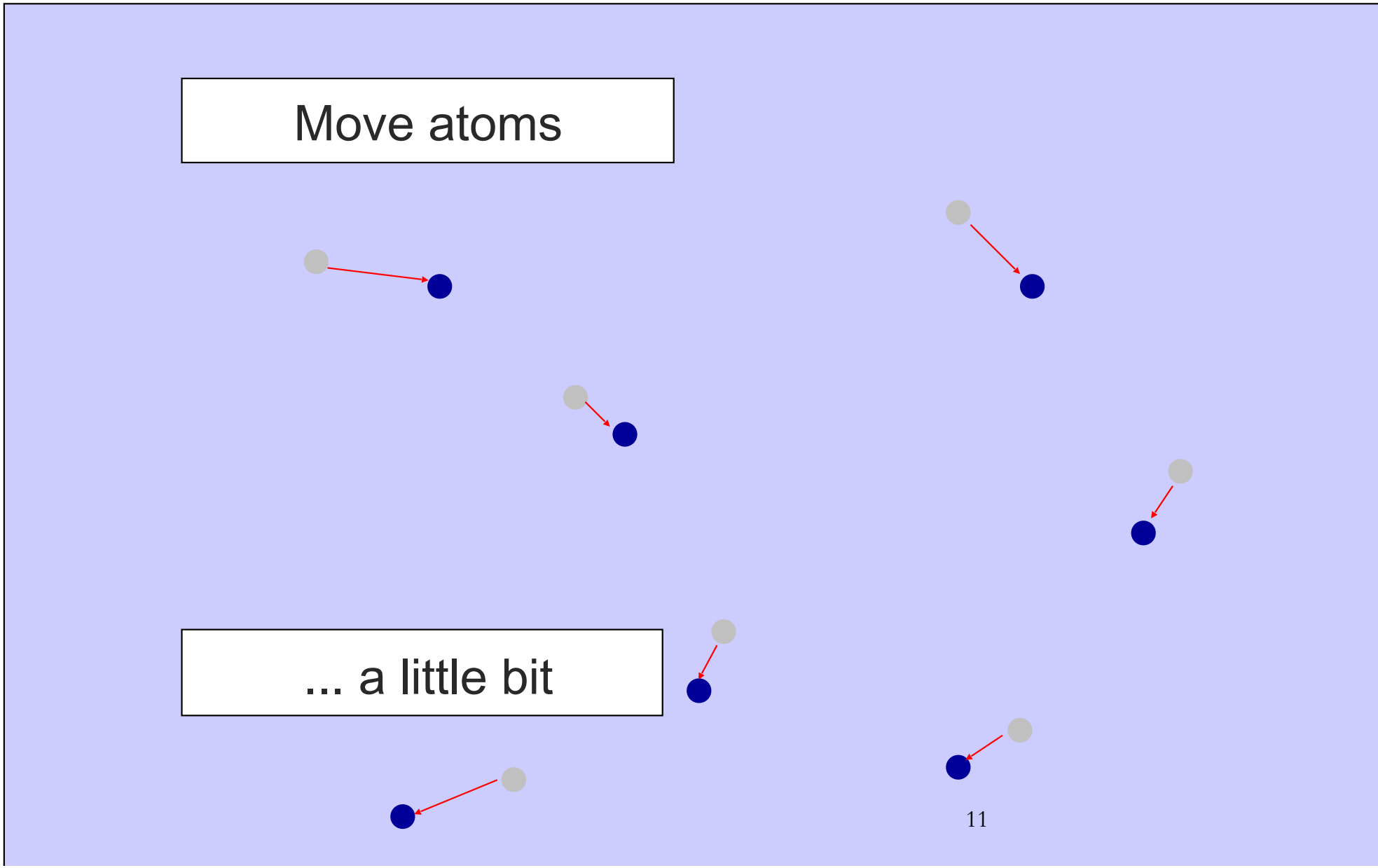


Molecular mechanics force field

Molecular Dynamics



Molecular Dynamics



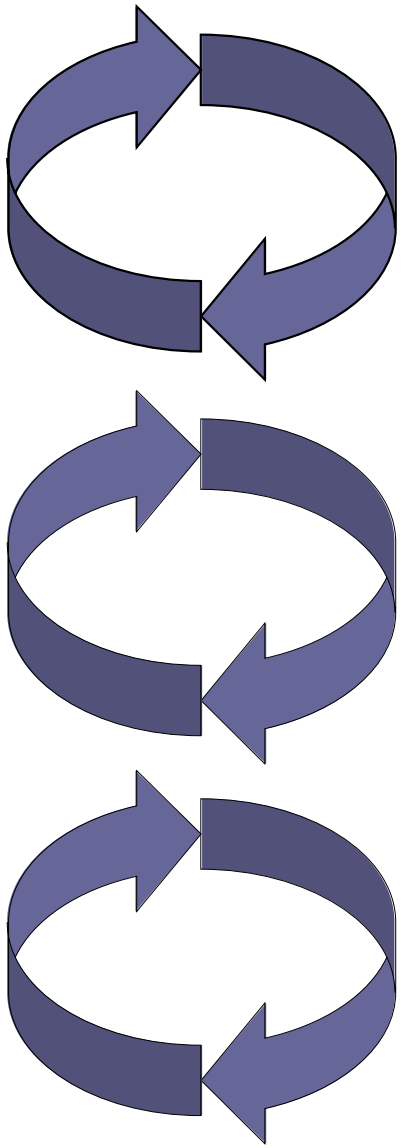
Molecular Dynamics

Iterate

... and iterate

... and iterate

Integrate Newton's laws of motion

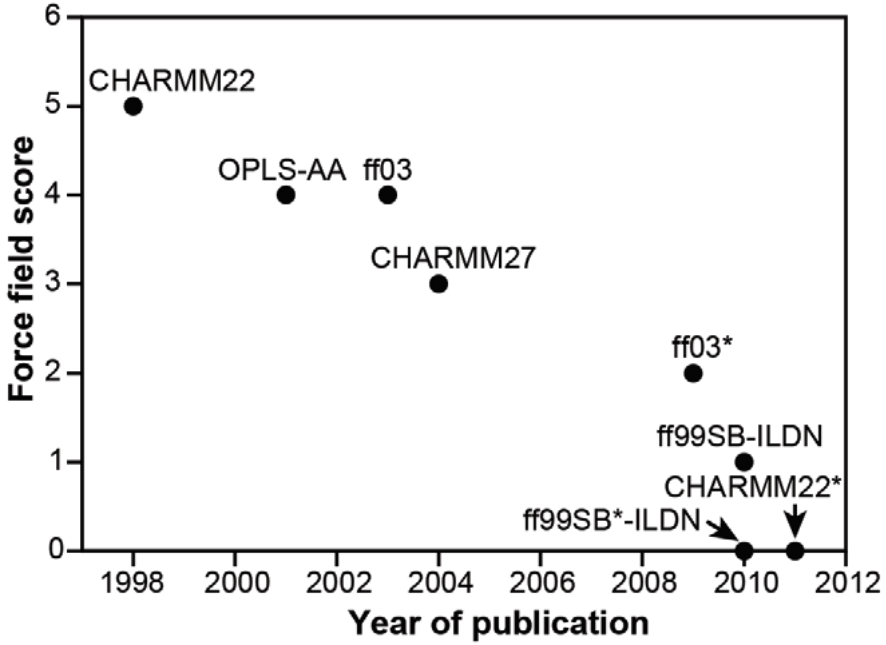


MD simulations are slow, but they've sped up substantially

- Simulations require short time steps for numerical stability
  - 1 time step ≈ 2 fs (2×10^{-15} s)
- Structural changes in proteins can take nanoseconds (10^{-9} s), microseconds (10^{-6} s), milliseconds (10^{-3} s), or longer
  - Millions to trillions of sequential time steps for nanosecond to millisecond events
- Until recently, simulations of 1 microsecond were rare
- Advances in computer power have enabled microsecond simulations (and even millisecond simulations in special cases)
- Enabling longer simulations is an active research area, involving:
  - Algorithmic improvements
  - Parallel computing
  - Hardware: GPUs, specialized hardware

MD simulations are approximate, but they've become more accurate

- Molecular mechanics force fields (the functional forms used to compute forces on atoms) are inherently approximations
- They have improved substantially over the last decade, though many limitations remain



Here force fields with lower scores are better, as assessed by agreement between simulations and experimental data. Even the force fields with scores of zero are imperfect, however!

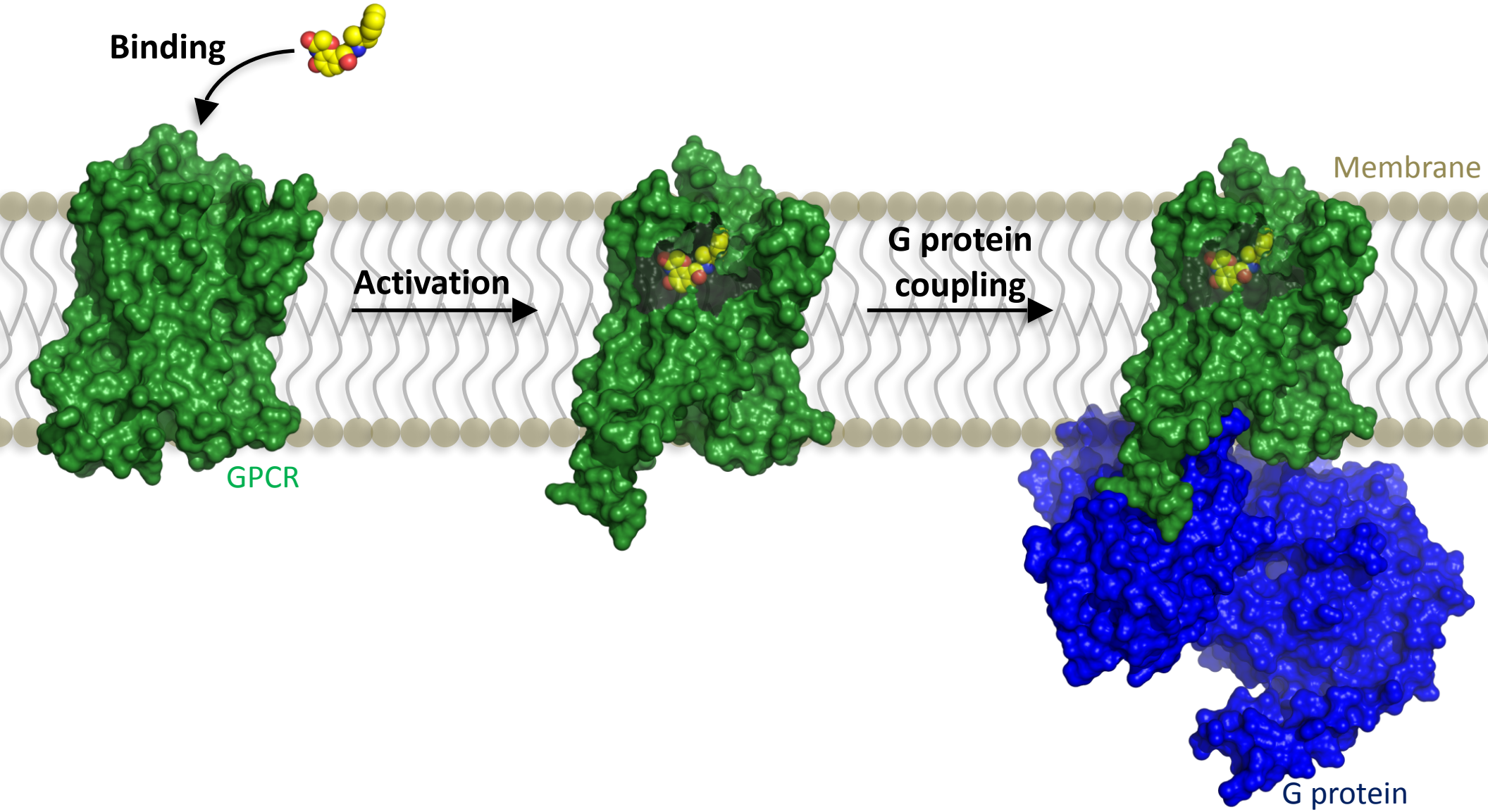
Lindorff-Larsen et al., *PLOS One*, 2012

An application of MD simulation:

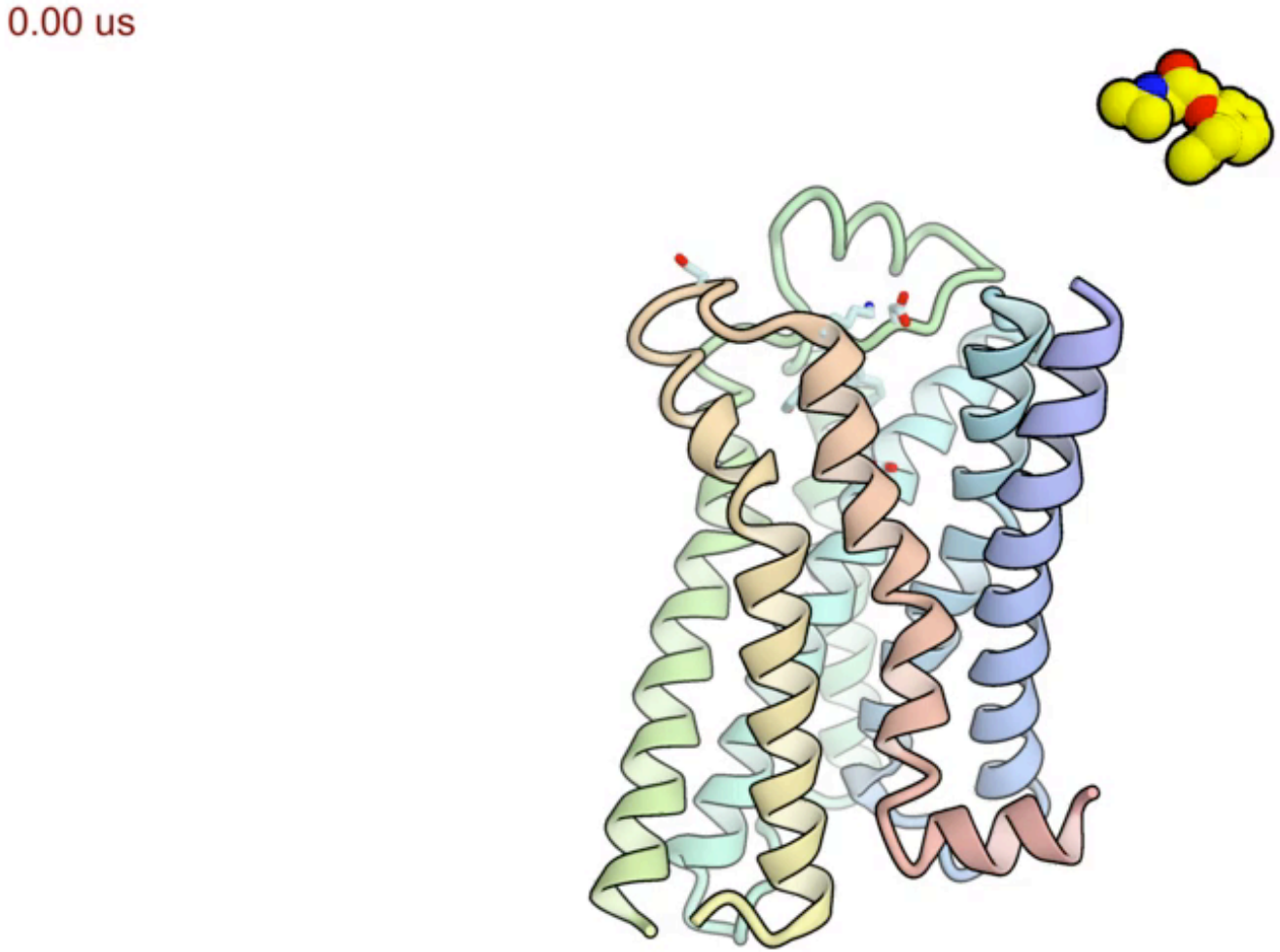
“Structural basis for modulation of a G-protein-coupled receptor by allosteric drugs”

Dror et al., *Nature* 503: 295–299 (2013)

G protein–coupled receptor (GPCR) signaling



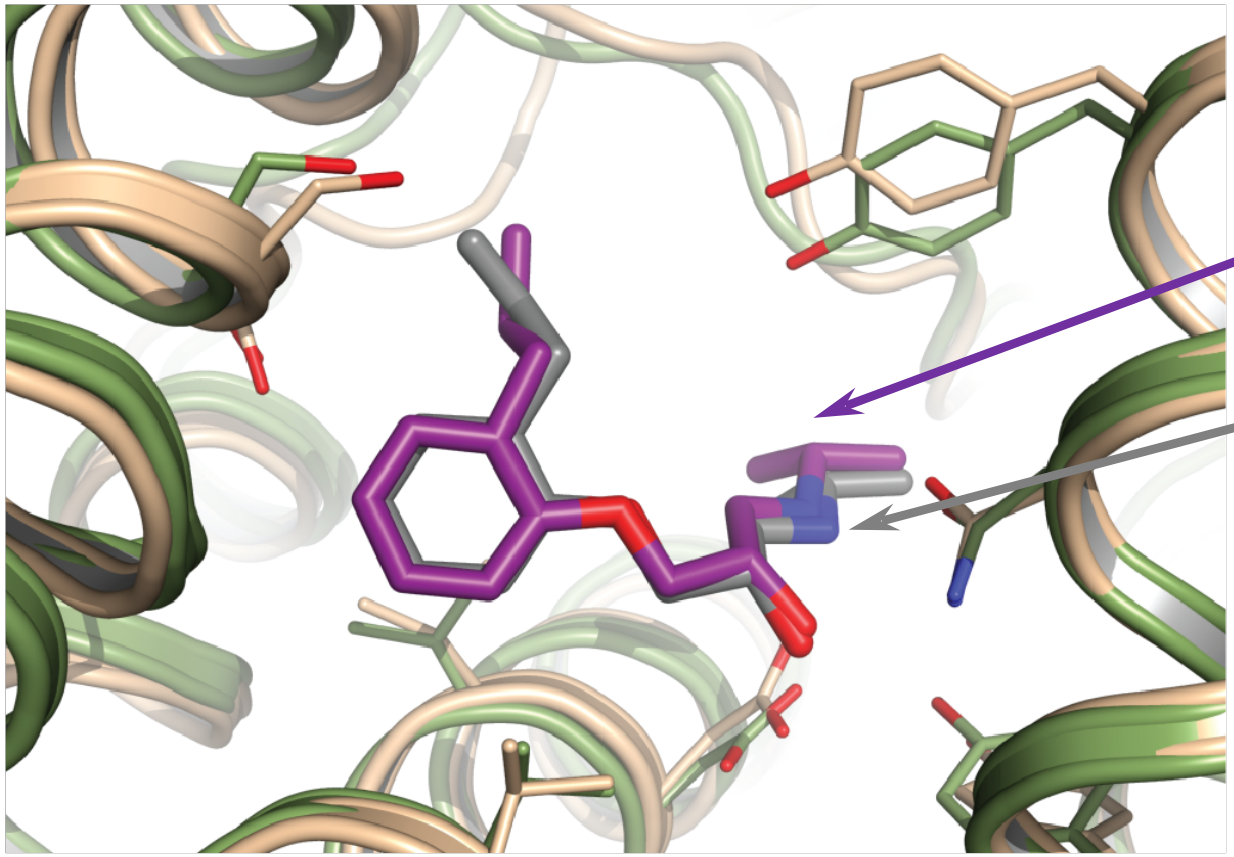
Spontaneous drug binding in simulation



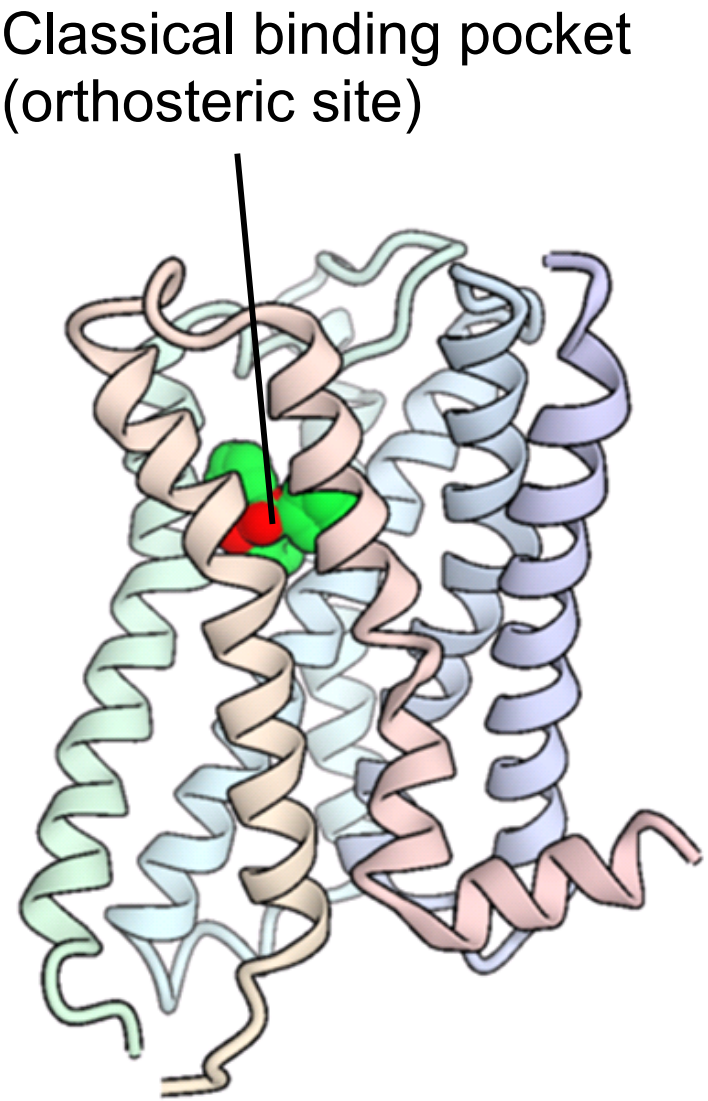
Beta-blocker alprenolol binding to the β_2-adrenergic receptor

Dror et al., *PNAS* 2011

Final pose matches experimental structure

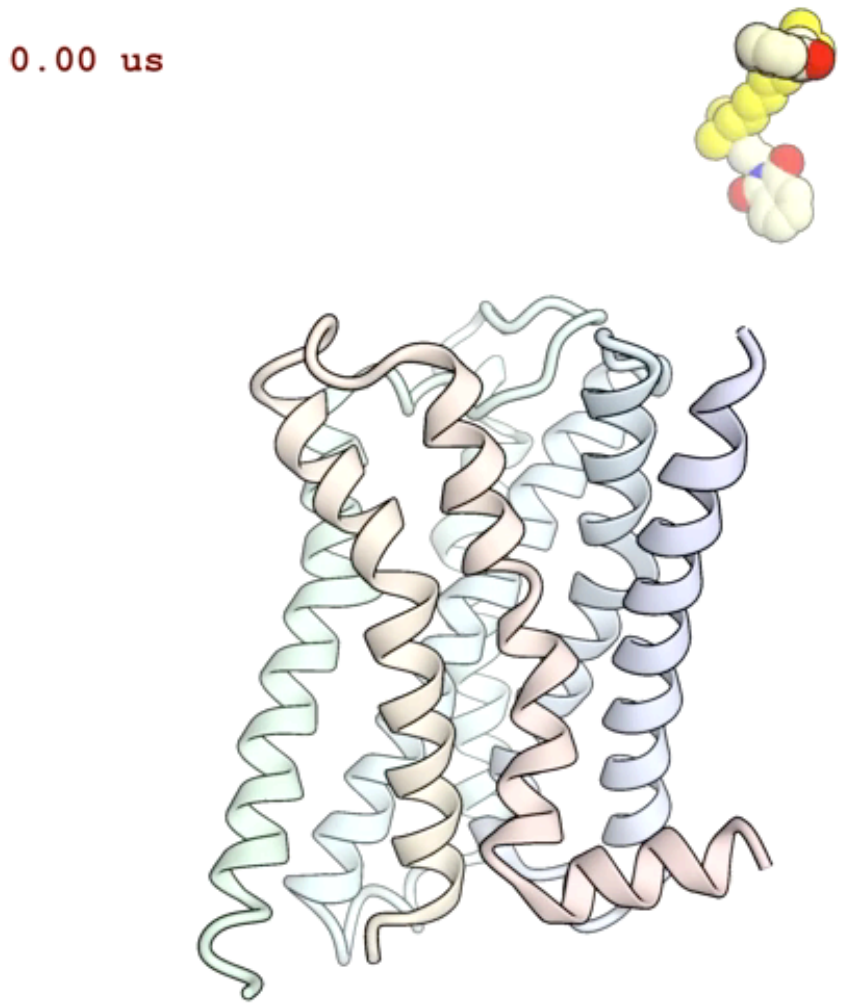


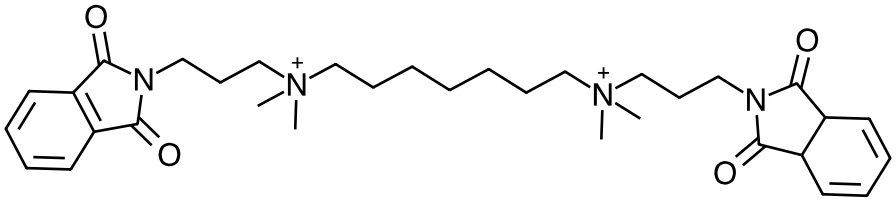
Allosteric drugs



- Allosteric modulators bind anywhere but the orthosteric site
- Allosteric modulators promise:
  - Selectivity between GPCR subtypes
  - Fine control of responses to body's natural signaling patterns
- Until recently, not clear how such modulators bind, and even less clear how they exert their effects

Allosteric modulator binding to muscarinic acetylcholine receptor

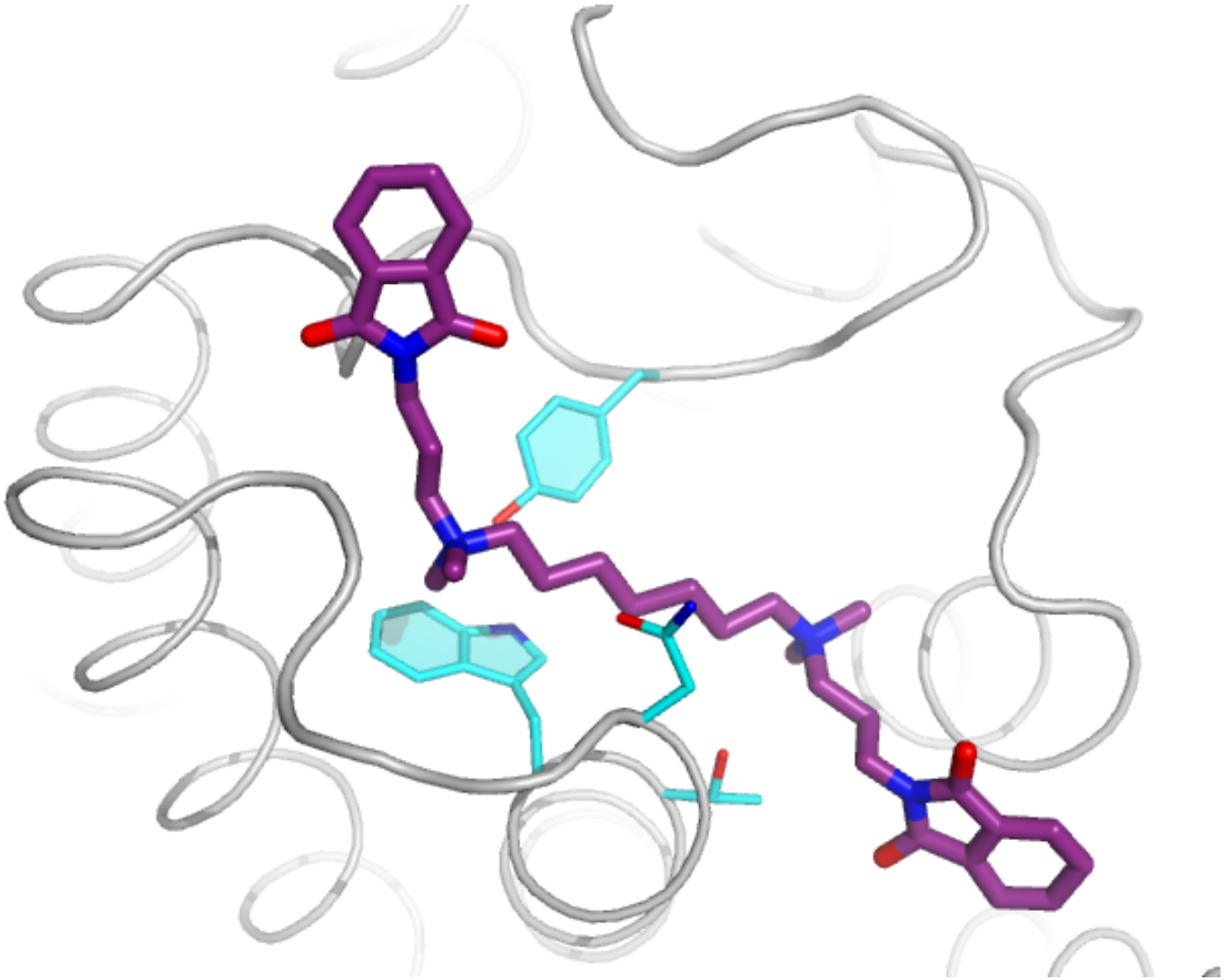




C_7/3-phth binding to M2 muscarinic receptor

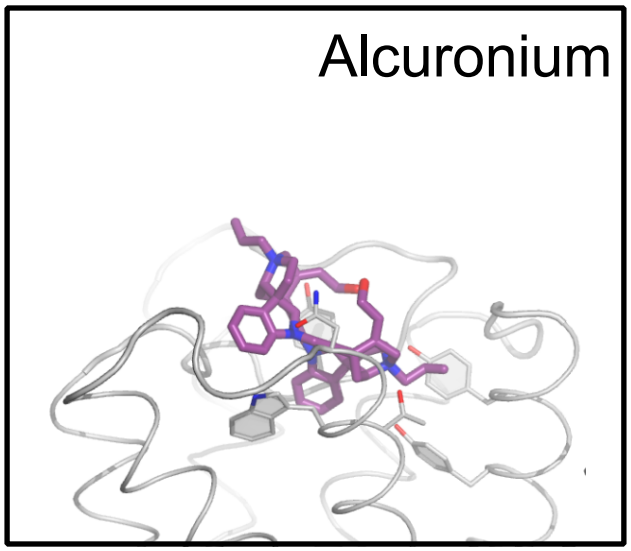
Dror et al., *Nature* 2013

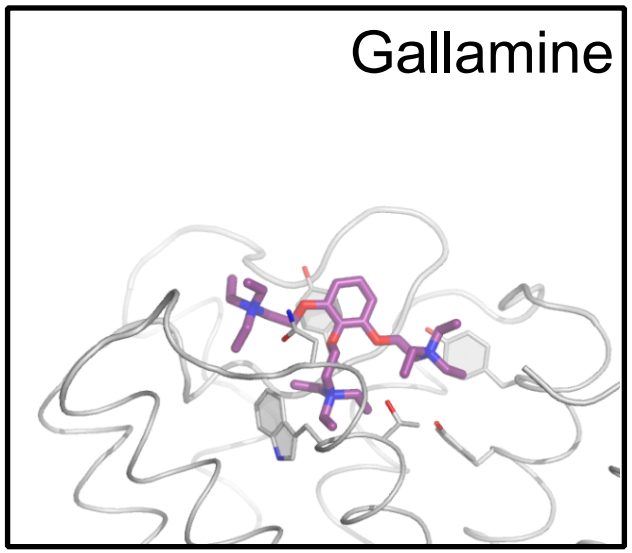
Bound pose agrees with mutagenesis data

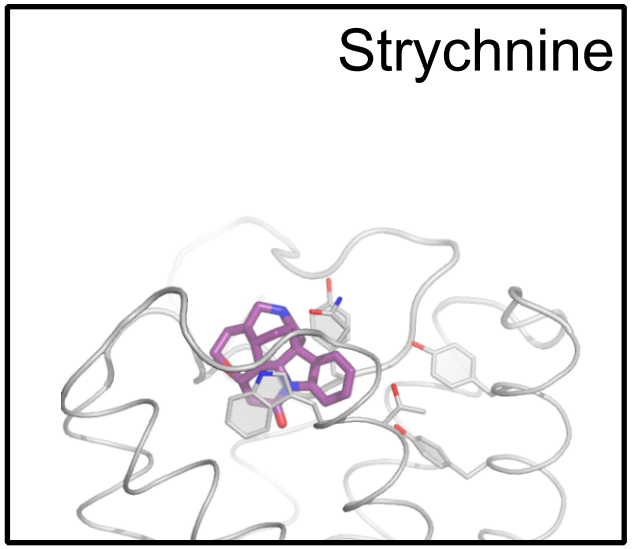


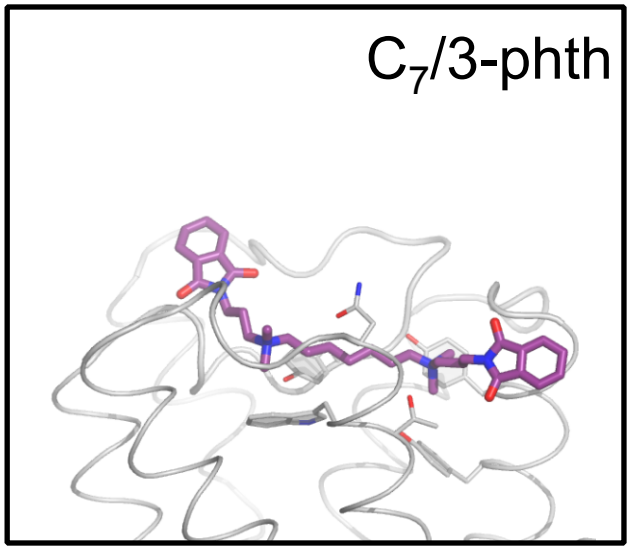
- Residues whose mutation caused >5-fold loss of affinity for bis-amino alkane ligands

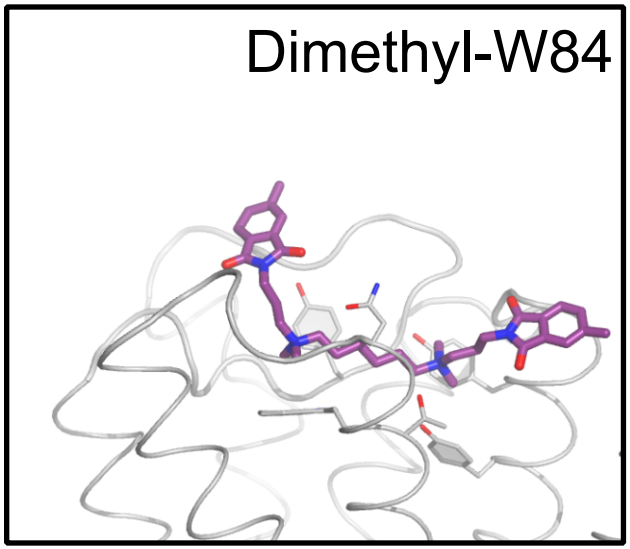
Apply the same methodology to structurally diverse allosteric modulators ...



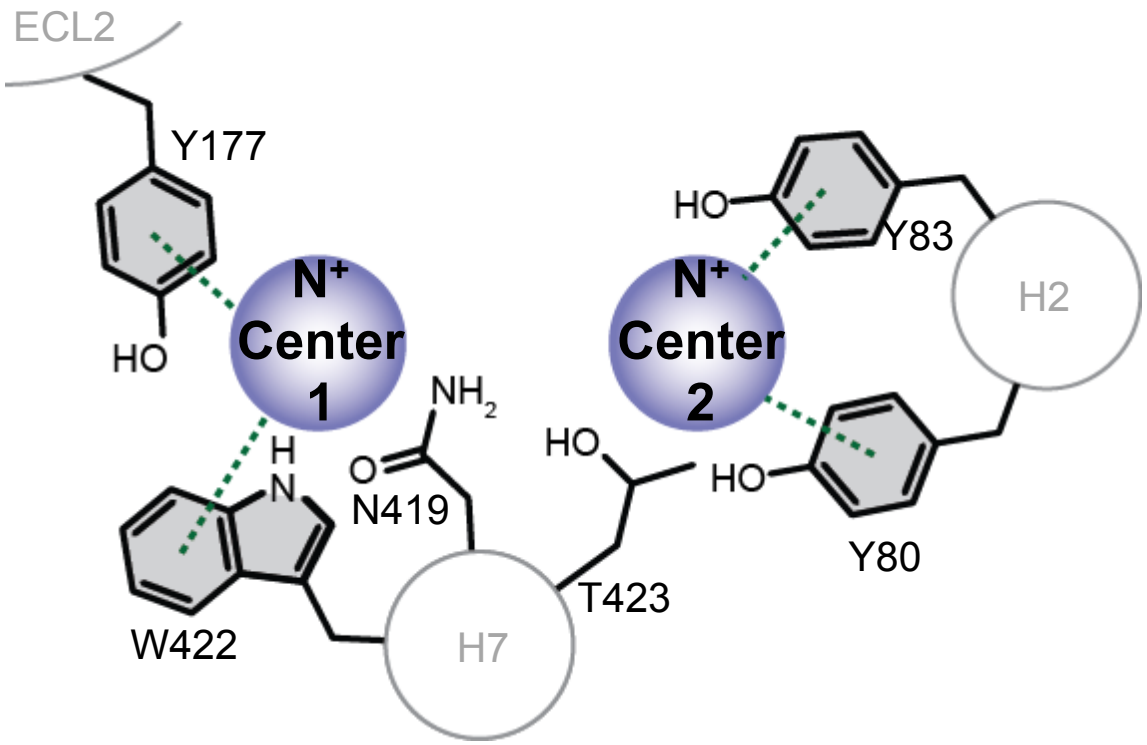


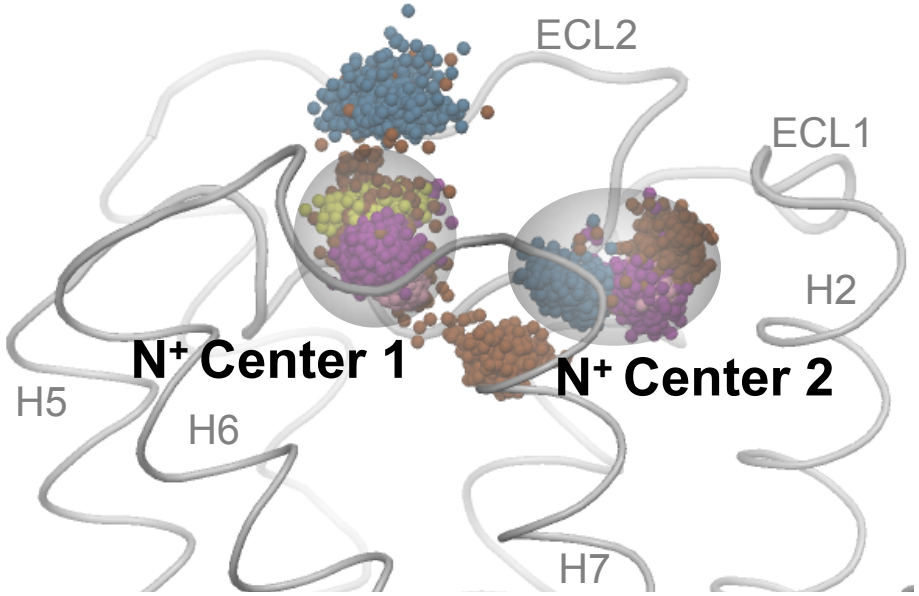






Structurally diverse allosteric modulators share a common binding mode



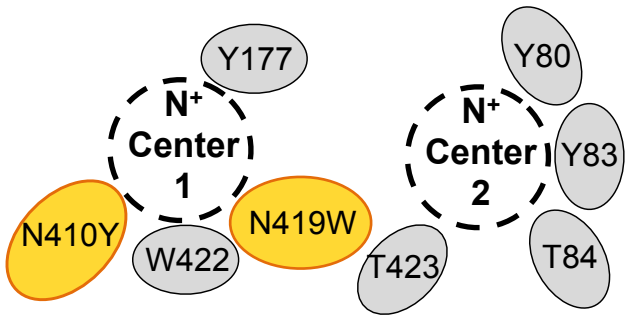


- Each modulator binds with a positively charged nitrogen in one or or both of two positions
- These binding modes are different from those predicted previously

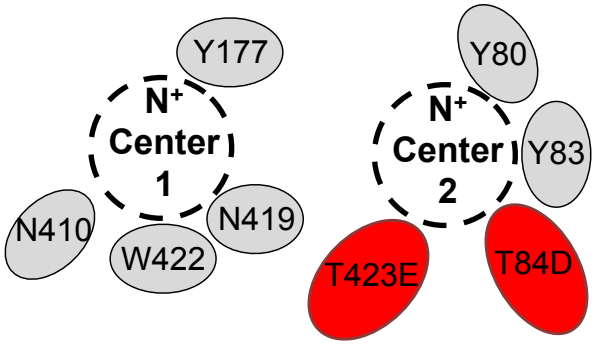
Experimental validation of new predictions

Mutagenesis predictions

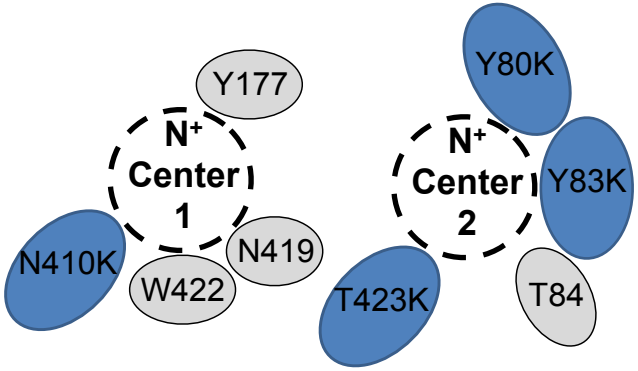
Increase affinity
Increased cation–π interactions



Increase affinity
Charge–charge attraction

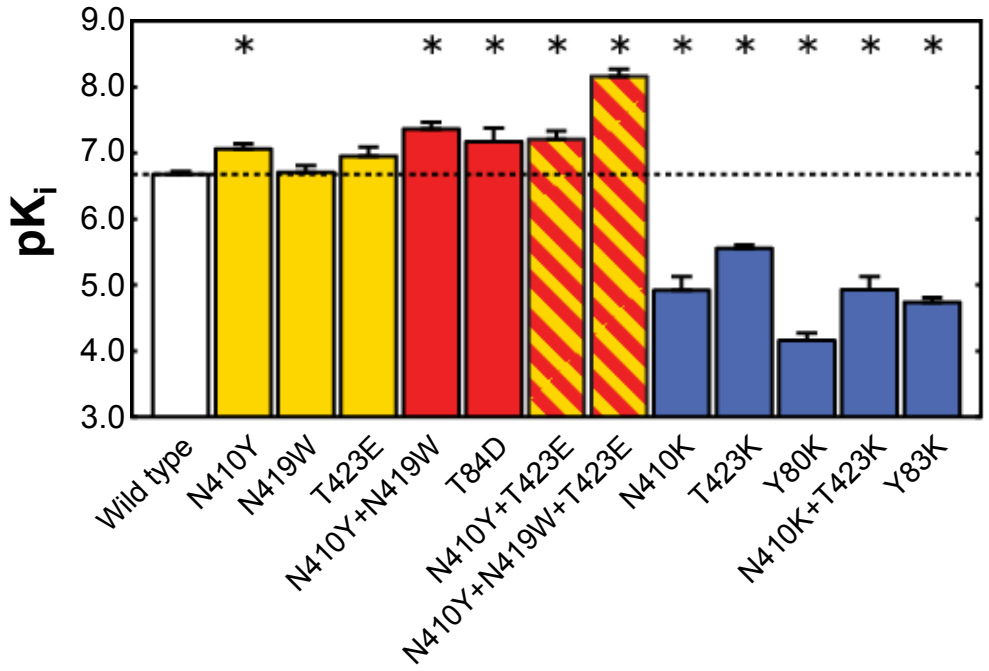


Decrease affinity
Charge–charge repulsion

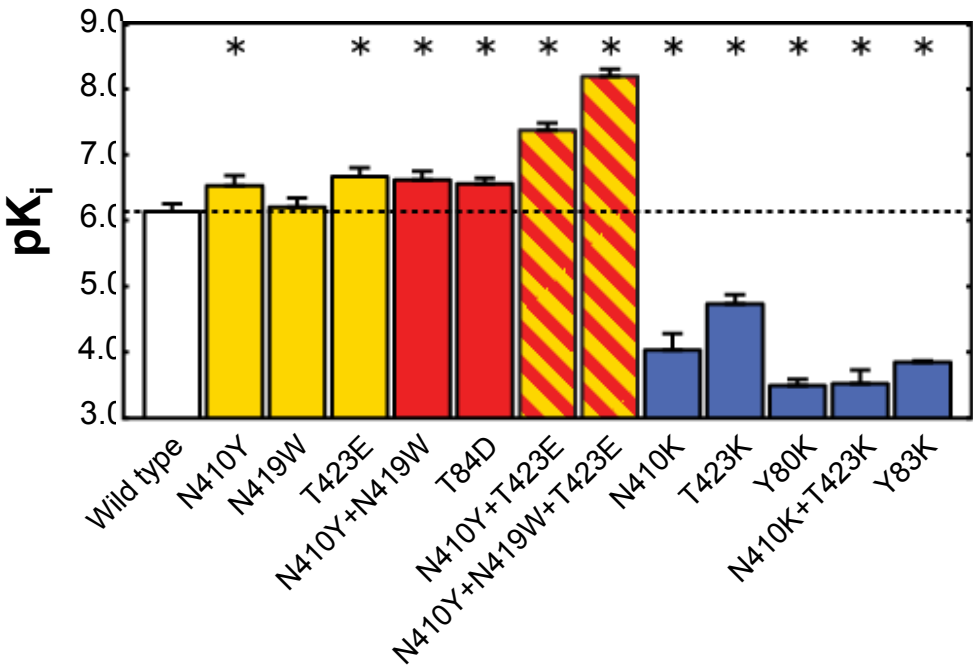


Mutagenesis results (Christopoulos lab, Monash University)

C_7/3-phth



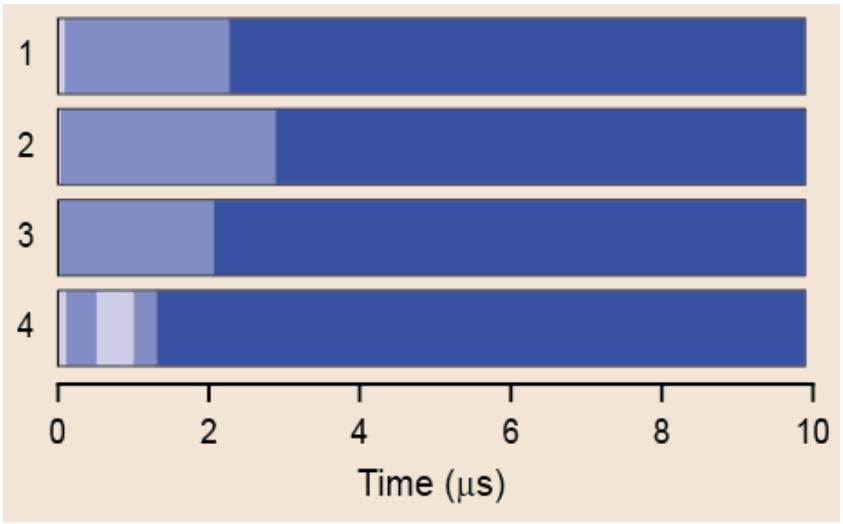
Gallamine



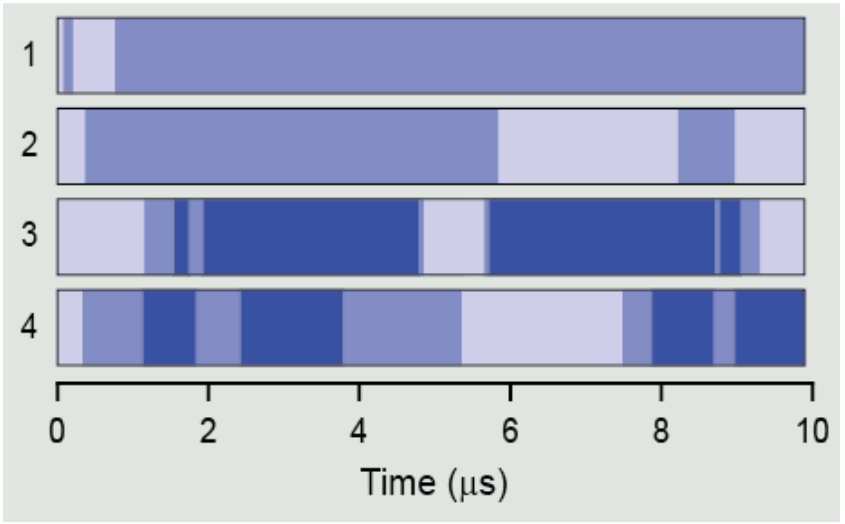
What is the basis of allosteric modulation?

- In simulation, as in experiment, allosteric ligands modulate affinity of orthosteric ligands.
  - C_7/3-phth, a negative allosteric modulator (NAM), lowers affinity of the classical antagonist NMS

No ligand in orthosteric site



NMS in orthosteric site



Allosteric ligand tightly bound

Allosteric ligand loosely bound

Allosteric ligand not bound

Dror et al., *Nature* 2013

Allostery is *symmetric*

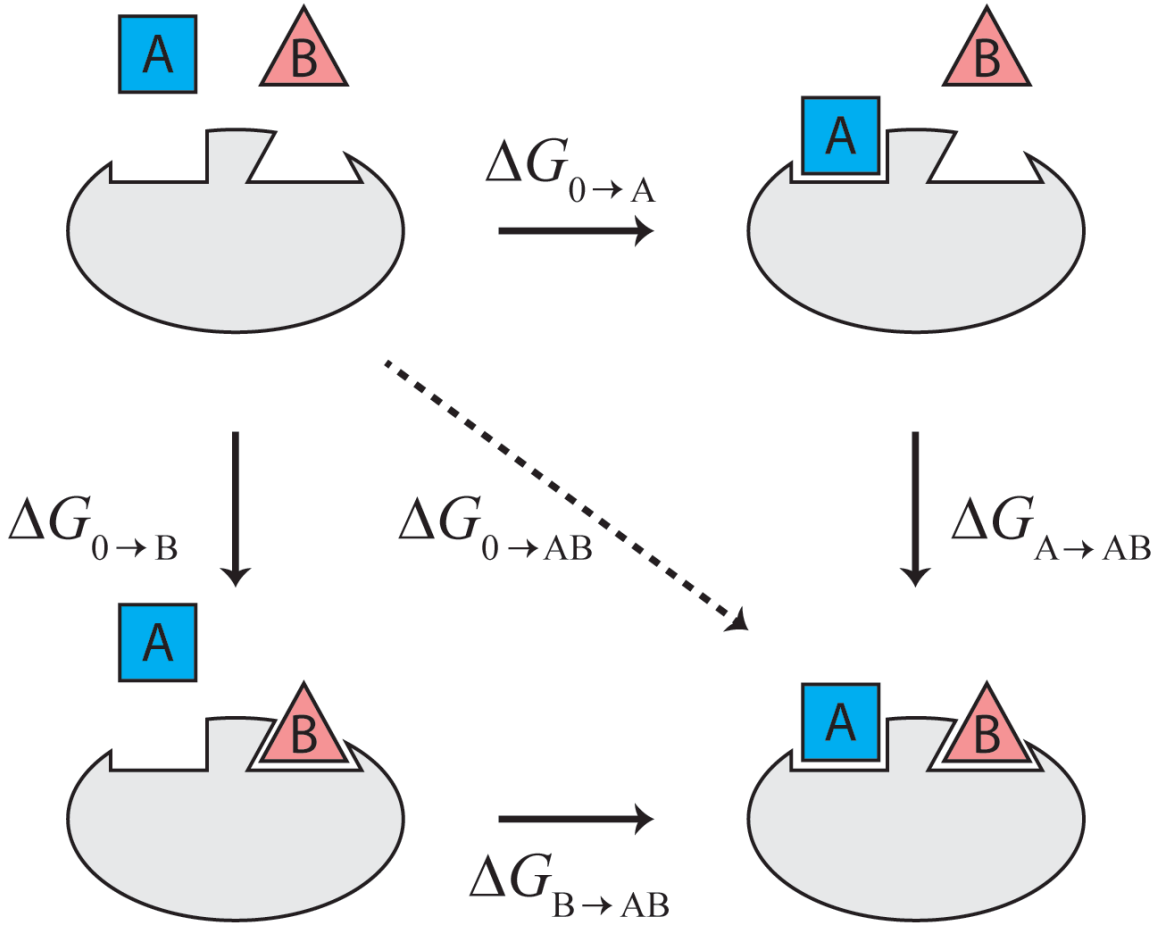


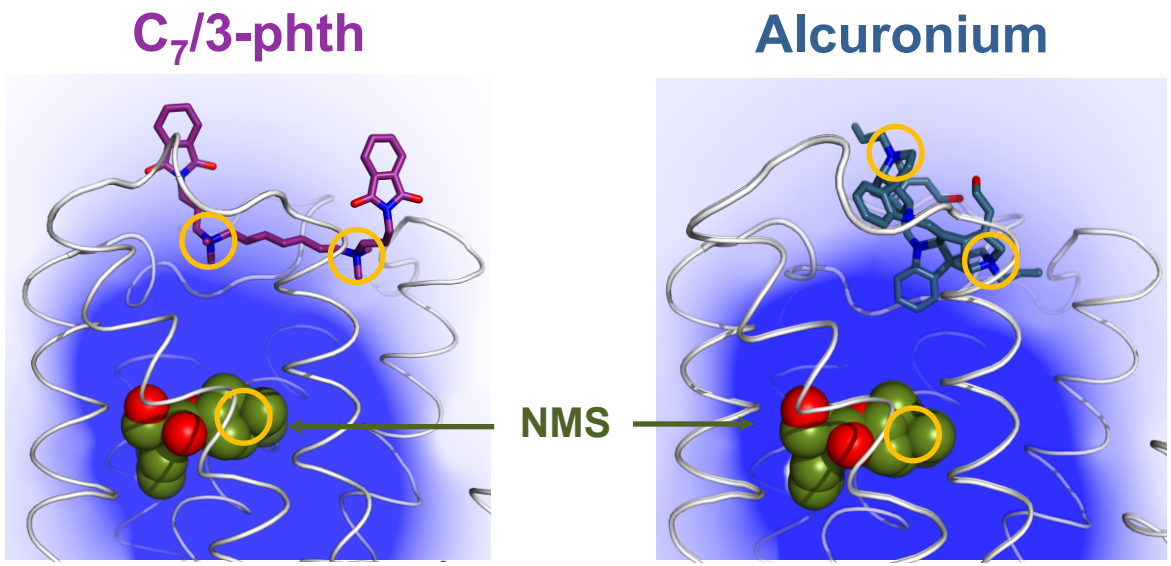
Figure credit: Sam Hertig

$$\Delta G_{A\to AB} - \Delta G_{0\to B} = \Delta G_{B\to AB} - \Delta G_{0\to A}$$

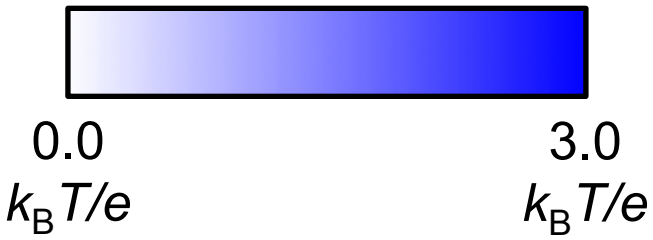
Difference in binding energy of A with or without B bound equals difference in binding energy of B with or without A bound

Mechanism 1: Electrostatic interaction between ligands

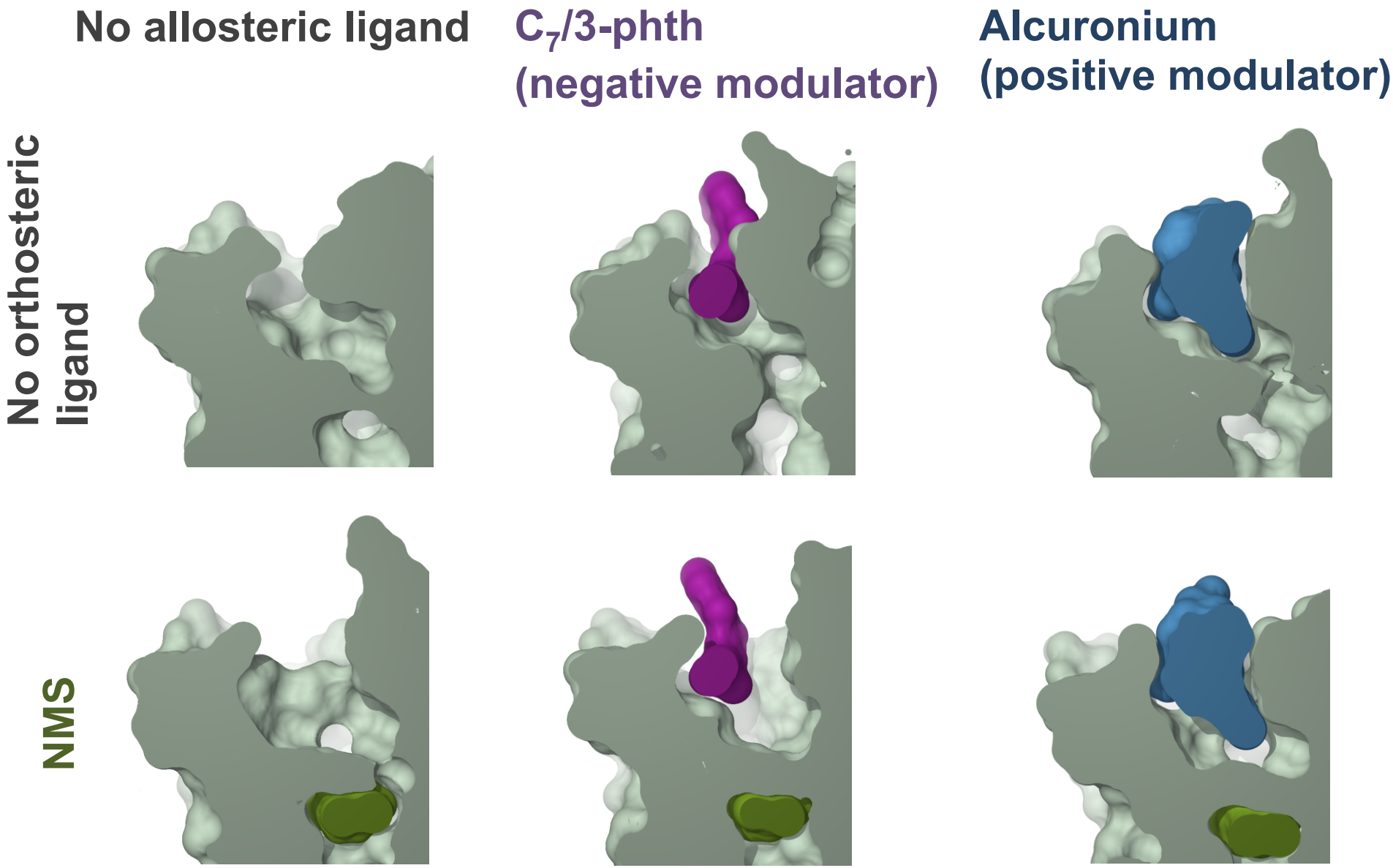
- Positively charged allosteric ligand repels positively charged orthosteric ligand



Difference in electrostatic potential due to addition of NMS



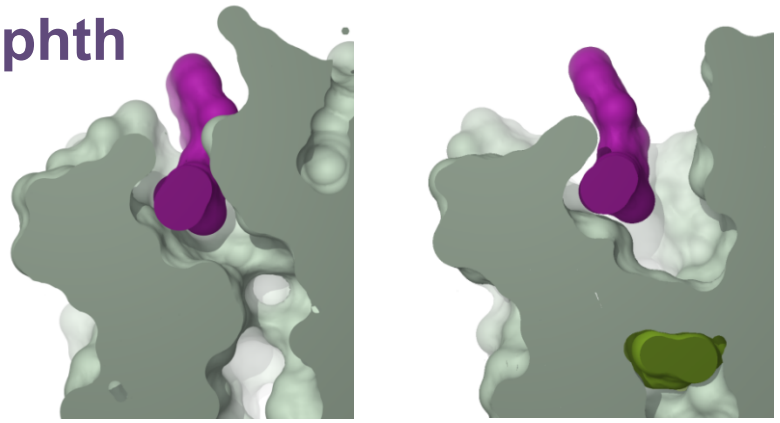
Mechanism 2: Coupled conformational change of orthosteric and allosteric sites



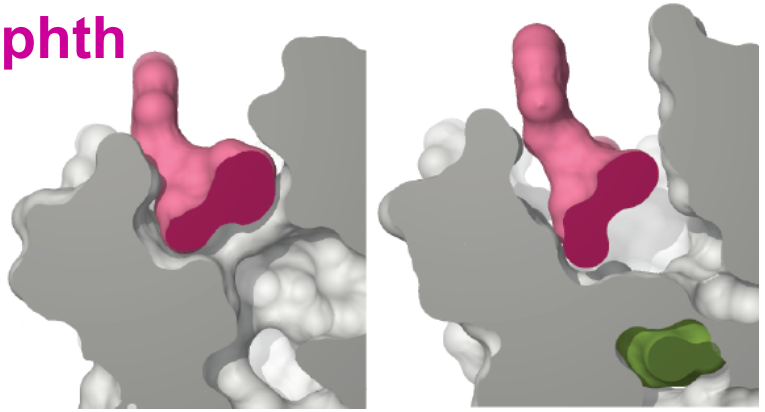
Designing an allosteric modulator

- **Prediction**: Computationally designed modulator 4P-C_7/3-phth binds like C_7/3-phth but forces open allosteric site, making cooperativity more positive (less negative)

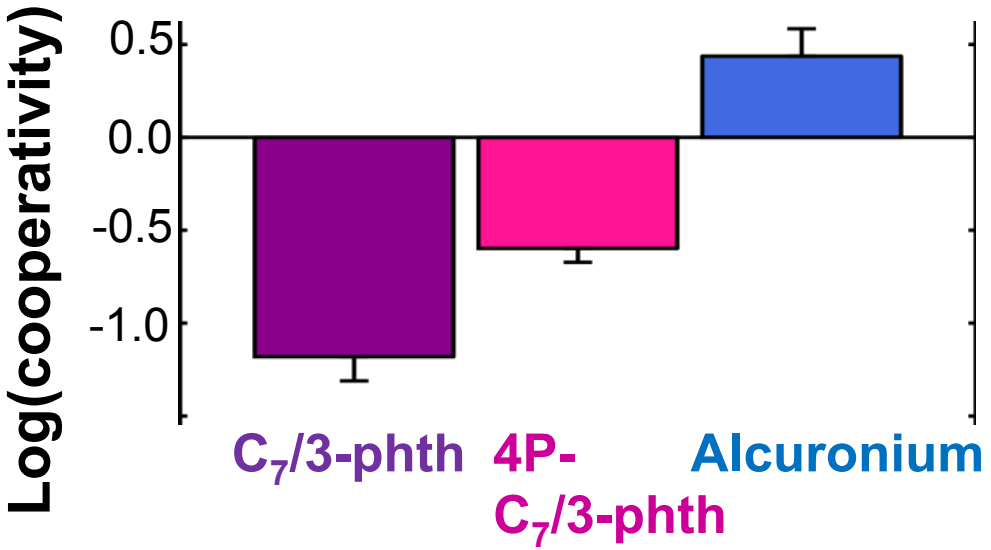
C_7/3-phth



4P-C_7/3-phth



- **Experimental validation**: 4P-C_7/3-phth has less negative cooperativity than C_7/3-phth despite binding more tightly



Limitations of study

- Computational:
  - We didn't explicitly compute cooperativity between allosteric and orthosteric ligands
  - We have good evidence for negative cooperativity between the allosteric modulator C_7/3-phth and the orthosteric ligand, but weaker evidence for cooperativity of most other allosteric modulators
- Experimental:
  - We didn't perform experimental validation of the electrostatic mechanism
  - We didn't solve a crystal structure with one of these modulators bound

Limitations of study

- The big one (in my opinion):
  We did not study allosteric modulation of acetylcholine, the neurotransmitter that is the natural ligand for this receptor
  - When developing an allosteric modulator as a drug, you generally care most about cooperativity with the natural ligand
  - Acetylcholine is an agonist: it favors activation of the receptor
  - But we had only an inactive-state structure of the receptor, not an active-state structure
  - So instead, we studied modulation of NMS, which does not favor activation

Analysis of simulation data:

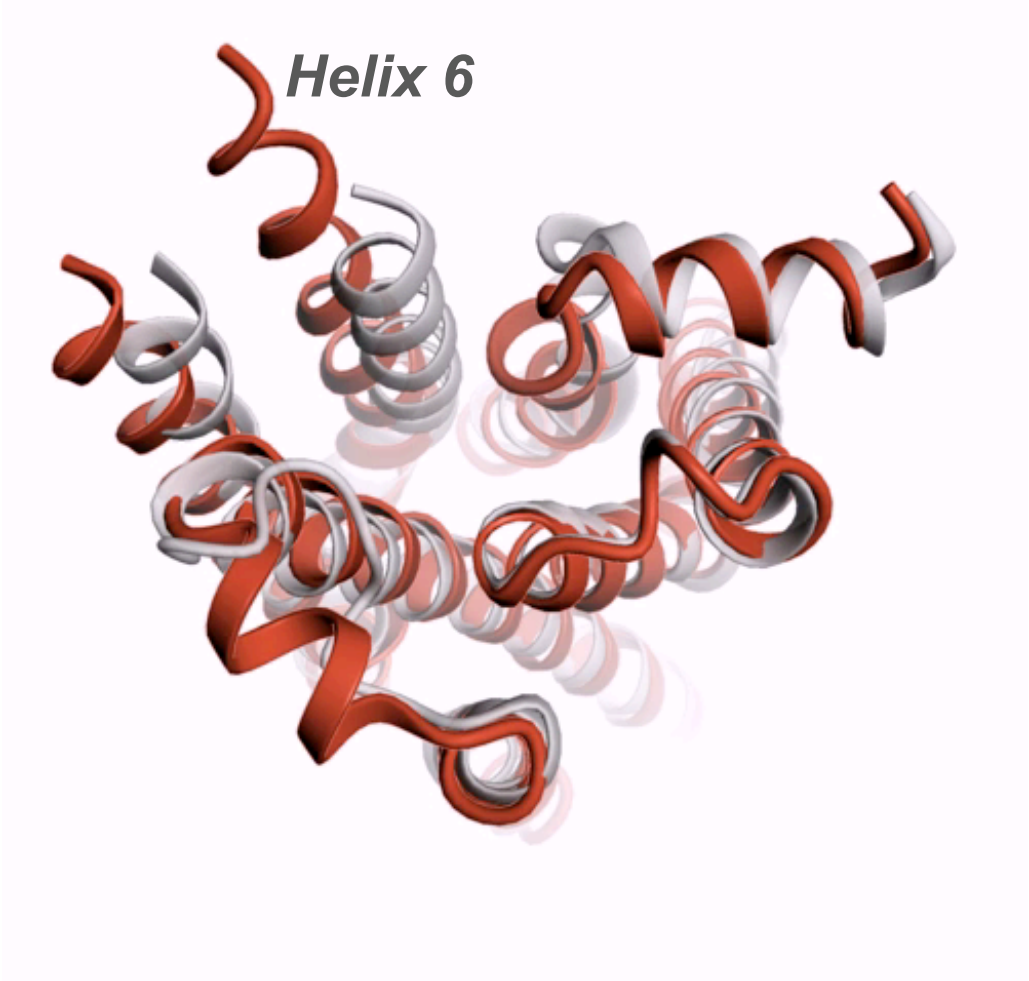
“Identifying localized changes in large systems: change-point detection for biomolecular simulations”

Fan et al., *PNAS* 112:7454–7459 (2015)

The challenge

- MD simulations generate a lot of data
  - Example: simulate a 50,000-atom system for 1 μs
  - That's half a billion time steps
  - The simulation calculates the position and velocity of every atom at every time step
  - One doesn't usually save to disk at every time step, but there's still *a lot* of data to examine
- Sometimes one knows precisely what to look for
- In other cases—particularly when using simulations to understand functional mechanisms—extracting meaningful information from simulations involves protracted visual and manual analysis (i.e., staring at the results for a long time)
- Can we automate this process?

An example: using simulation to understand the mechanism of GPCR activation

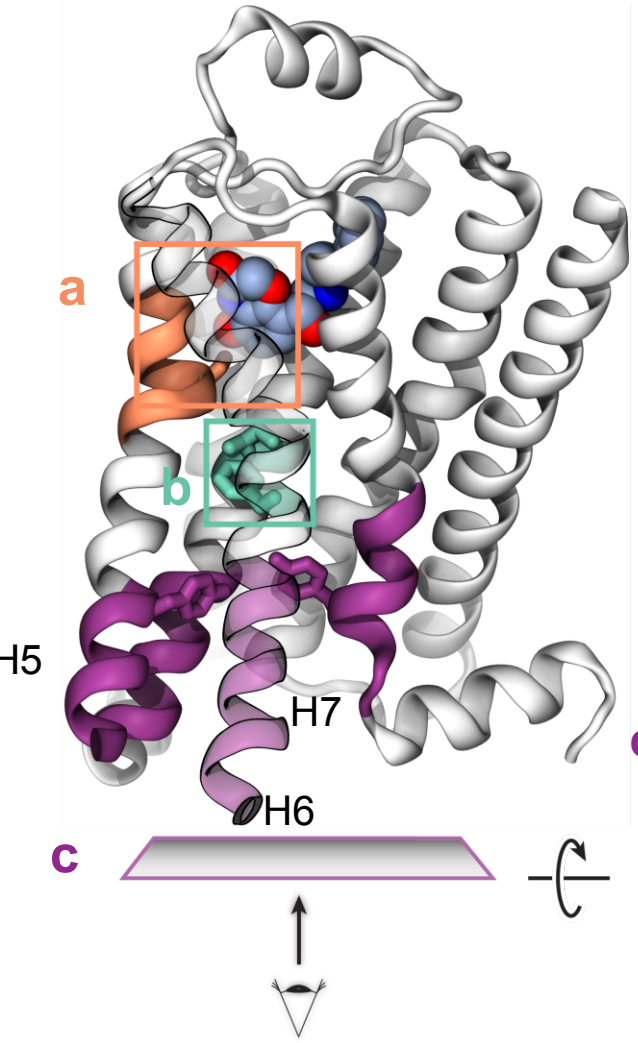


Simulation vs.
Inactive crystal structure

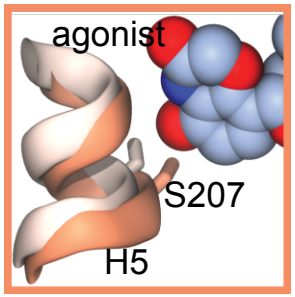
Simulation of β_2-adrenergic receptor transitioning spontaneously from its active state to its inactive state

Rosenbaum et al., *Nature* 2010; Dror et al., *PNAS* 2011

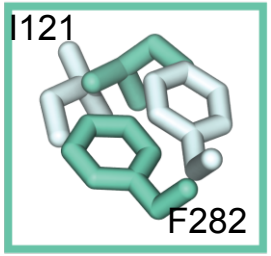
After months of staring at simulation results: There are three key, “loosely connected” regions; each adopts multiple conformational states



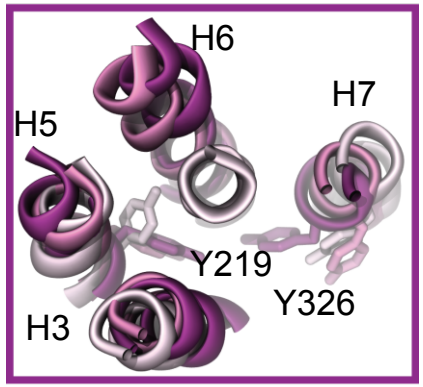
a Ligand-binding site

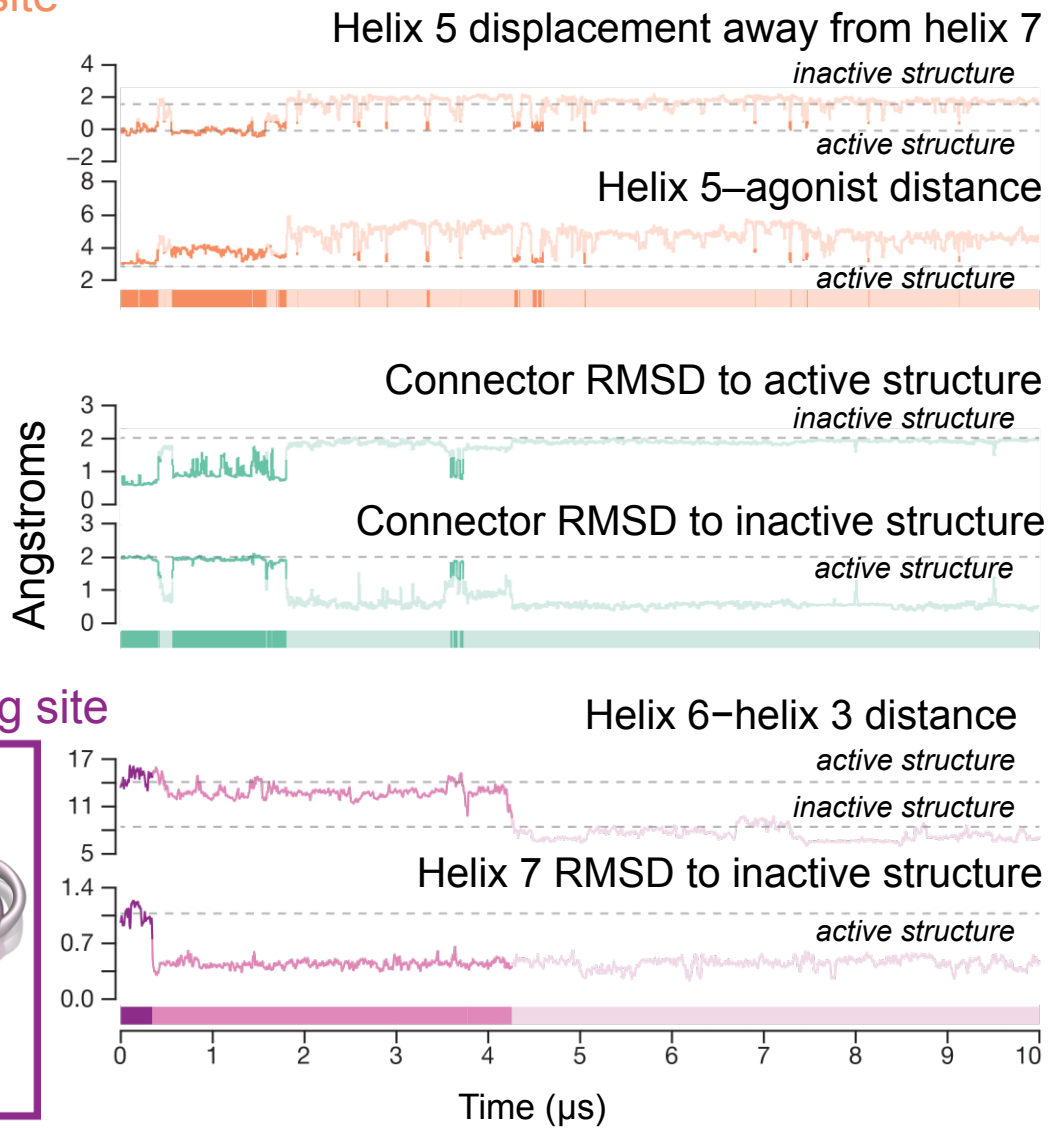


b Connector



c G protein–binding site

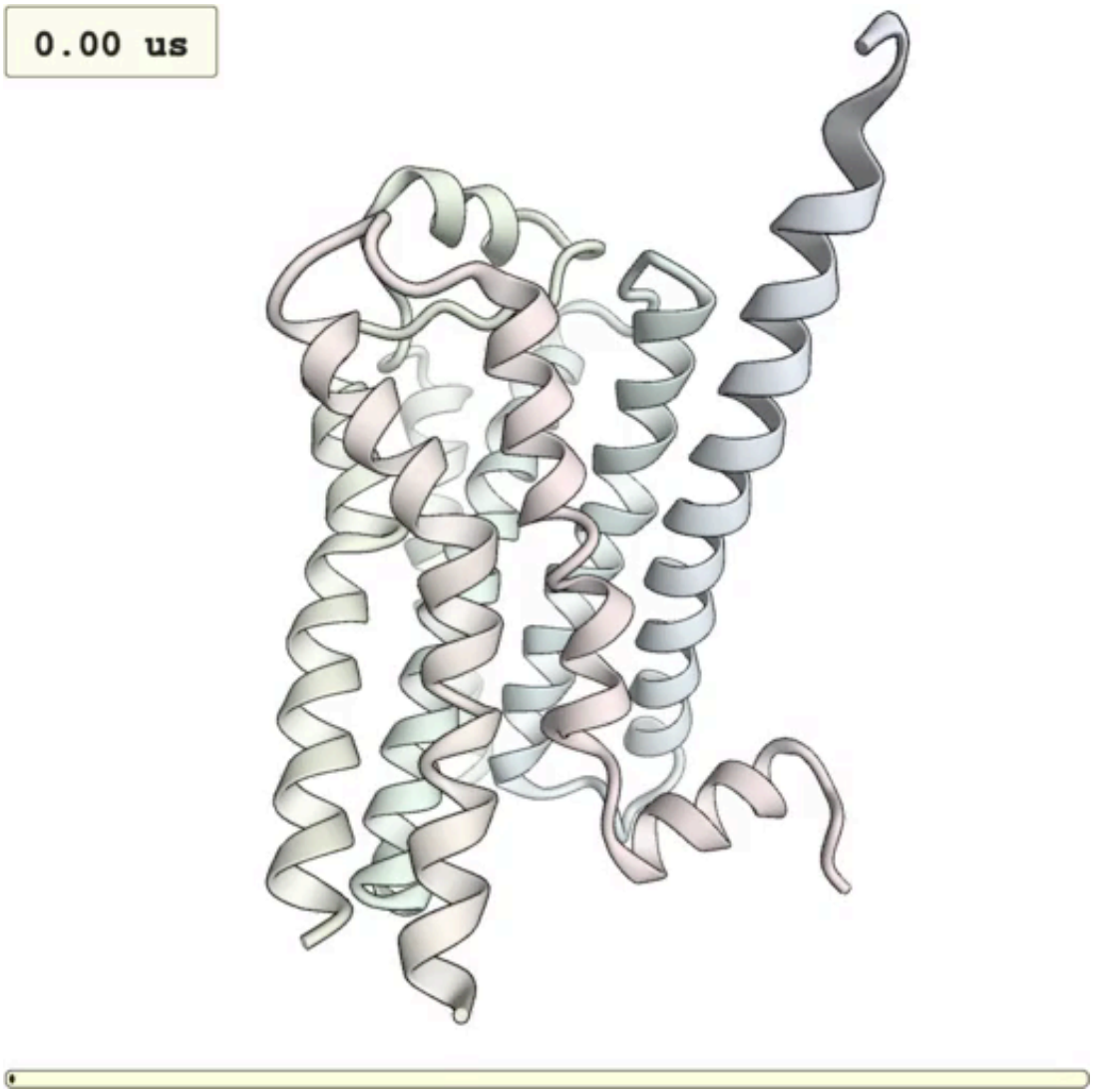


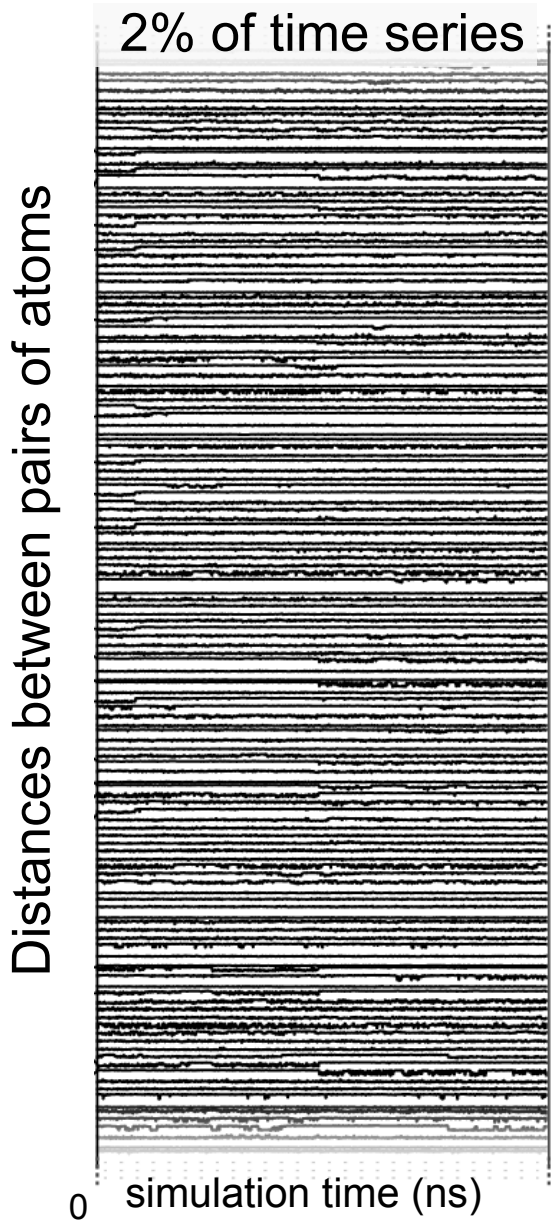


What are the “important events”?

- They usually involve conformational (that is, structural) changes
- These changes can be subtle: they might involve only a very small part of the protein
- The protein is moving *constantly*
- We tend to care most about *rare* changes

Can we approach this as a changepoint detection problem?





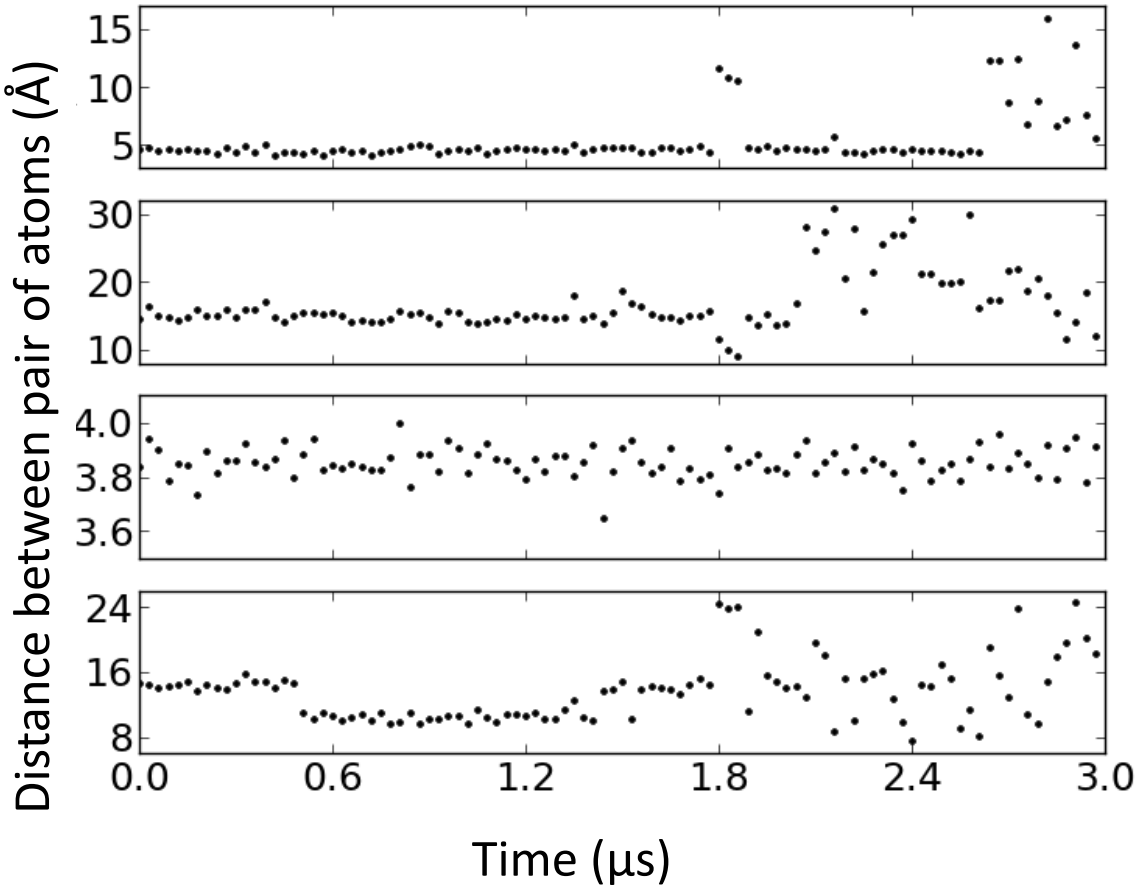
Challenge: Each change affects only a few of the thousands of time series that describe a simulation

Fan et al., *PNAS*, 2015

Detection of simultaneous change points

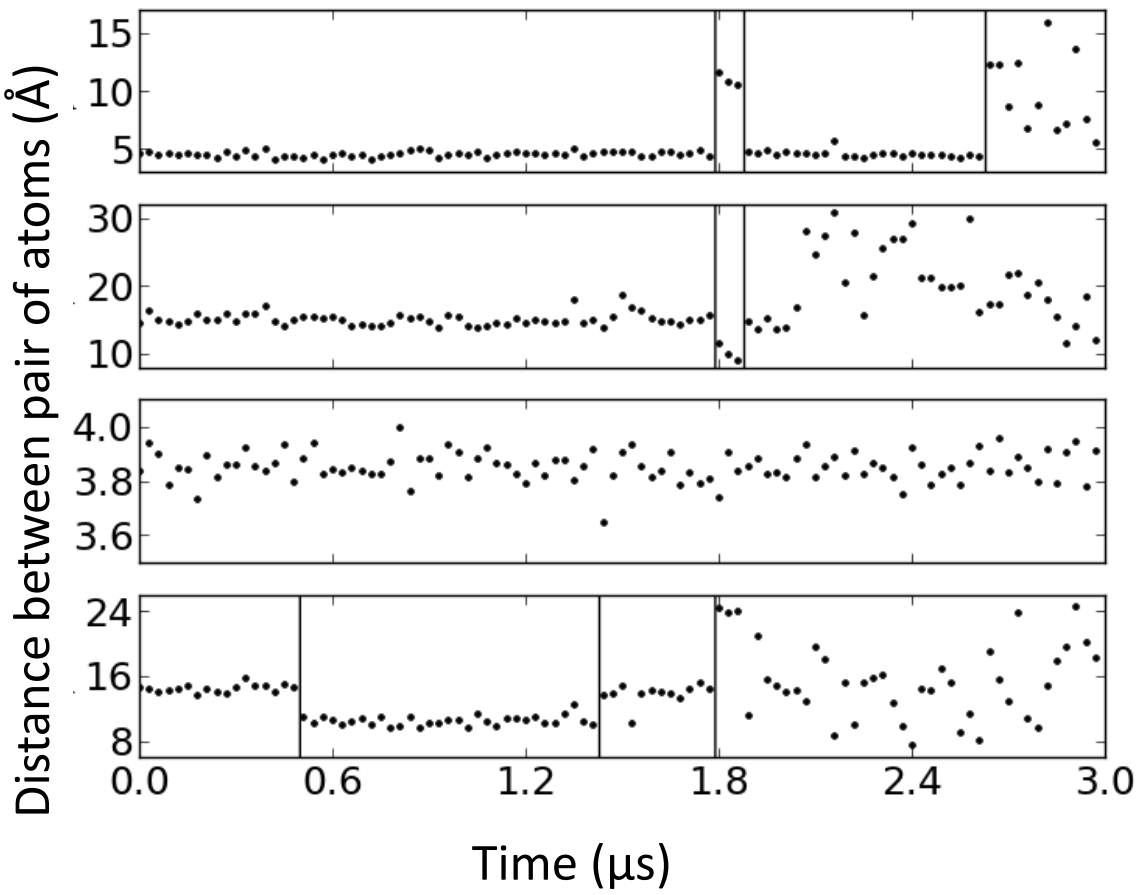
- Approach this as a statistical change-point detection problem, but:
  - Determine both change times and the observables that change at each time
  - Search, in particular, for simultaneous changes of multiple observables
- We can formulate this as a giant optimization problem
  - We can solve this problem efficiently by iterative application of recently introduced dynamic programming algorithms

Simultaneous changepoint detection: A simple example



Four sample time series (of hundreds/thousands), each corresponding to the distance between a pair of protein atoms in a simulation

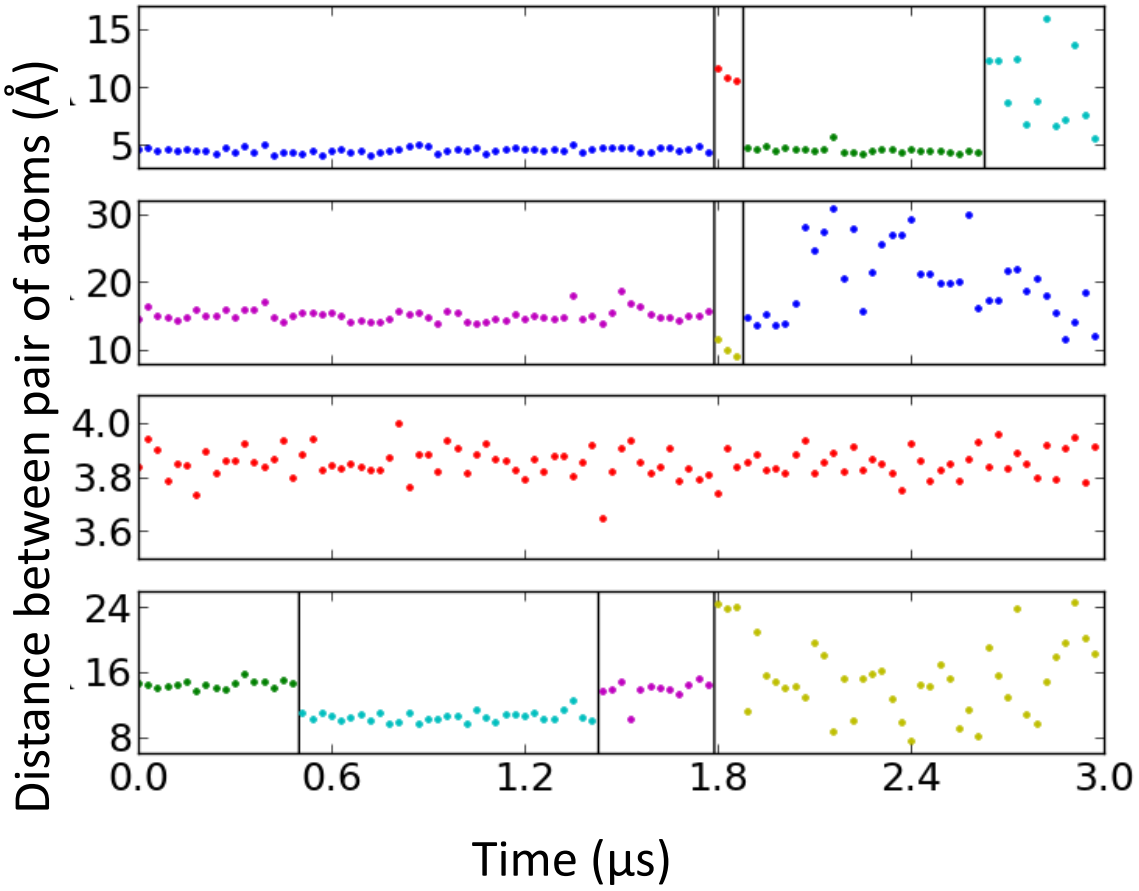
Simultaneous changepoint detection: A simple example



Wish to choose a small set of changepoints such that each observable has constant statistics between changepoints.

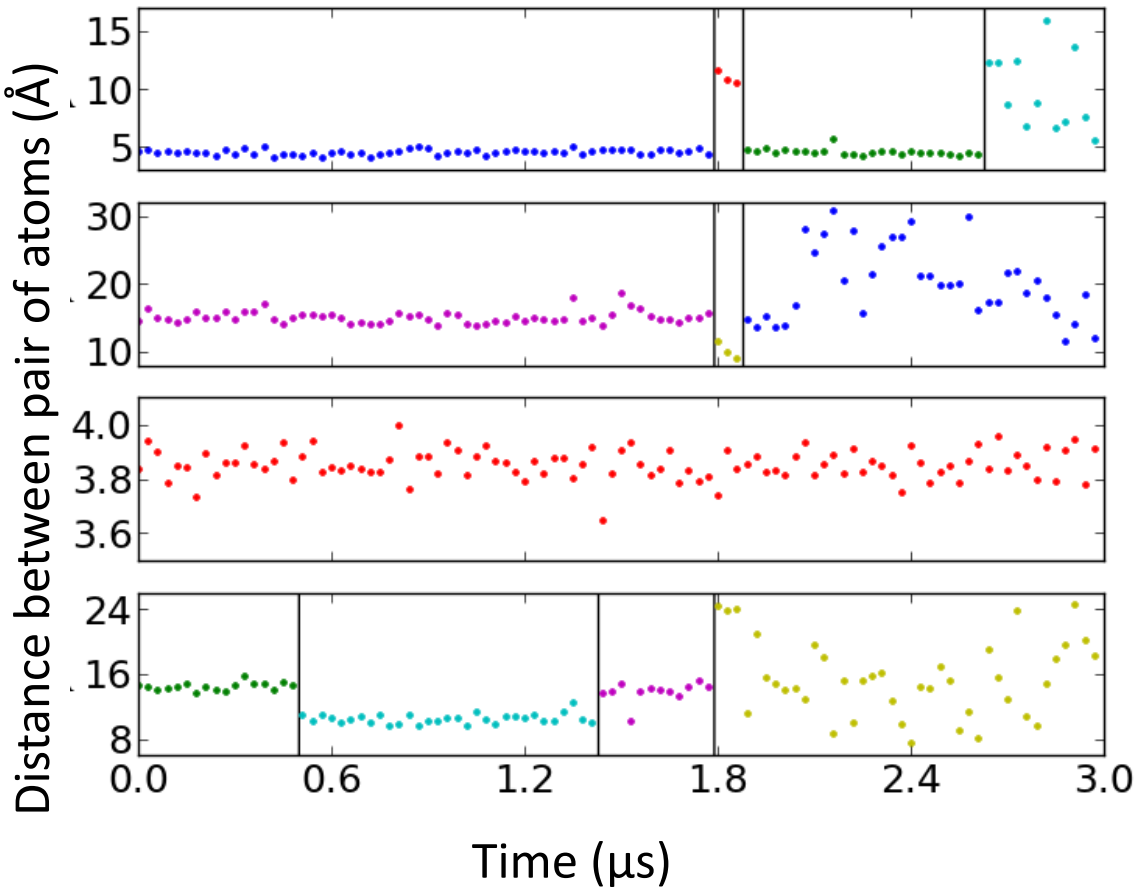
Exploit the fact that changes in different observables are likely to occur simultaneously, especially if the atoms involved are nearby.

Our approach



Solve a big optimization problem to determine when changes occur, which observables they affect, and how the statistics of those observables change (i.e., what parts of the protein change at what times, and how).

Our approach



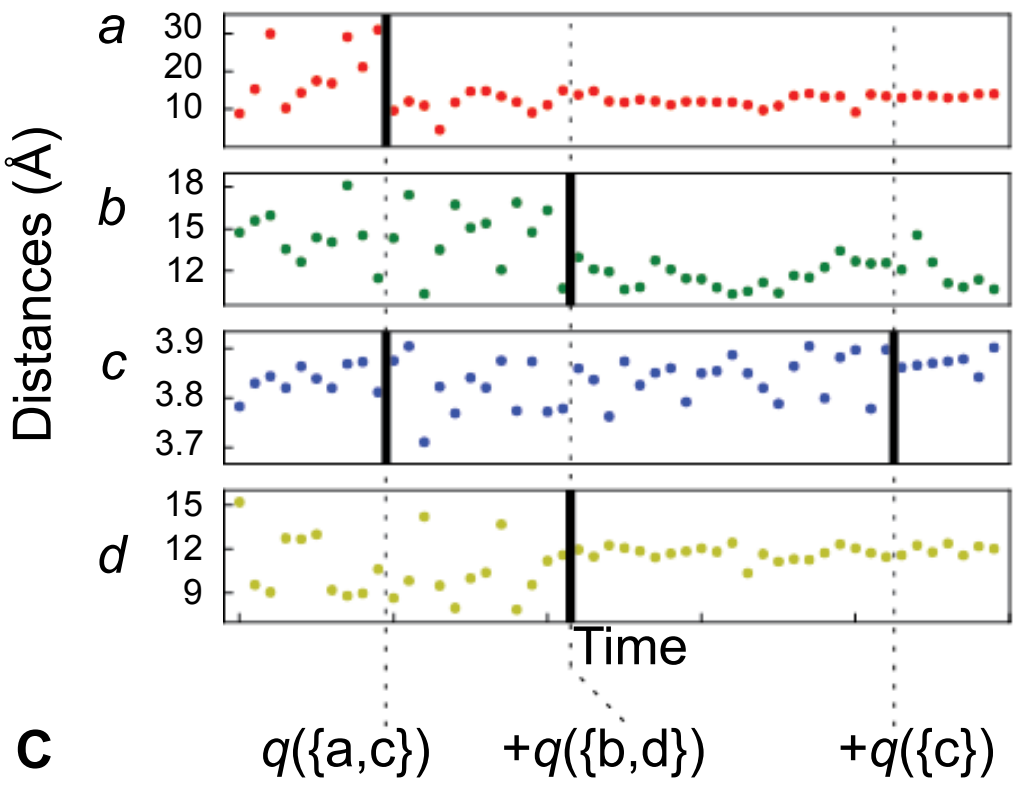
Choose changepoints, and model parameters for each segment, to maximize:

[Likelihood of data given model] – [Penalty function]

where the penalty function increases with the number of changepoints, but less so if multiple changepoints are simultaneous (especially if the atoms involved are nearby one another).

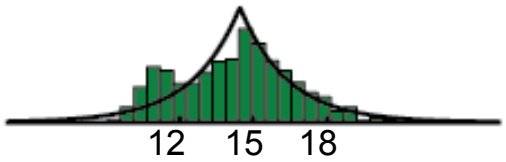
Our approach: Simultaneous penalized likelihood estimation (SIMPLE) change point detection

A Candidate change points



B Data likelihood given change points

$$\hat{l}(\;\;)+\hat{l}(\;\;)+\hat{l}(\;\;)+\ldots$$



$\hat{l}$ = log likelihood under best-fit model

C

$q(\{a,c\})$ $+q(\{b,d\})$ $+q(\{c\})$

Change-point penalties

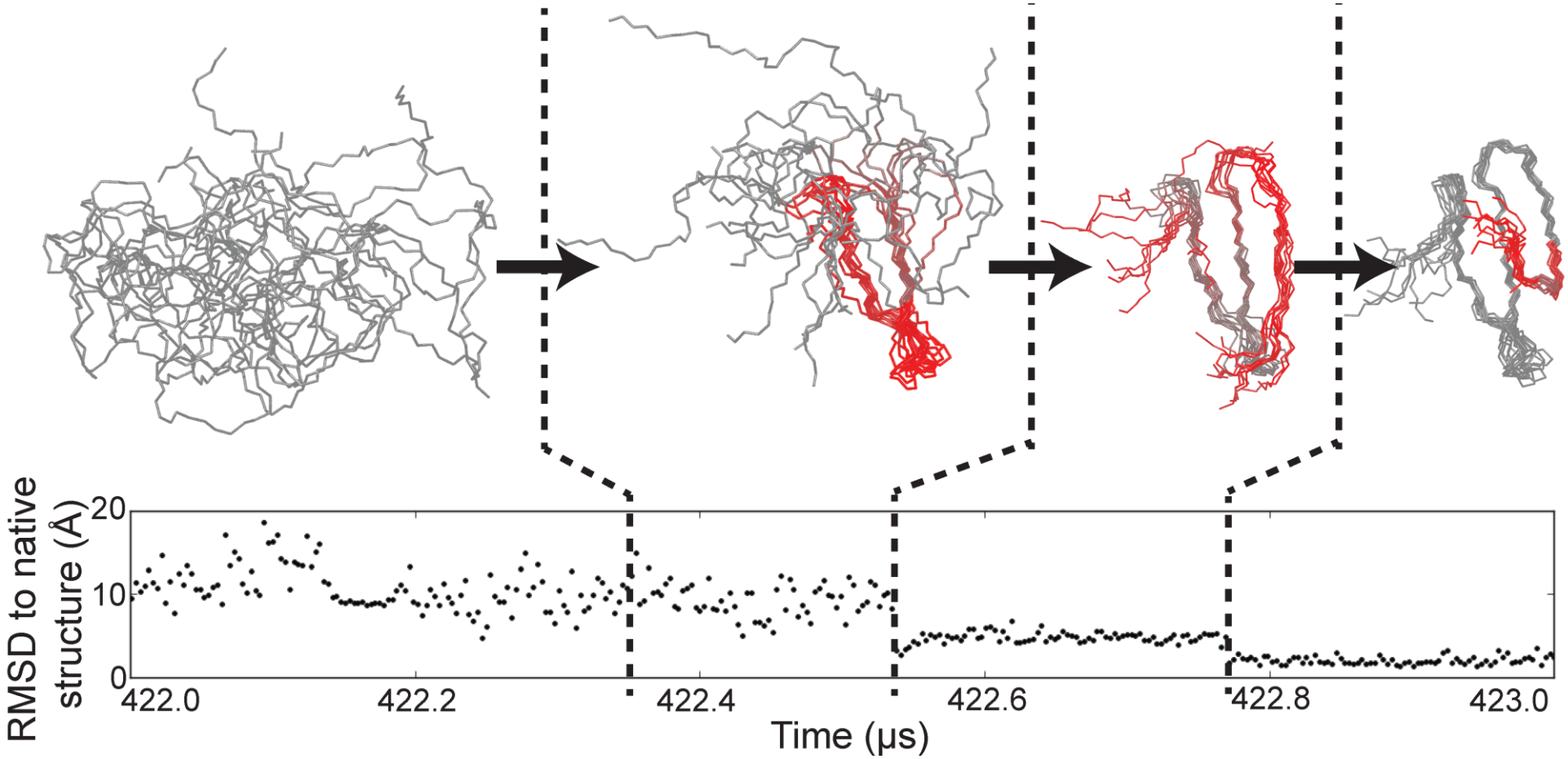
D

$$\sum \hat{l} - \lambda \sum q$$

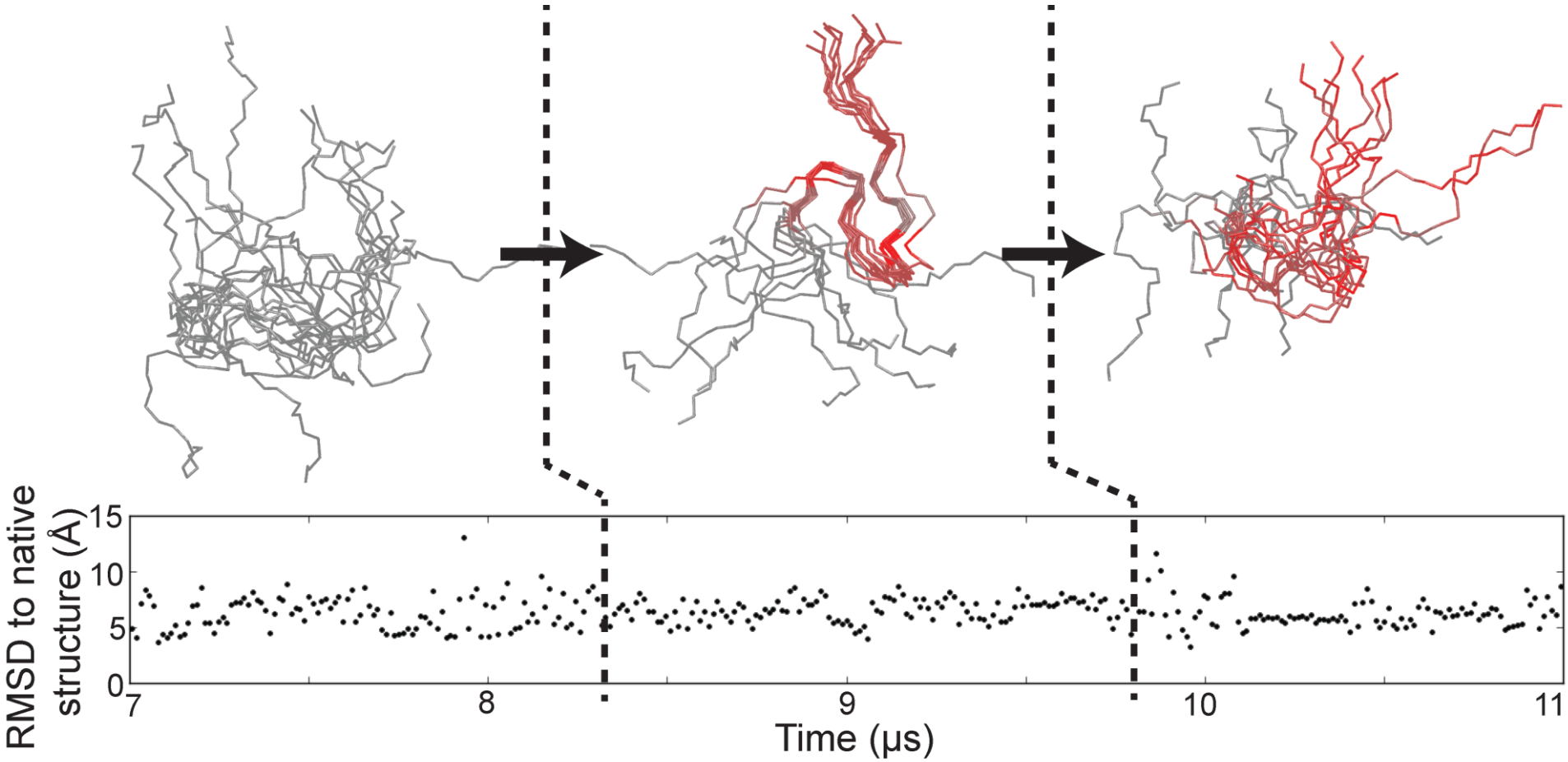
Objective function to maximize

Penalty increases with number of changes, but less so if changes are simultaneous

Detected change points in WW domain folding simulation

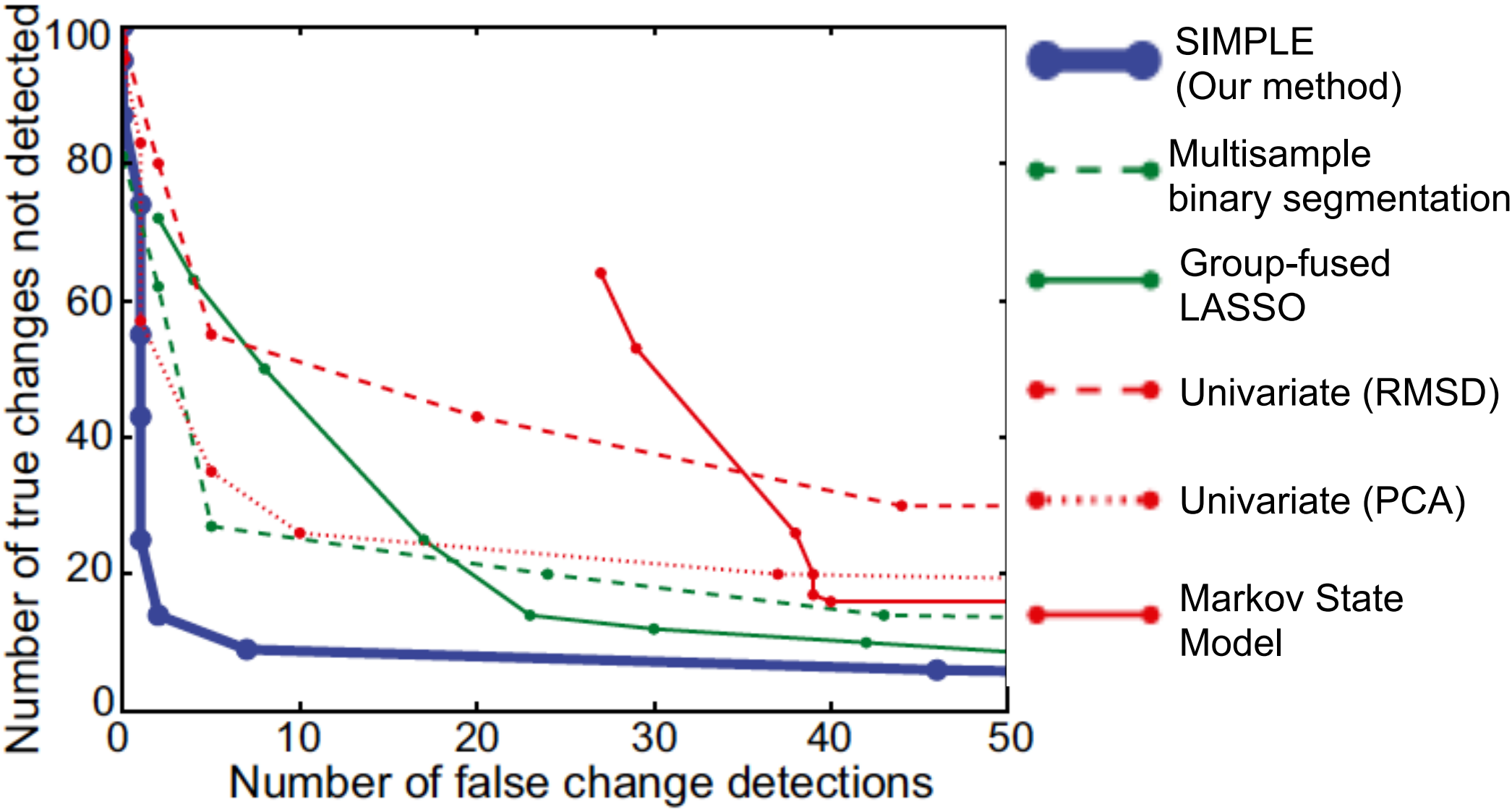


Detected change points in simulation of unfolded Trp cage



Performance comparison

(on synthetic data for which we know the true change times)



Limitations of the study

- Basic problem definition: is one always looking for “sudden” changes of this sort?
- No guarantee that optimization algorithm will converge to global optimum
- Lack of good software for visualizing the results
  - We actually developed (and released) such software, but its portability is poor
- Although the paper describes a general method, it doesn’t demonstrate application of the method to other types of data

Supplemental slides

Problem formulation as optimization

- Laplace likelihood $p(x|\mu, v) = \frac{1}{2v} e^{-\frac{|x-\mu|}{v}}$
  - Detect changes in median and mean absolute deviation
  - More robust than Gaussian likelihood
  - Computationally tractable (pay logarithmic factor)
  - Invariant to location and scale of data
- Sub-additive penalty: $q(S_1 \sqcup S_2) < q(S_1) + q(S_2)$
  - Preference for grouping changes into shared change times
  - Couples time series into a single optimization problem

Theoretical consistency guarantee

In the limit of increasing amount of data between changes,

$$\mathrm{P}(global\ max\ gives\ correct\ changes) \to 1,$$

assuming:

- Medians and/or mean absolute deviations of data distributions change
- Data distributions have sub-exponential tails
- Sub-additive penalty function q
- Penalty function scales as $\mathrm{O}(\log T^2) < q < O(T)$

Algorithmic approach

- Algorithm for maximizing objective function involves:
  - Iterative refinement of change times and changes per time series
  - Pruned dynamic programming algorithm for single time series (Killick, Fearnhead, Eckley 2012)
- Typical-case runtime $O(JT \log T)$ for J time series, T frames per time series
- Parallelizable across the J time series (garden implementation uses MPI)