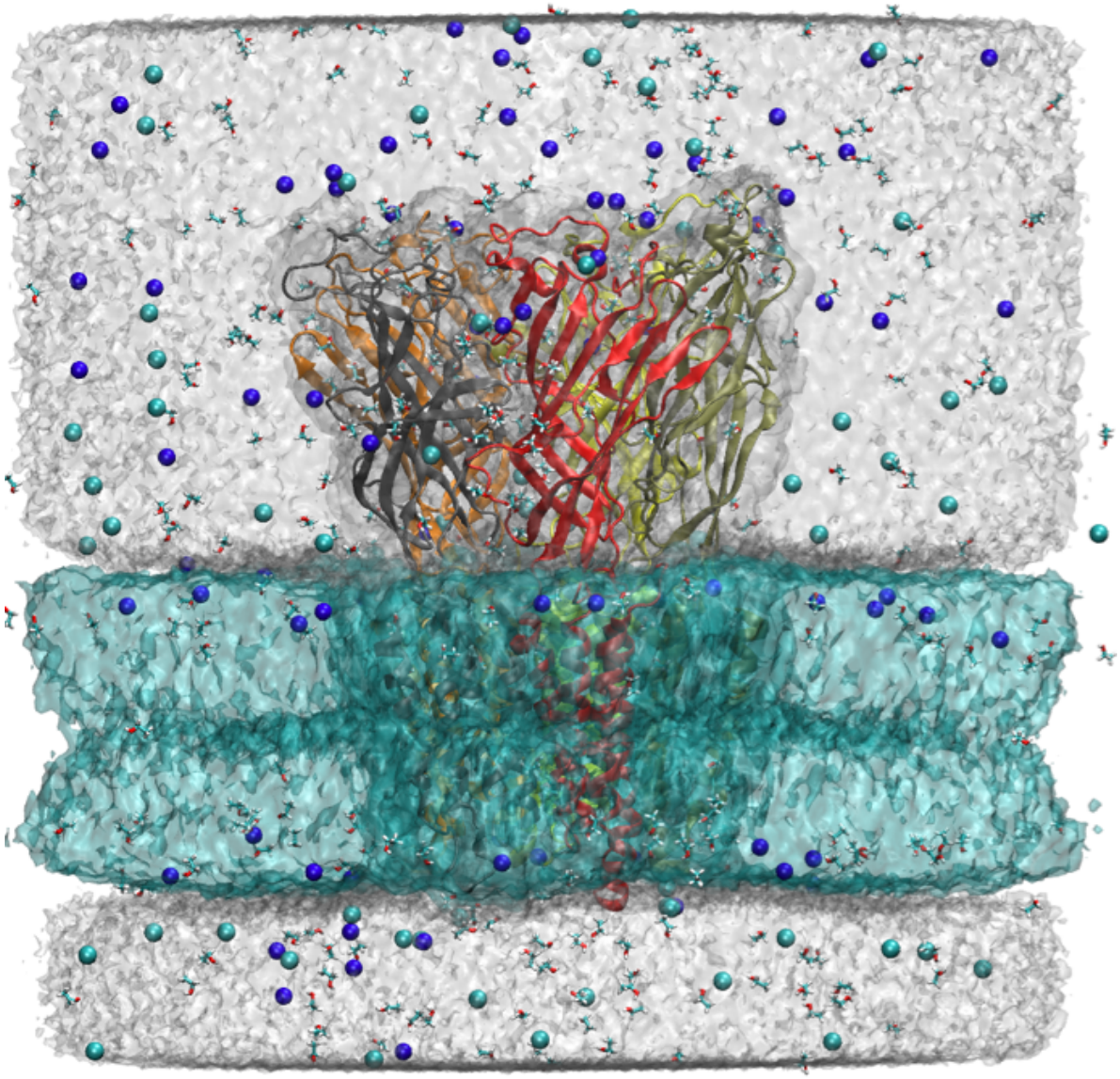
Replica exchange molecular dynamics

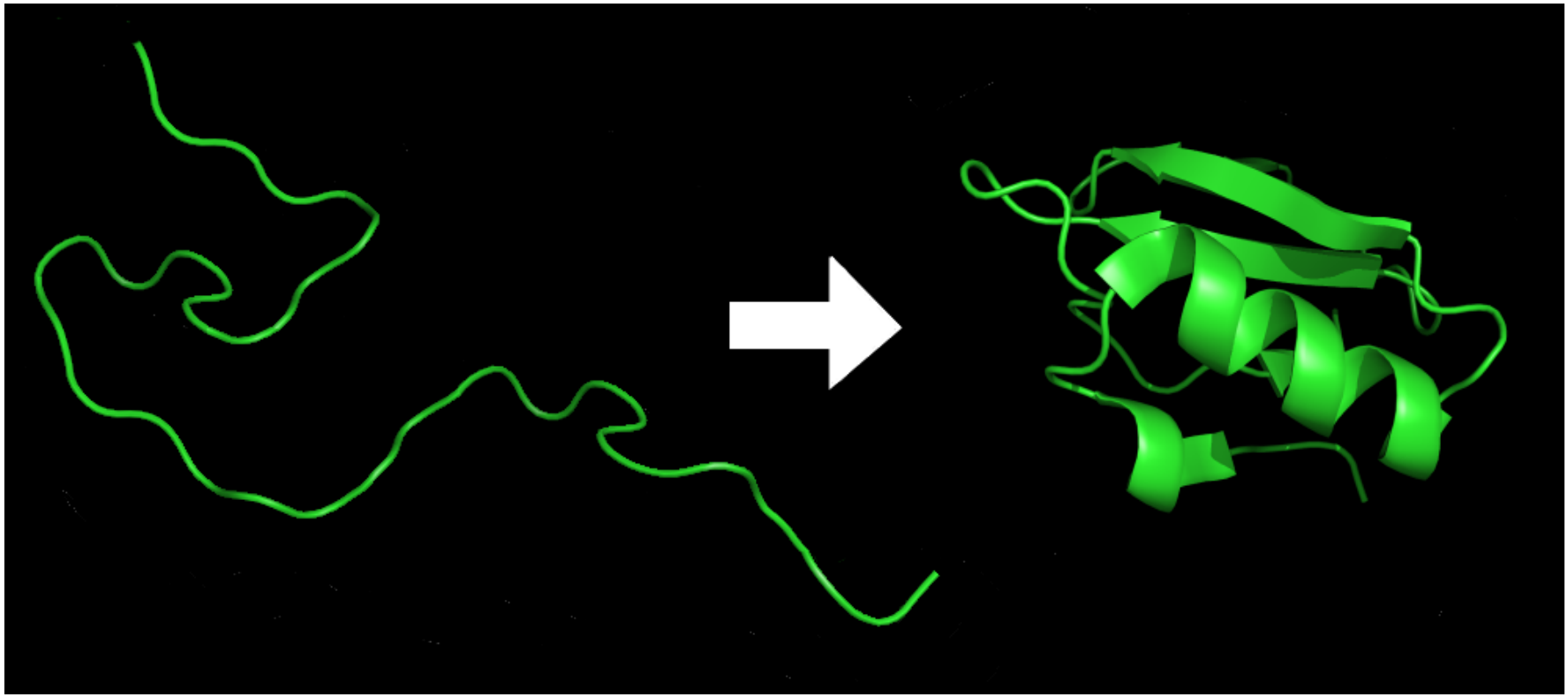
Advanced molecular dynamics course, KTH

Dr Mark Abraham (mjab@kth.se)

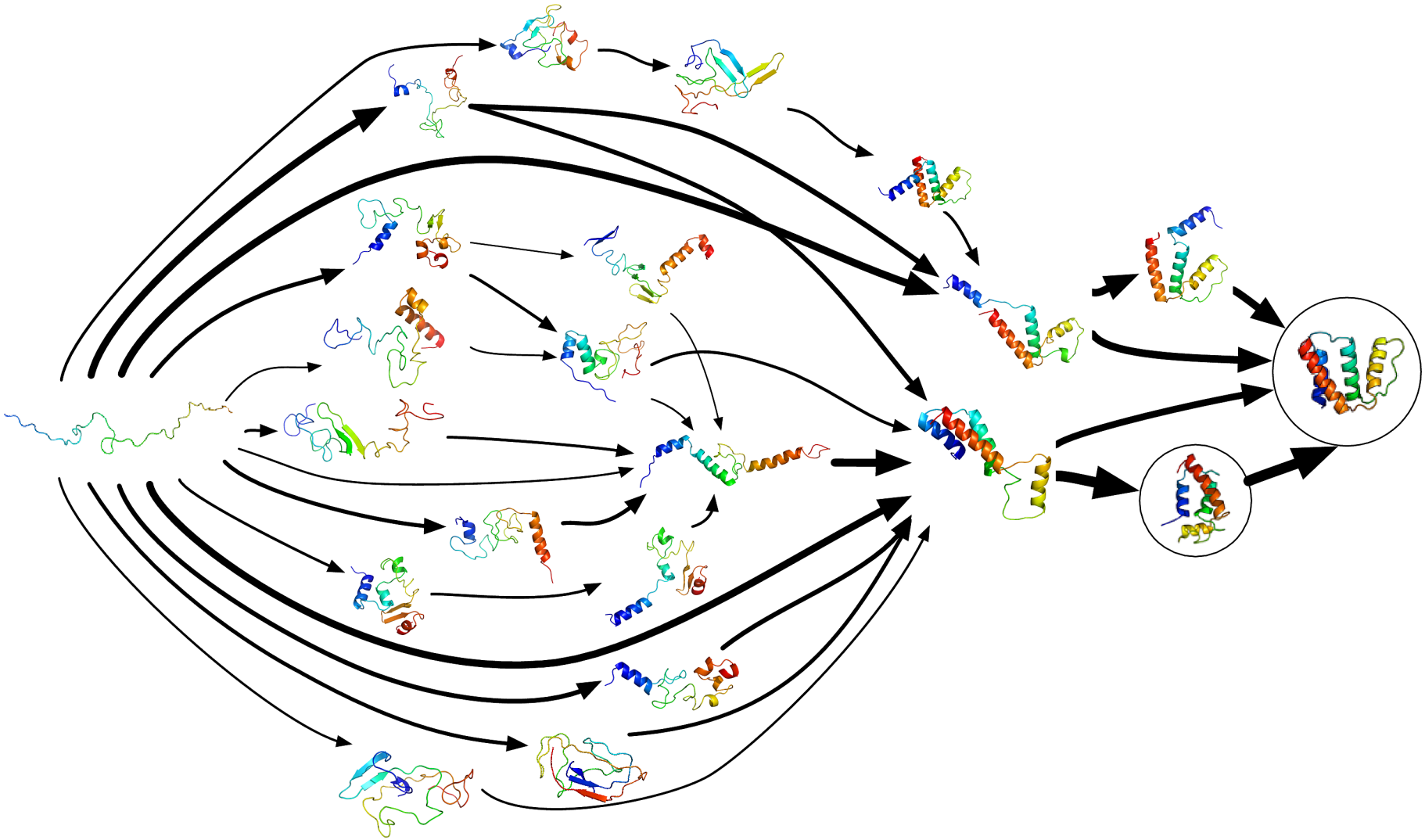
Ion channels



Protein folding

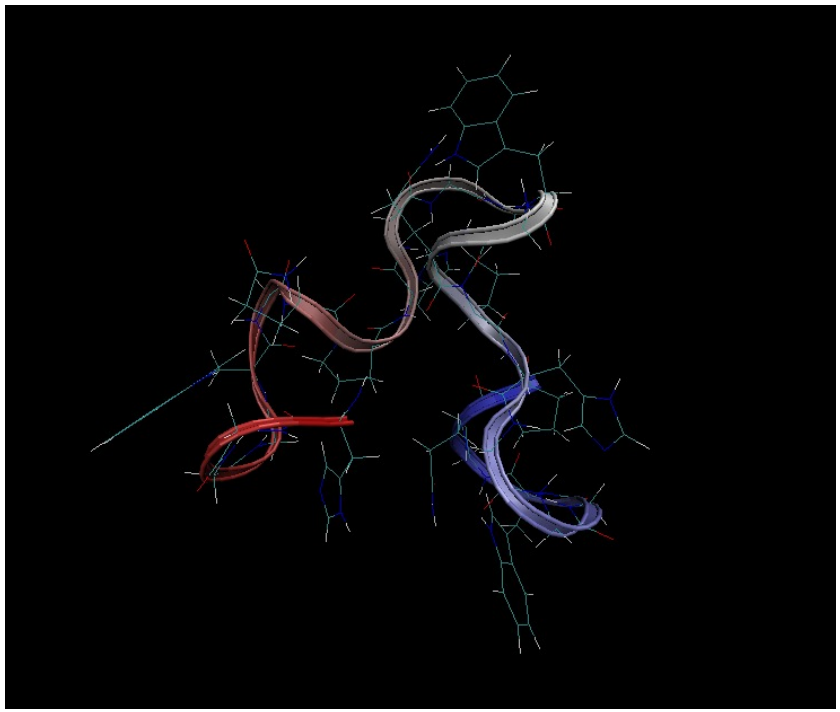


Markov state models

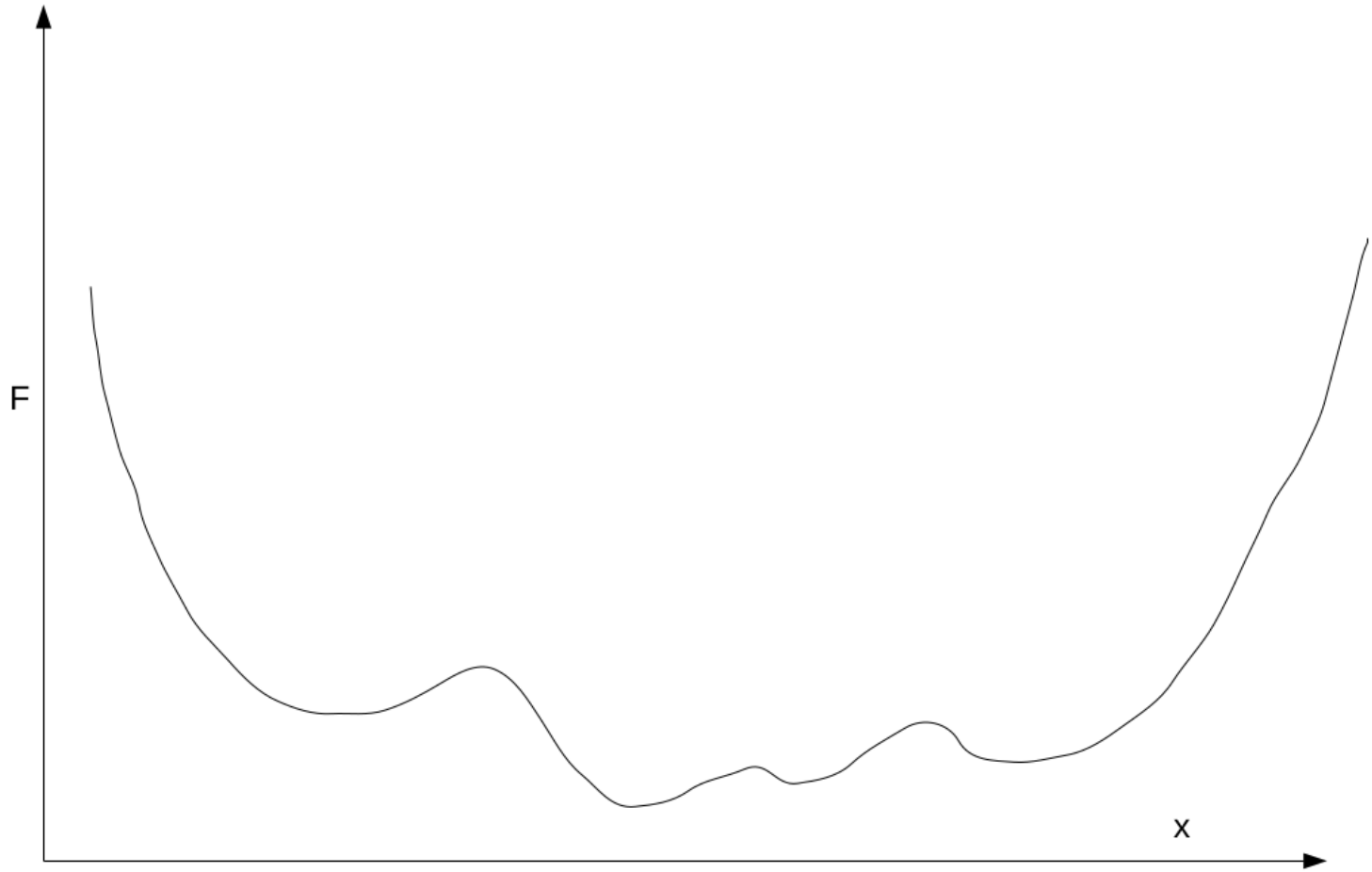


Sampling is often frustrated

- many different motions (bonds, angles, side chains, secondary structure deformation)
- different motions have different time scales
- difficult to parameterize a model that gets it all right
- more difficult to sample from it afterwards

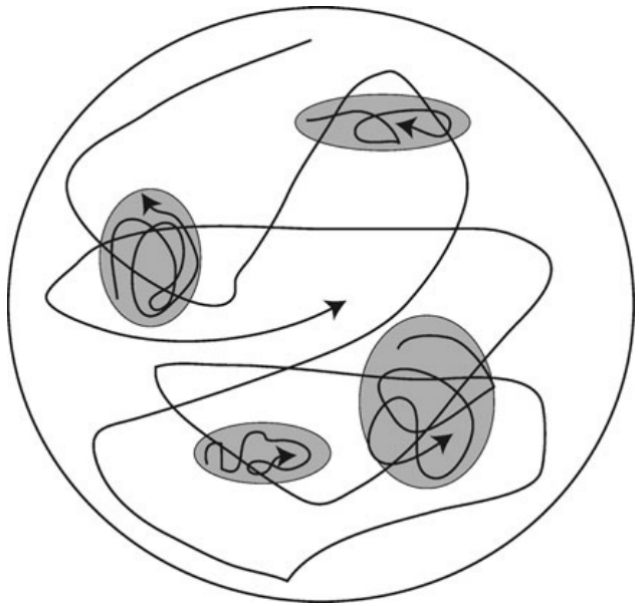


Barriers in MD



Frustration from barriers

- barriers of more than a few kT exist, and are hard to cross
- need extremely large amount of brute-force sampling to get over them
- makes solving problems like protein folding exceedingly computationally expensive



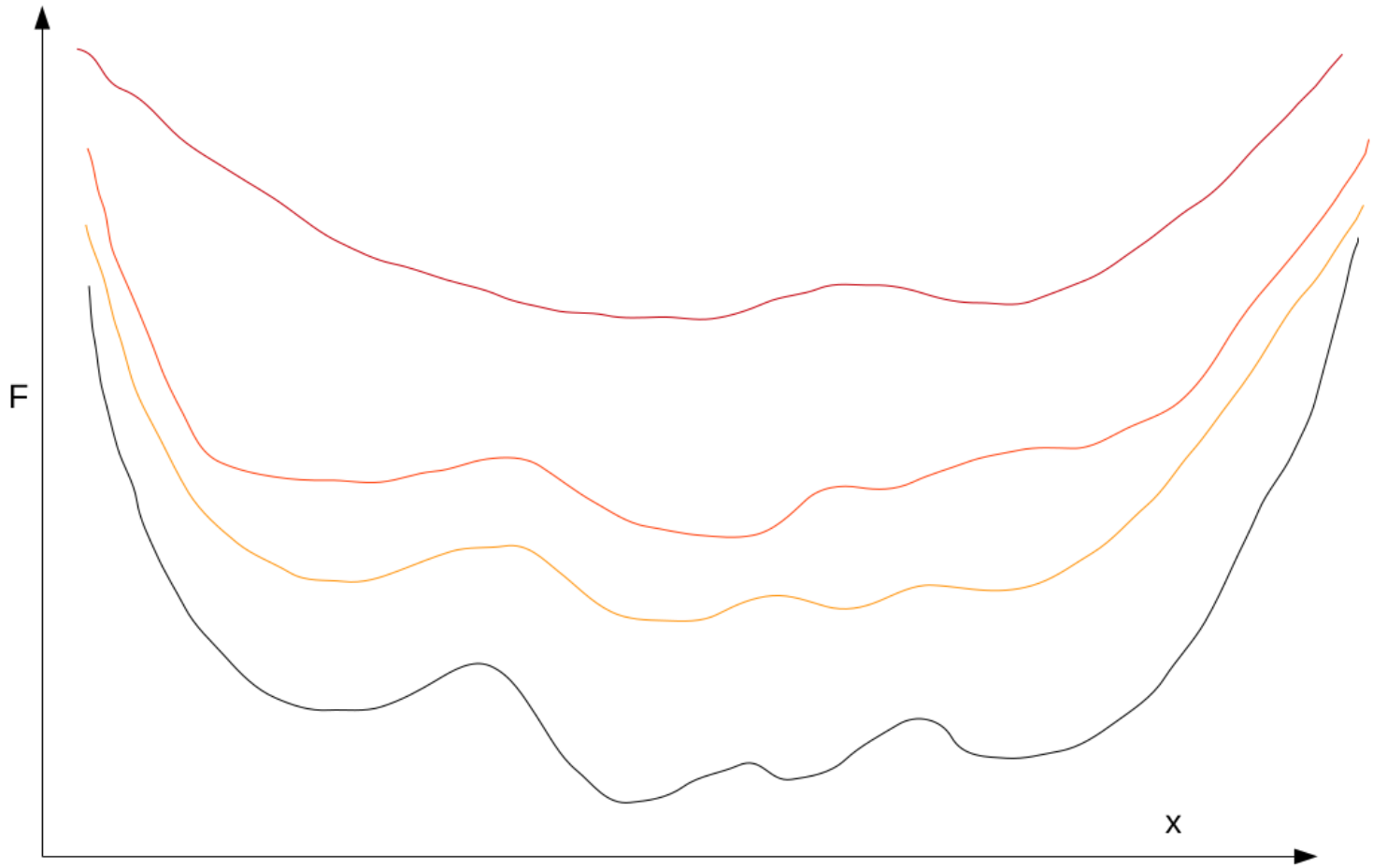
Ways to grapple with the problem

- give up on the fine detail, and use a coarse-graining approach (next Tuesday's topic)
- accelerate the sampling (work smarter! Today's topic)
- throw more hardware at it (e.g. Folding@Home)
- write faster software (hard, very hard; also next Tuesday's topic)

Accelerating the sampling

- if the problem is that kT is too small...
  1. increase T
  2. sample widely
  3. ...
  4. profit!
- unless the landscape changes.... (gulp)

Landscapes change with temperature



Simulated tempering

- a Monte Carlo approach to permit system to move in the space of a "control parameter"
- typically that is temperature
- only collect data when the system returns to the parameter value of interest
- this is correct if the (Metropolis) exchange criterion is correctly constructed

For a state s,

$$P((\beta, s) \rightarrow (\beta', s)) = \min(1, \frac{w(\beta', s)}{w(\beta, s)})$$

where $\beta = \frac{1}{kT}$ and $w(\beta, s) = \exp[-\beta U(s) + g(\beta)]$

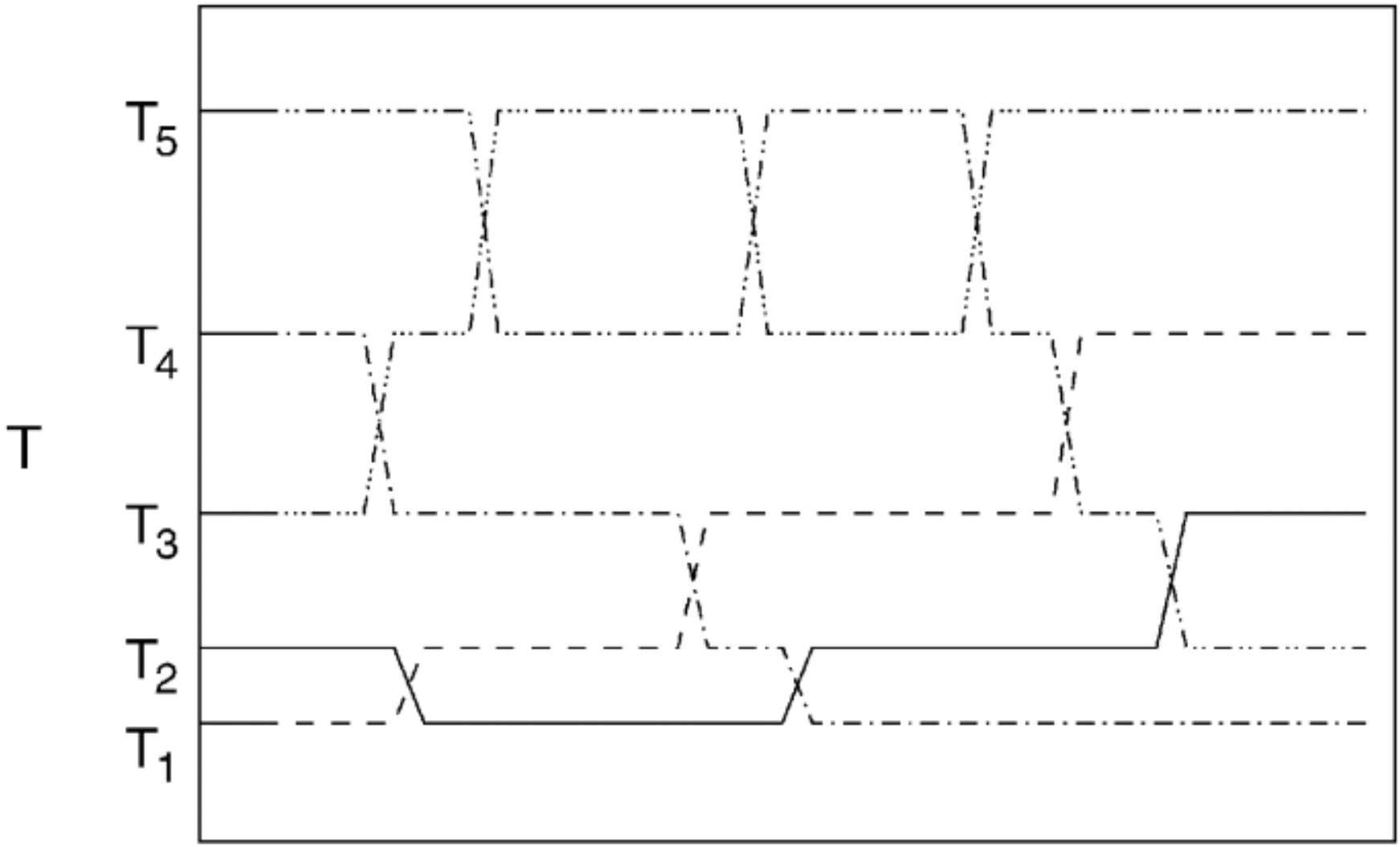
Simulated tempering (2)

- correct if the exchange criterion is constructed correctly
  - the optimal $g(\beta)$ is the free energy...
  - so you're good if you already know the relative likelihood of each conformation at each temperature...
- works great if you already know the answer to a harder problem than the original
- (but you can use an iterative scheme to converge on the answer)

Parallel tempering (a.k.a. replica exchange)

- side-steps the prior-knowledge problem by running an independent copy of the simulation at each control parameter
- (note, throwing more hardware at the problem!)
- now the exchange is between copies at different control parameters, each of which is known to be sampled from a correct ensemble already
- this eliminates $g(\beta)$ from the generalized exchange criterion...

Parallel tempering



Rescaling the momenta

- when proposing an exchange, can do anything to any coordinate
- accept exchange only when detailed balance is preserved
- it is convenient for the average KE after exchanges to be consistent with the target ensemble
- so rescale the momenta as

$$p_i^{\text{new}} = \sqrt{\frac{T^{\text{old}}}{T^{\text{new}}}}\, p_i^{\text{old}}$$

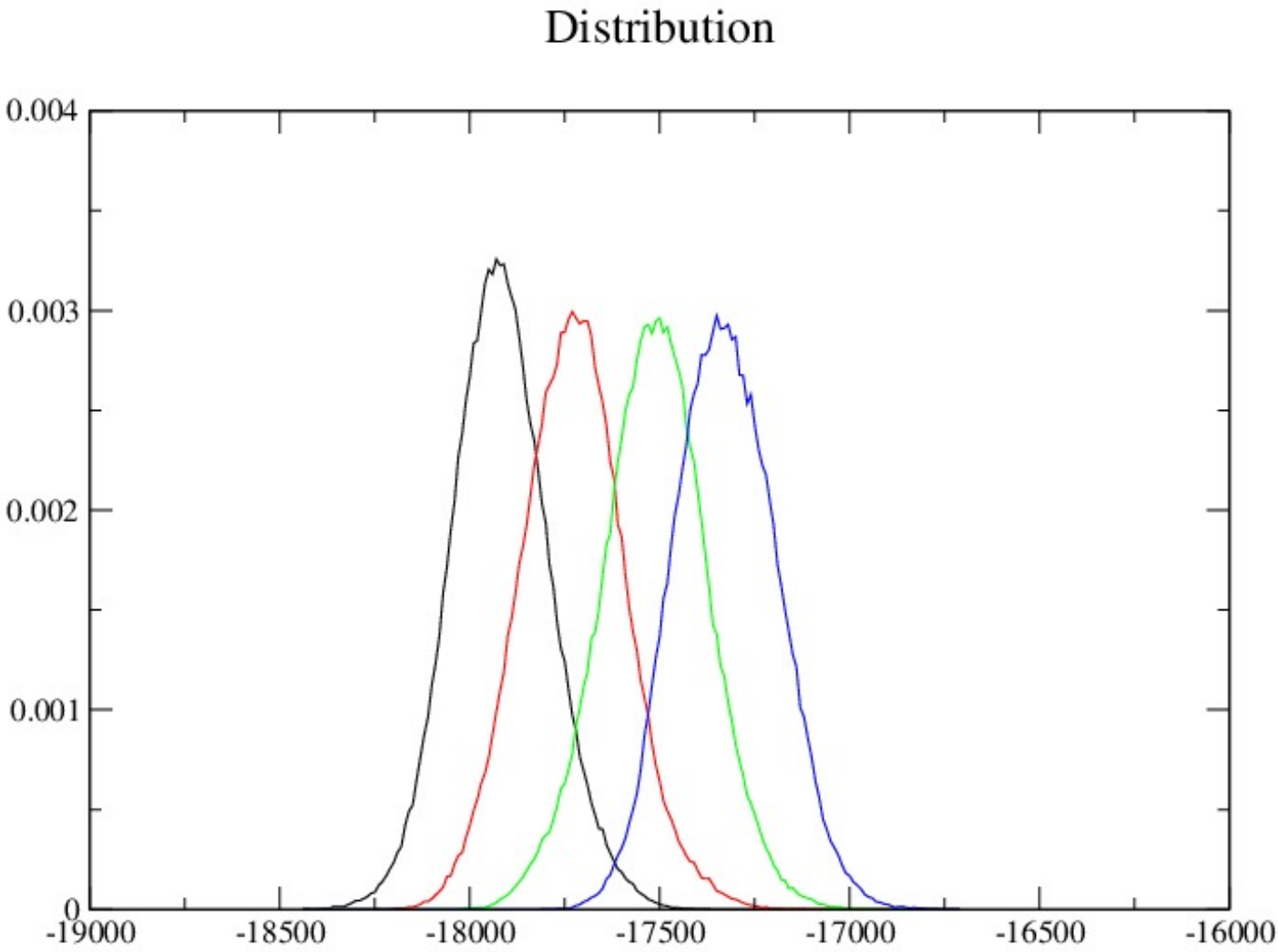
Parallel tempering - the exchange criterion

$$P((\beta, s) \leftrightarrow (\beta', s')) = \min(1, \frac{w(\beta, s')w(\beta', s)}{w(\beta, s)w(\beta' s')})$$

For Boltzmann weights, this reduces to

$$P((\beta, s) \leftrightarrow (\beta', s')) = \min(1, \exp[(\beta' - \beta)(U(s') - U(s))])$$

Parallel tempering - understanding the exchanges

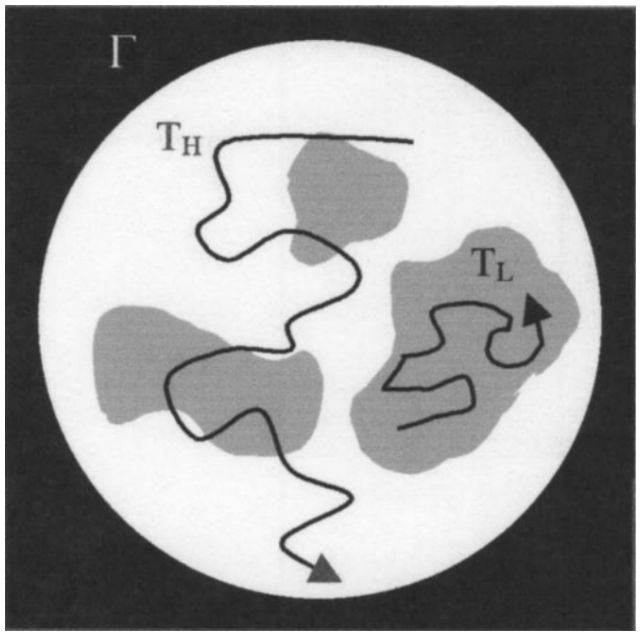


Is this real?

- recall that $P(\beta, s) \propto \exp[-\beta U(s)]$
- any scheme that satisfies detailed balance forms a Markov chain whose stationary distribution is the target (generalized) ensemble
- so we require only that
  $P(\beta, s)P((\beta, s) \to (\beta', s)) = P(\beta', s')P((\beta', s') \to (\beta, s'))$
- which is what was constructed!
- However, dynamical information is lost when exchanges happen

Might this work?

- high-temperature replicas hopefully can cross barriers
- if the conformations they sample are representative of lower-temperature behaviour, then they will be able to exchange down
- if not, they won't



Ensembles commonly used

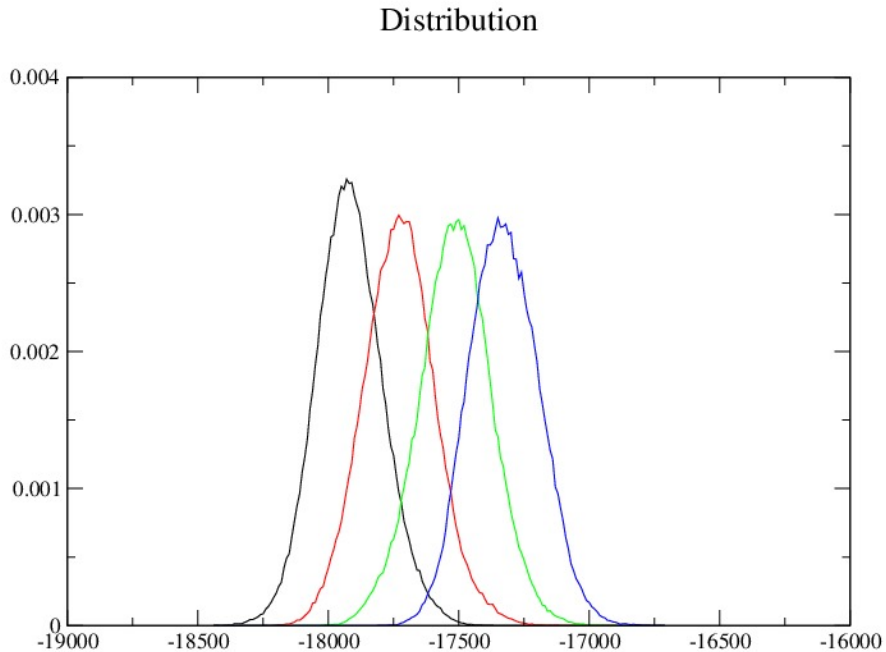
- natural to use the NVT ensemble with an increasing range of T and constant V
- remember that we must rescale the velocities to suit the new ensemble in order to construct the above exchange criterion
- probably this should use a velocity-Verlet integrator (x and v at same time)
- in principle, can use other ensembles like NPT

Ensembles commonly used

- NVT at constant volume must increase P with T
- that seems unphysical
- worse, the force fields are parameterized for a fixed temperature
- but the method doesn't require that the ensembles correspond to physical ones
- merely need overlap of energy distribution
- the size of the overlap determines the probability of accepting an exchange

Problems with replica exchange

- molecular simulations typically need lots of water
- thus lots of degrees of freedom
- energy of the system grows linearly with system size
- width of energy distributions grow as $\sqrt{size}$
- need either more replicas or accept lower overlap



Unphysics is liberating

- if there's no need to be physical, then may as well be explicit about it
- large number of proposed schemes

Example: resolution exchange

- run replicas at different scales of coarse graining
- at exchange attempts, not only rescale velocities, but reconstruct the coordinates at higher/lower grain level

Hamiltonian replica exchange

- T isn't the only possible control parameter
- could gradually turn on a restraint or biasing potential
- control parameters can be multi-dimensional, e.g. in a free-energy calculation, could change both alchemical transformation parameter λ and T

Replica exchange with solute tempering (REST)

- selectively "heat" only a small region of the system
- modify the parameters to scale the energy, rather than heating (recall that $P(\beta, s) \propto \exp[-\beta U(s)]$)
- advantage that the energy distribution of only part of the system increases over control parameter space
- needs many fewer replicas for given control parameter space
- implemented in many MD packages, including GROMACS, by PLUMED plugin (https://www.plumed.org/)

Choices in molecular dynamics studies

- Solvation model
- Resolution of model physics
- Force field
- Statistical ensemble to sample
- Starting condition(s)
- Simulation time step
- Observables
- Data collection rate

Additional choices in replica exchange studies

- Which control parameter? (T, λ)
- At which control parameters to collect data
- Range of control parameter space
- Number of replicas
- Spacing of replicas
- Exchange probability
- Exchange attempt interval

Shameless plug: https://dx.doi.org/10.1021/ct800016r

Average Exchange probability

Recall

$$P((\beta, s) \leftrightarrow (\beta', s')) = \min(1, \exp[(\beta' - \beta)(U(s') - U(s))])$$

So

$$P_{\text{ave}}((\beta, s) \leftrightarrow (\beta', s')) = \int \int \min(1, \exp[(\beta' - \beta)(U_1(s') - U_2(s))])\, dU_1\, dU$$

Generally, you want replicas whose temperatures increase roughly exponentially

Web server for helping choose T for REMD

http://folding.bmc.uu.se/remd-temperature-generator/

Based on https://dx.doi.org/10.1039/B716554D

Interval between exchange attempts

- Ideally, after MD step, attempt exchange
- Doesn't really matter if the exchange probability is low, you'll get some exchanges eventually
- Does this spamming help?

Interval between exchange attempts

- observables like potential energy have autocorrelation times
- for e.g. protein in water, it's about 1 ps
- if you exchange more frequently than that, you get back exchanges https://dx.doi.org/10.1063/1.2404954
- so either estimate or measure the autocorrelation time, and exchange about that often

Practical replica exchange in GROMACS

- uses the multi-simulation feature which requires building with an MPI library
- need enough resources for each simulation on its own
- put each simulation in a unique directory, using the multidir feature
- equilibrate there
- choose the number of simulation steps between exchange attempts on the command line
- use the most recent versions of GROMACS for minimal communication between simulations

De-multiplexing

- some MD packages write a continuous trajectory of each simulation system
- others (including GROMACS) write a continuous ensemble
- demux.pl script in the GROMACS installation will convert the trajectory files between the two, based on the exchange information in the log file

Questions?

Tutorial

Run the online Jupyter notebook via a binder at https://mybinder.org/v2/gl/gromacs%%2Fonline-tutorials%%2Fsimple-remd/main or the "Default binder" link from https://gitlab.com/gromacs/online-tutorials/simple-remd/-/tree/main.

You can access the git repository containing the materials via

```
git clone https://gitlab.com/gromacs/online-tutoria
```

Or download as as zip archive from https://gitlab.com/gromacs/online-tutorials/simple-remd/-/archive/main/simple-remd-main.zip.

Optional

Build MPI-enabled GROMACS. Get e.g. openmpi or mpich2 packages for your distro. Then configure GROMACS as normal, but add the following flag to the CMake line:

```
cmake -DGMX_MPI=on
```