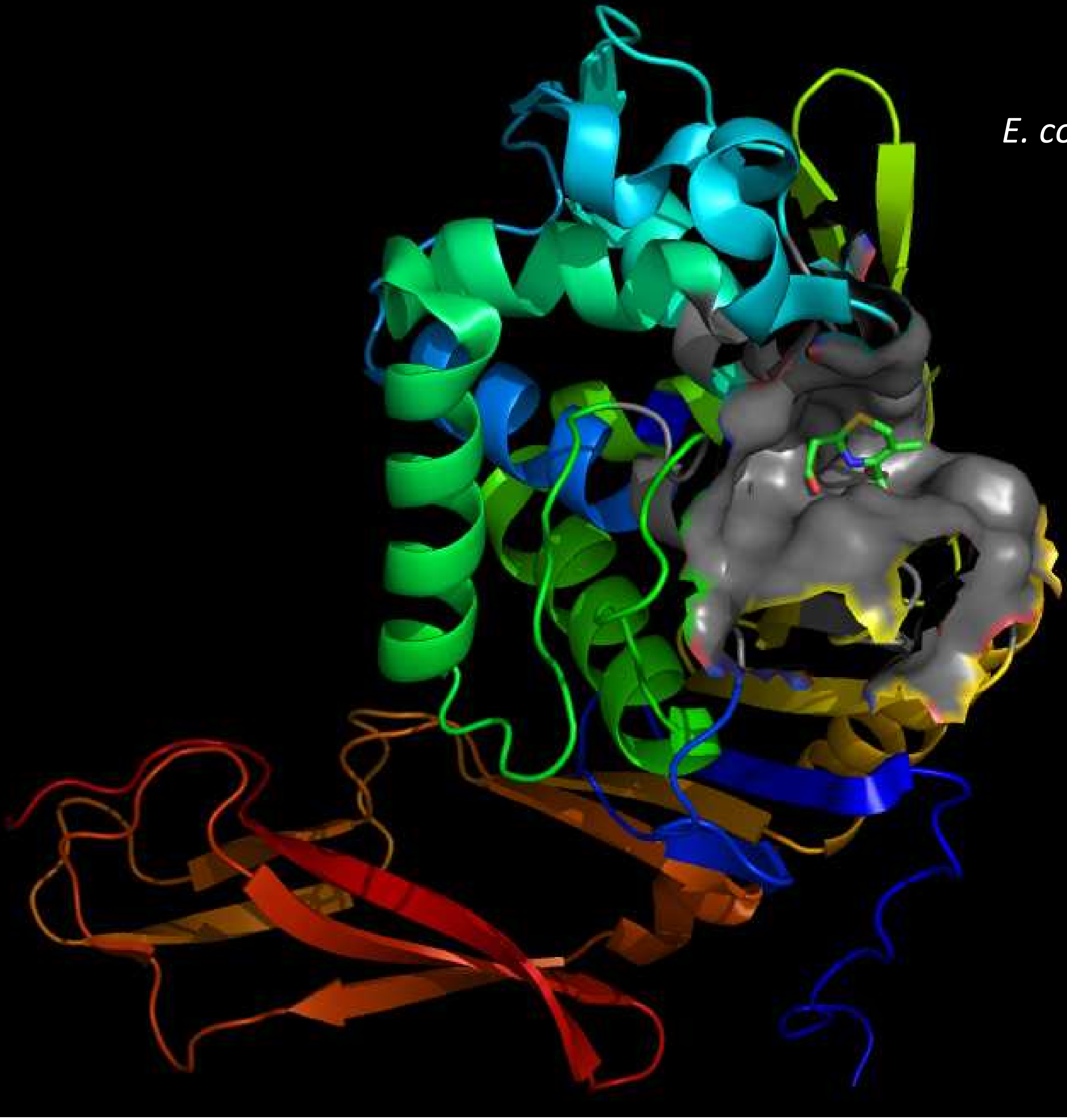
Hybrid Vigor

Using Heterogeneous HPC to Accelerate Chemical Biology

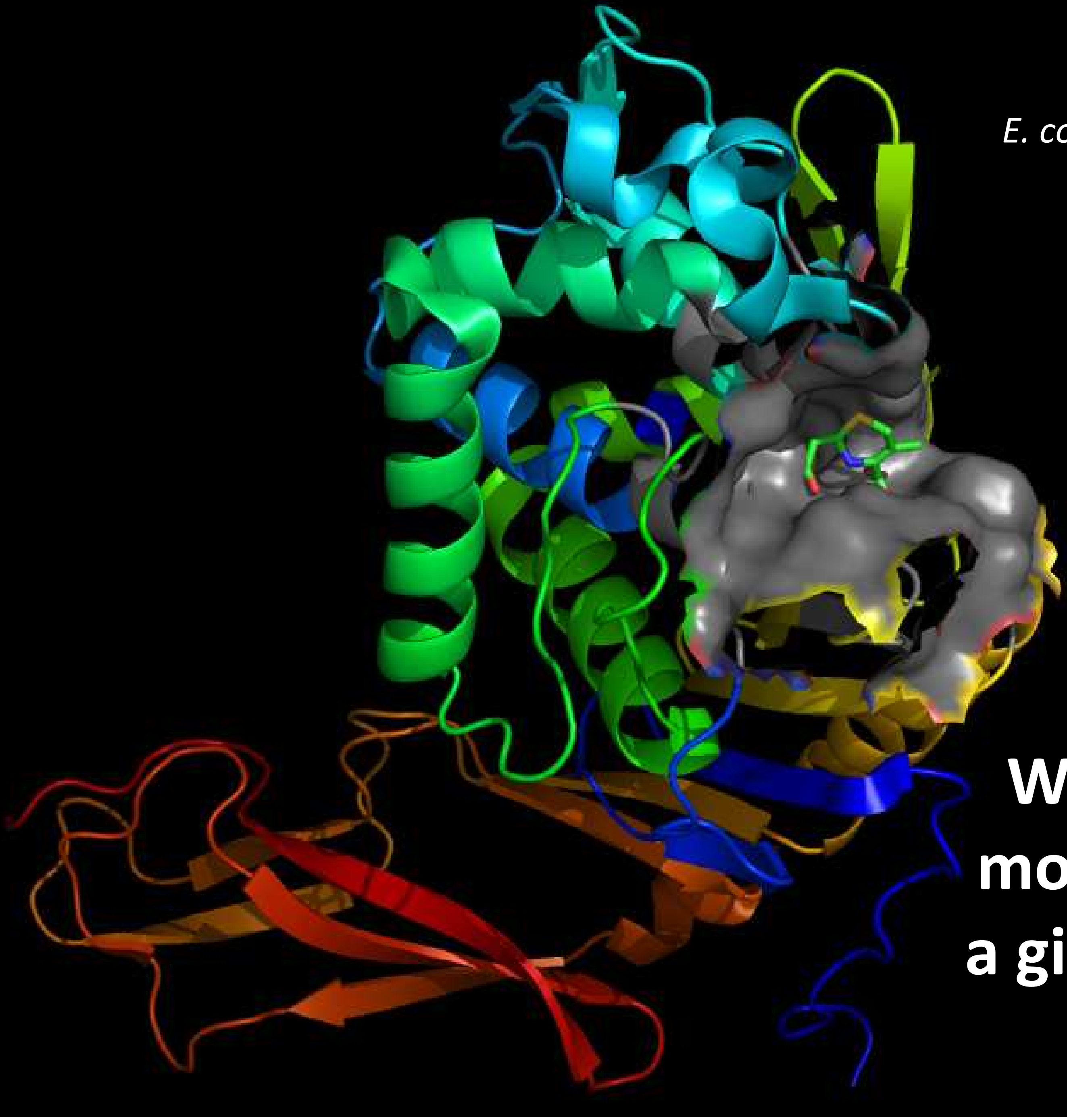
Imran Haque
Department of Computer Science
Stanford University

http://cs.stanford.edu/people/ihaque
http://folding.stanford.edu

Bio-Molecular Simulations on Future
Computing Architectures @ ORNL, 17 Sep 2010

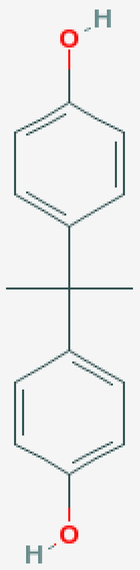


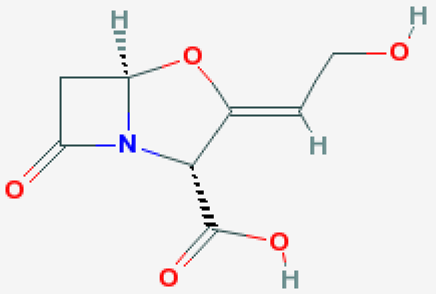
E. coli protein ???

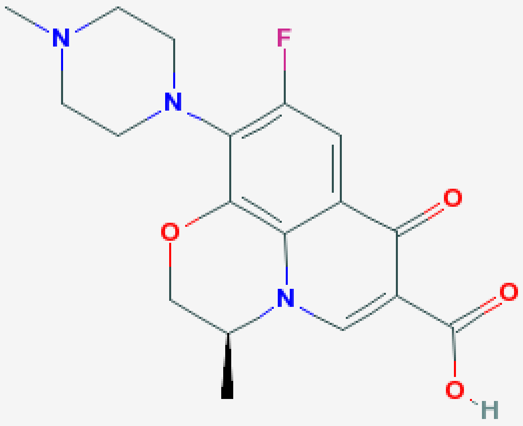


E. coli penicillin binding protein 5

Which small molecules will a given protein bind?

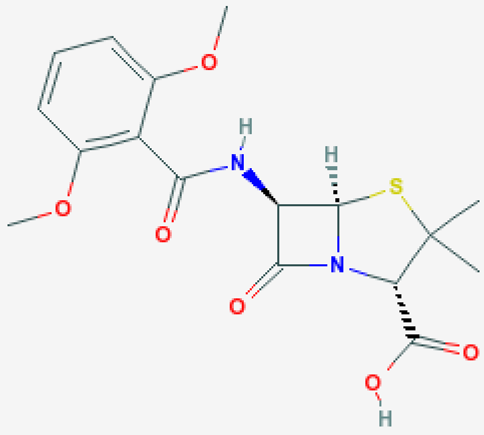


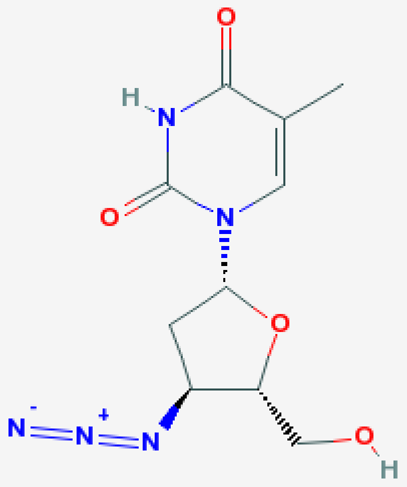


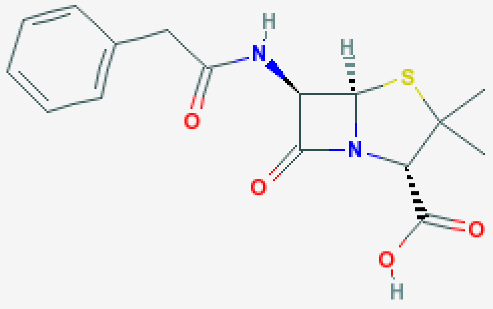


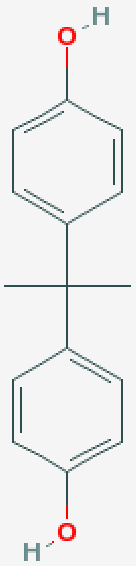
What do these compounds do?

- inhibit penicillin binding proteins?
- kill bacteria?
- kill viruses?

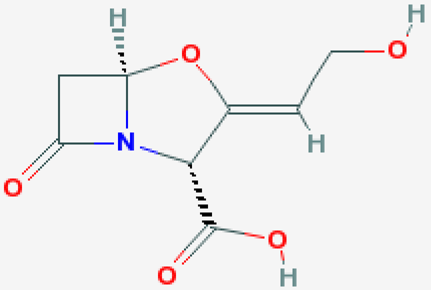




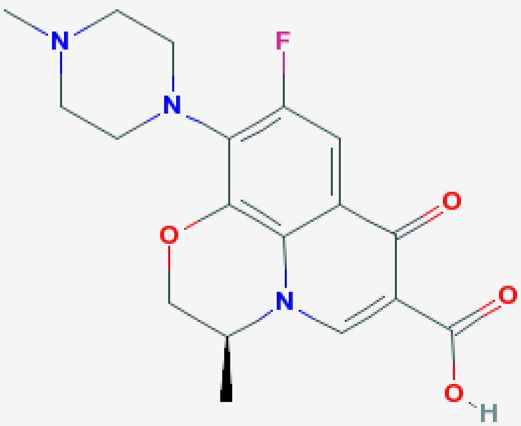




✖
✖
✖



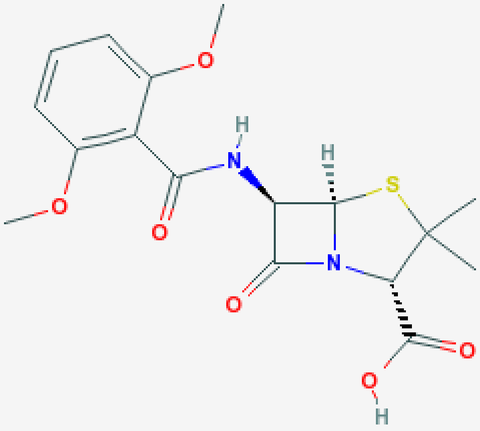
✖
✖
✖



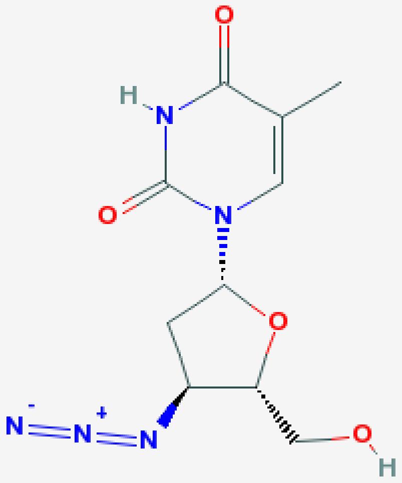
✖
✔
✖

What do these compounds do?

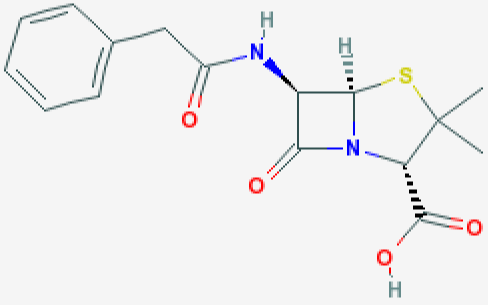
- inhibit penicillin binding proteins?
- kill bacteria?
- kill viruses?



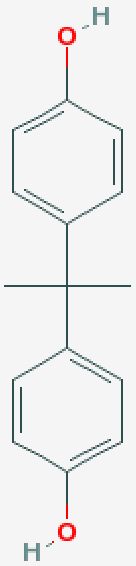
✔
✔
✖



✖
✖
✔

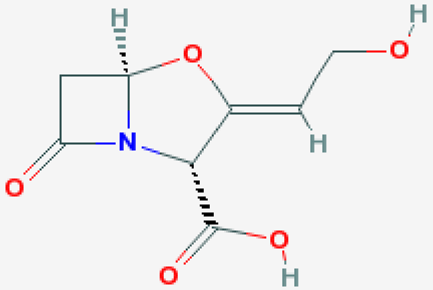


✔
✔
✖



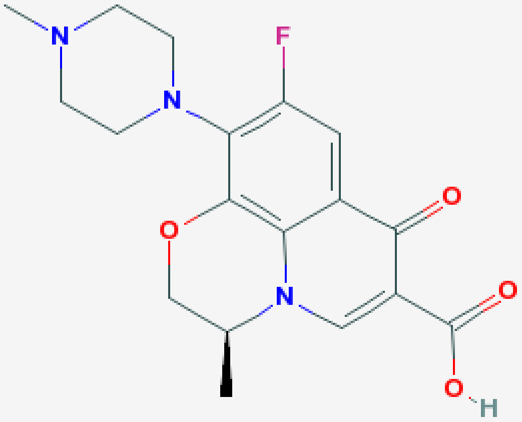
bisphenol A

estrogen mimic



clavulanic acid

beta-lactamase inhibitor

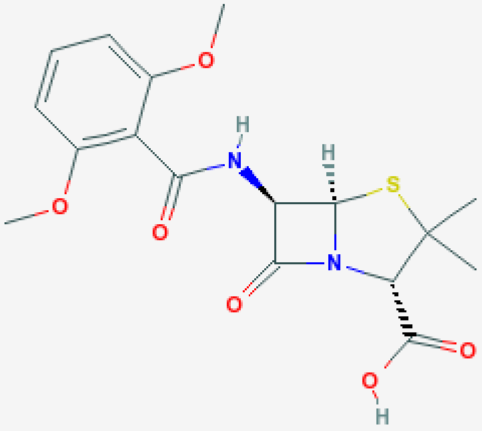


levofloxacin

DNA gyrase inhibitor

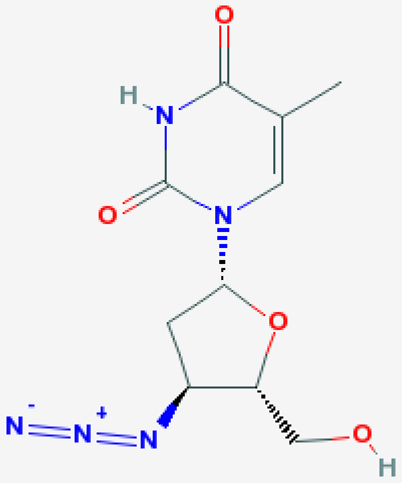
methicillin

beta-lactam antibiotic



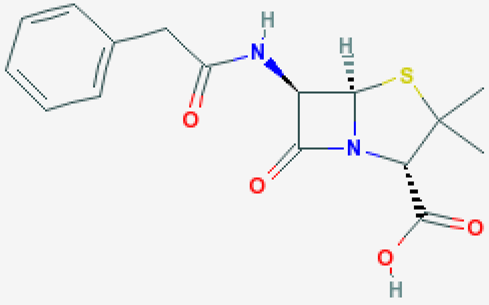
zidovudine

HIV RT inhibitor



penicillin G

beta-lactam antibiotic



Chemical Biology - Methods

- Experimental assays: expensive, labor-intensive
- Physical simulation?

http://simtk.org/home/openmm (Beauchamp, OpenMM team, Pande)

OpenMM – High Performance

Molecule	# atoms	ns/day	speedup*	GFLOPS (GPU)	GFLOPS (x86)
fip35	544	576	**128x**	311	657
villin	582	529	**136x**	328	692
lambda	1254	202	**255x**	547	1153
α-spectrin	5078	17	**735x**	805	1702

(*comparing a GTX280 to a single core of a 3GHz Core 2 Duo using the AMBER code; Fermi is ~2x faster!)

http://simtk.org/home/openmm (Beauchamp, OpenMM team, Pande)

OpenMM – Rapid Development

- Interface to Python
  - 8 lines to a customizable, high performance MD code
  - tweak to your heart's content, but keep high performance

```
import FF, Simulation
FField = FF.ForceField.LoadFromHDF("./Amber99.h5")
Conf   = FF.Conformation.LoadFromPDB("Test","./state0.pdb")
Topo   = FF.Topology.CreateTopologyFromConformation(Amber99,Conf)
Sim    = Simulation.Simulation.CreateSimulation(FField,Topo,Conf,
           Temp=300.,Friction=1.0,TimeStep=0.002,GBSA=True,BondConstr=True)
Sim.Step(50000)
Conf["XYZ"] = Sim.GetXYZ()
Conf.SaveToPDB("Traj2.pdb")
```

- Custom Force classes
  - code in equations, rather than CUDA/OpenCL, with high performance

```
map<string, CustomFunction*> functions;
functions["fn"]      = new MyCustomFunction();
ParsedExpression exp = Parser::parse("cos(x)*fn(x/2)",functions);
```

Limitations of traditional parallel MD

- Parallelism by spatial decomposition
  - each CPU gets assigned atoms
  - calculates the force for “its” atoms
  - communication between boxes
- Challenge
  - how to break up the problem for billions of processors when you only have millions of atoms?
  - What do you do when you only have thousands?!?!?
- What about scaling to billions of processors?
  - **can’t have # processors > # atoms**
  - **machine may not even run long enough to checkpoint/restart**

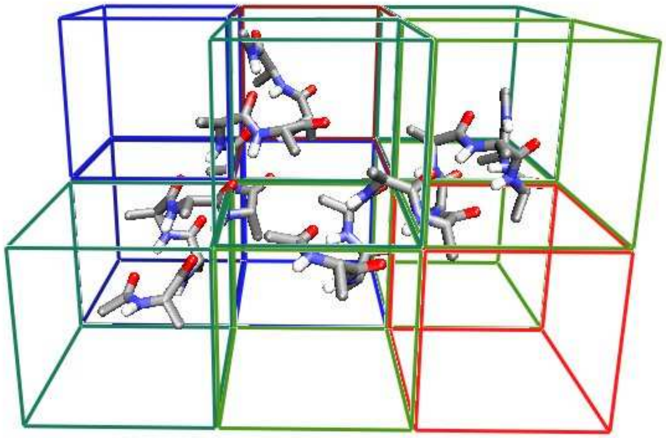
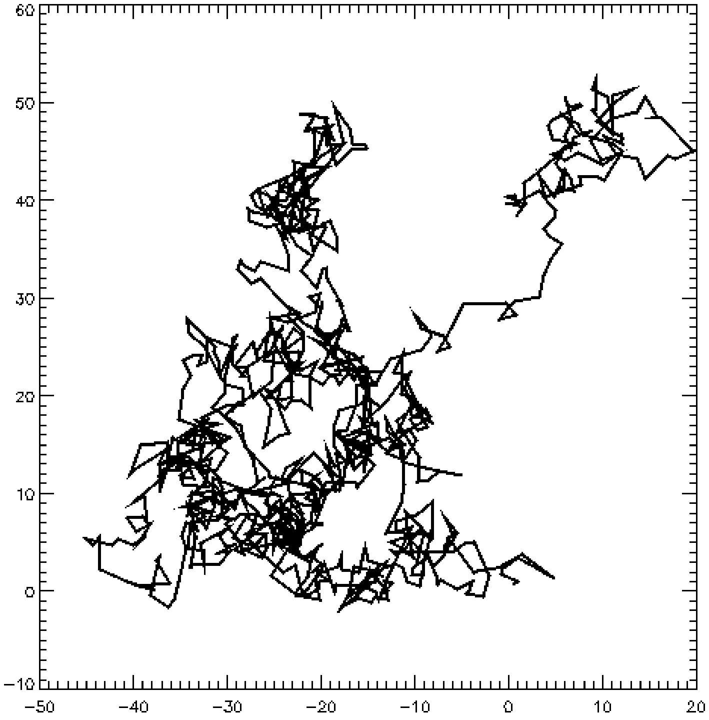
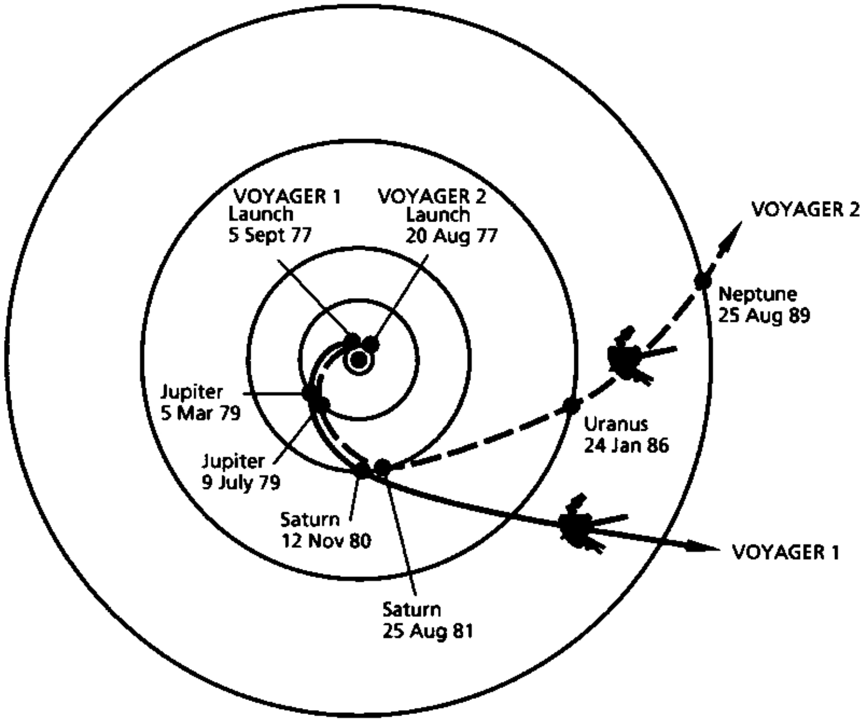


figure from http://www.ks.uiuc.edu/Research/Algorithms/

Anton from D. E. Shaw

How to think of MD simulations





YES!

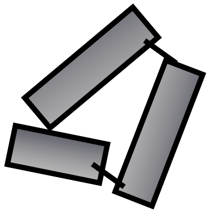
No

http://simtk.org/home/msmbuilder

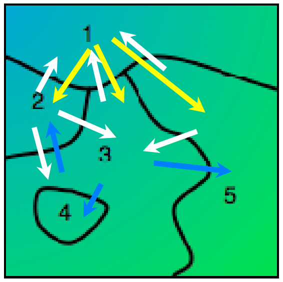
A statistical approach to simulation

1. Sample metastable states:

automatic algorithms to <u>adaptively sample</u> and <u>identify metastable states</u> via a <u>***kinetic*** clustering mechanism</u> (avoid one/low dimensional R.C.'s)

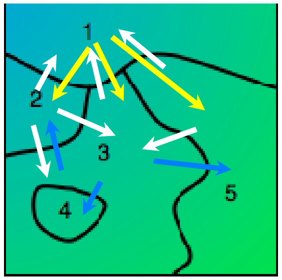


3 helix bundle



2. Build transition matrix:

use MD to sample transition probabilities (ideally adaptively -- which allows MSMs to be more efficient than very long runs)

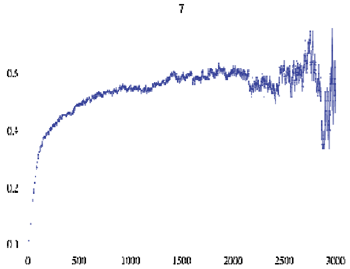


$$\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1N} \\ k_{21} & \ddots & & \\ \vdots & & & \\ k_{N1} & & & k_{NN} \end{bmatrix}$$

3. Use transition matrix:

transition matrix contains everything to predict structure, thermodynamics, and kinetics (built-in analysis via lumped MSM's)

$$\begin{bmatrix} k_{11} & k_{12} & \dots & k_{1N} \\ k_{21} & \ddots & & \\ \vdots & & & \\ k_{N1} & & & k_{NN} \end{bmatrix}$$

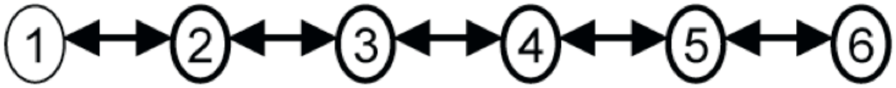


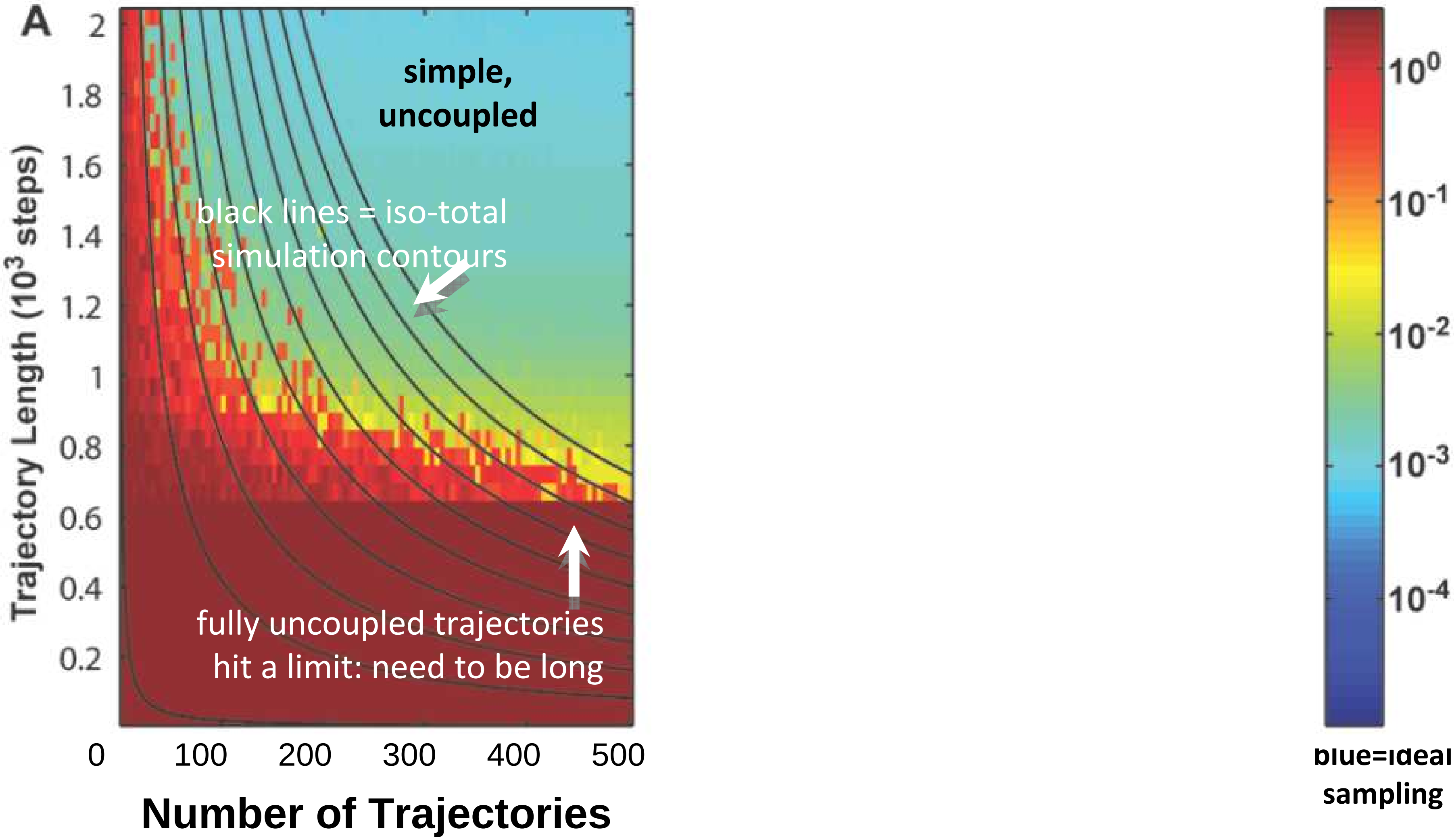
also see the work of: Caflisch, Chodera, Deuflhard, Dill, Hummer, Noé, Pande, Pitera, Singhal-Hinrichs, Roux, Schütte, Swope, Weber

(Bowman, Pande)

Shorter trajectories can be *more* efficient

Tests of a
linear network





simple,
uncoupled

black lines = iso-total
simulation contours

fully uncoupled trajectories
hit a limit: need to be long

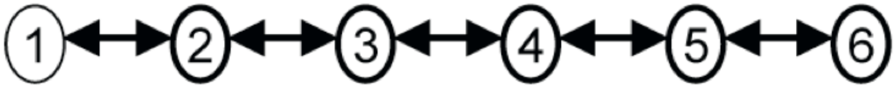
red=poor
sampling

blue=ideal
sampling

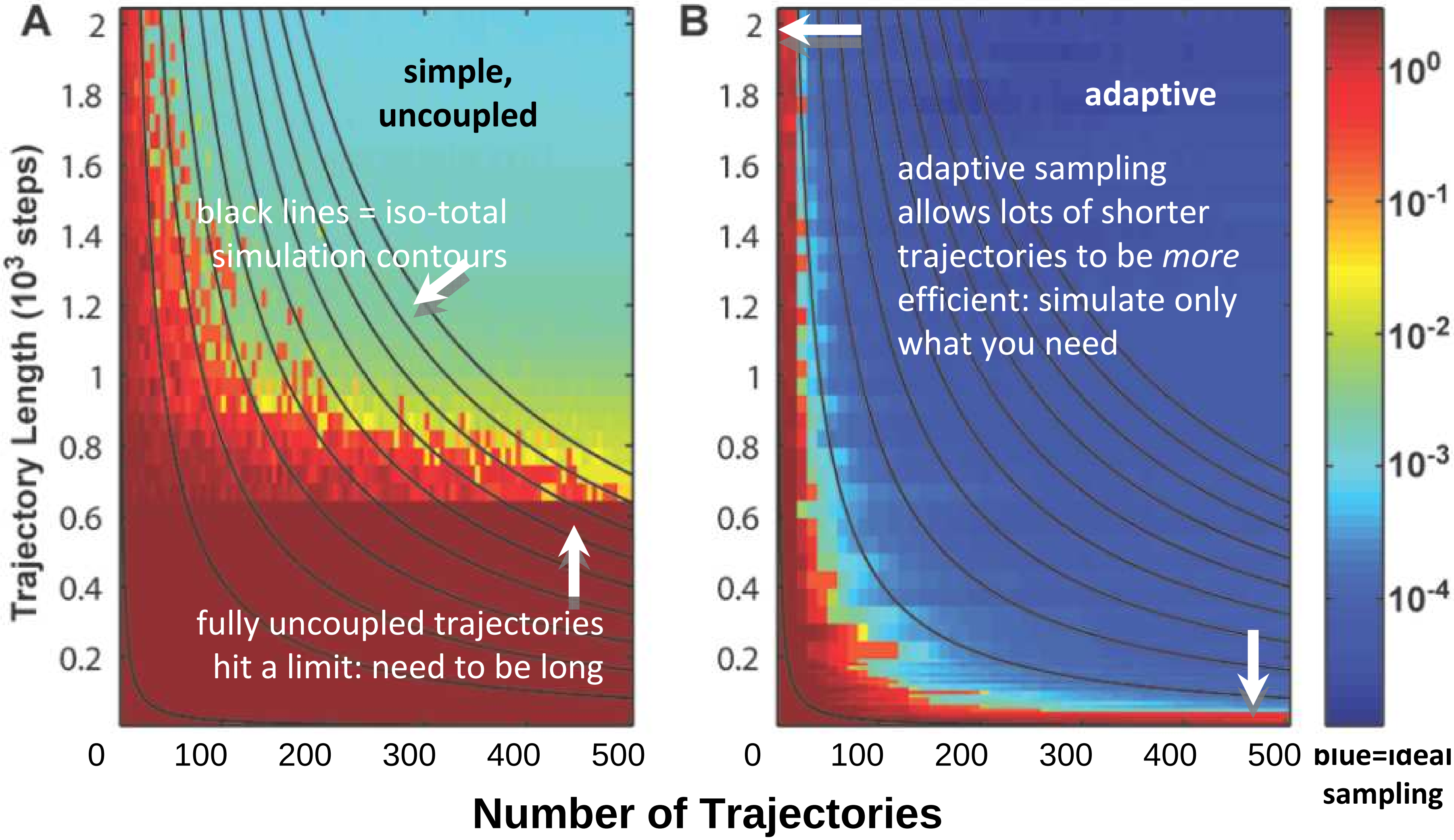
(Bowman, Pande)

Shorter trajectories can be *more* efficient

Tests of a linear network



red=poor sampling



simple, uncoupled

black lines = iso-total simulation contours

fully uncoupled trajectories hit a limit: need to be long

adaptive

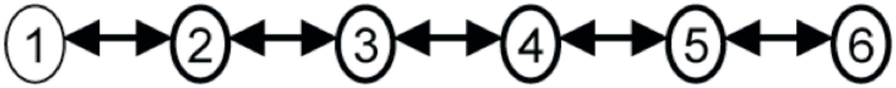
adaptive sampling allows lots of shorter trajectories to be *more* efficient: simulate only what you need

blue=ideal sampling

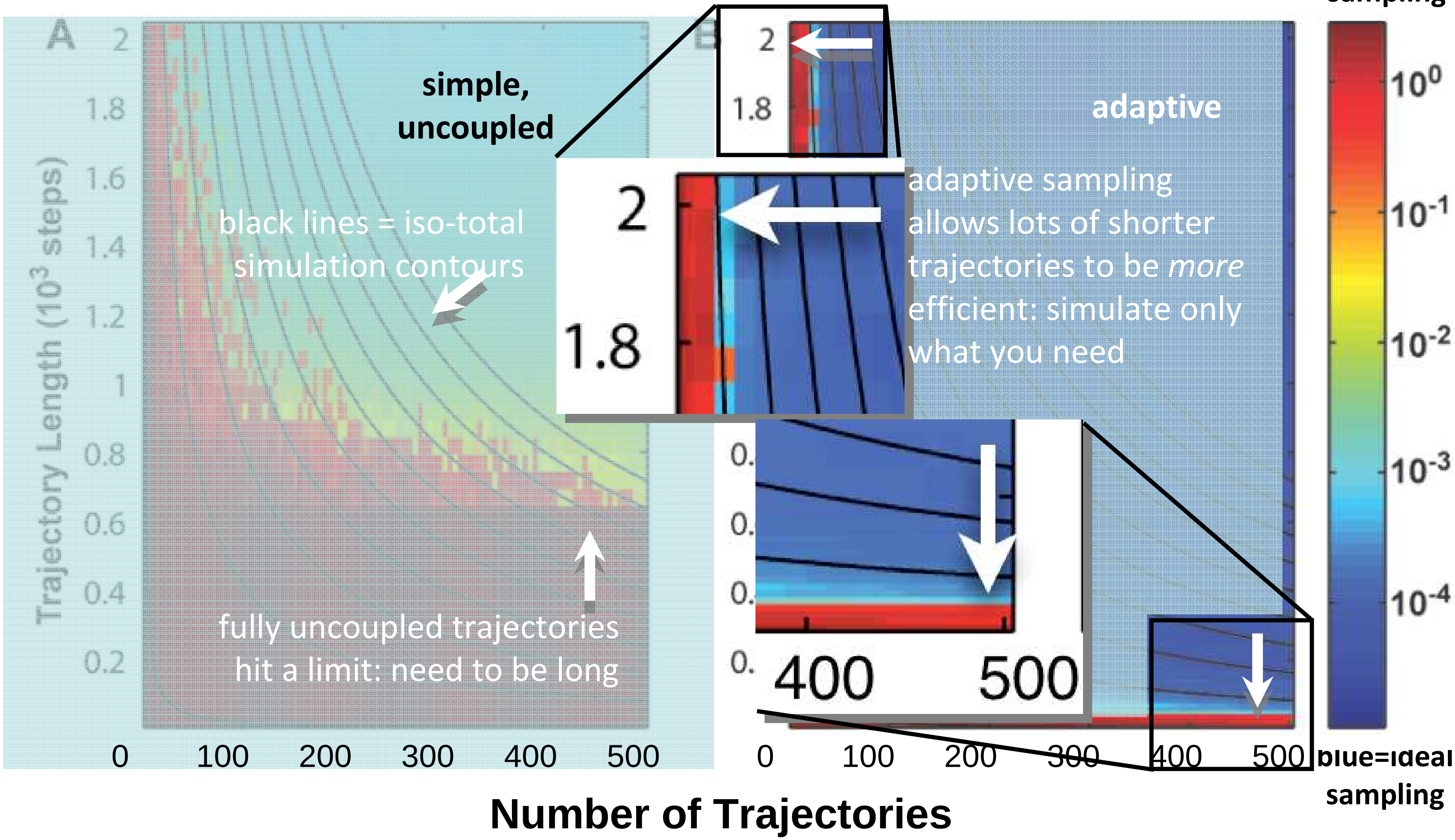
(Bowman, Pande)

Shorter trajectories can be *more* efficient

Tests of a linear network



red=poor sampling



simple, uncoupled

black lines = iso-total simulation contours

fully uncoupled trajectories hit a limit: need to be long

adaptive

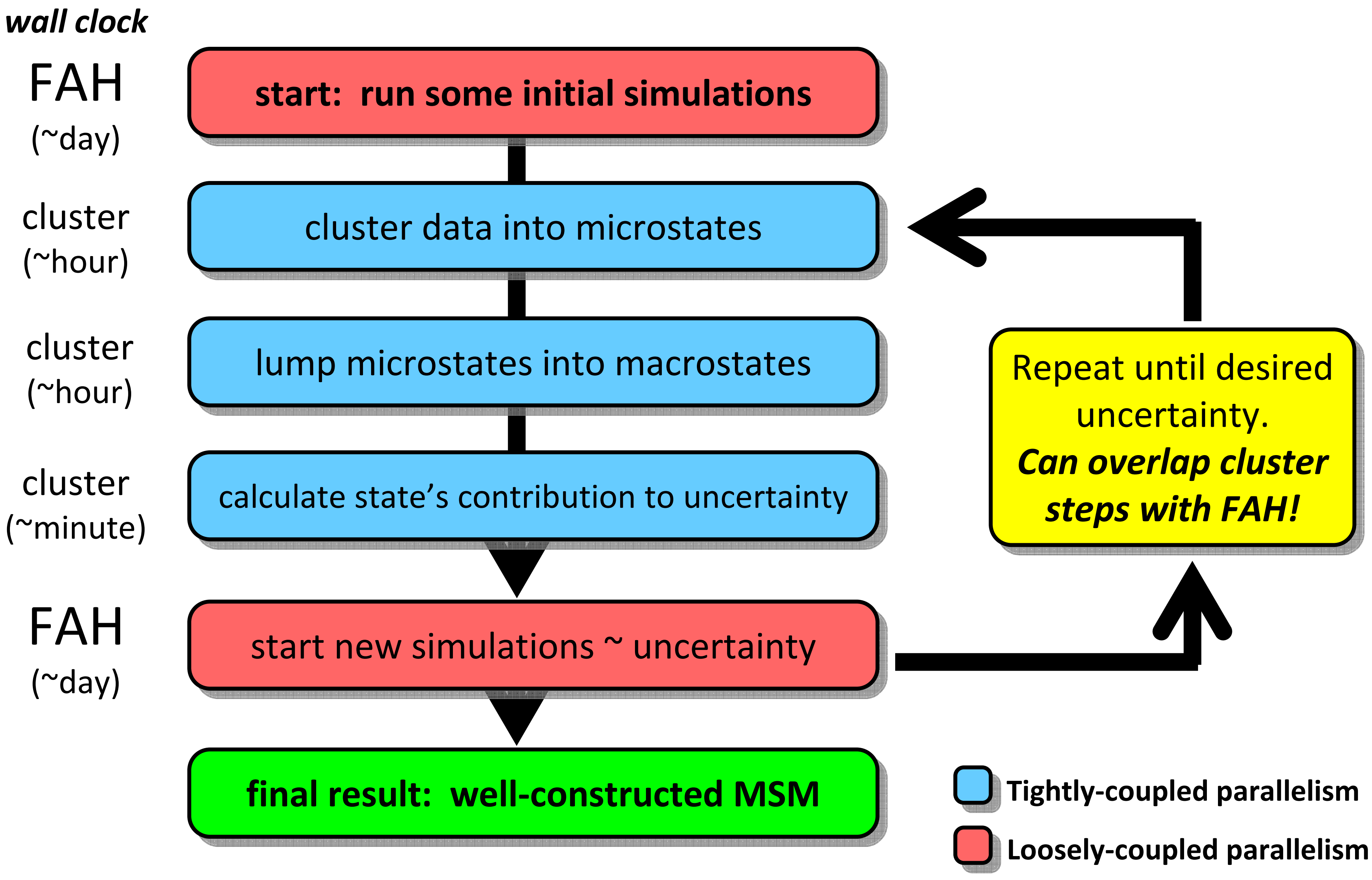
adaptive sampling allows lots of shorter trajectories to be *more* efficient: simulate only what you need

blue=ideal sampling

http://simtk.org/home/msmbuilder

(Singhal, Bowman, Haque, Pande)

Adaptive Sampling – Parallel + Resilient



wall clock

FAH (~day) — **start: run some initial simulations**

cluster (~hour) — cluster data into microstates

cluster (~hour) — lump microstates into macrostates

cluster (~minute) — calculate state’s contribution to uncertainty

FAH (~day) — start new simulations ~ uncertainty

final result: well-constructed MSM

Repeat until desired uncertainty. ***Can overlap cluster steps with FAH!***

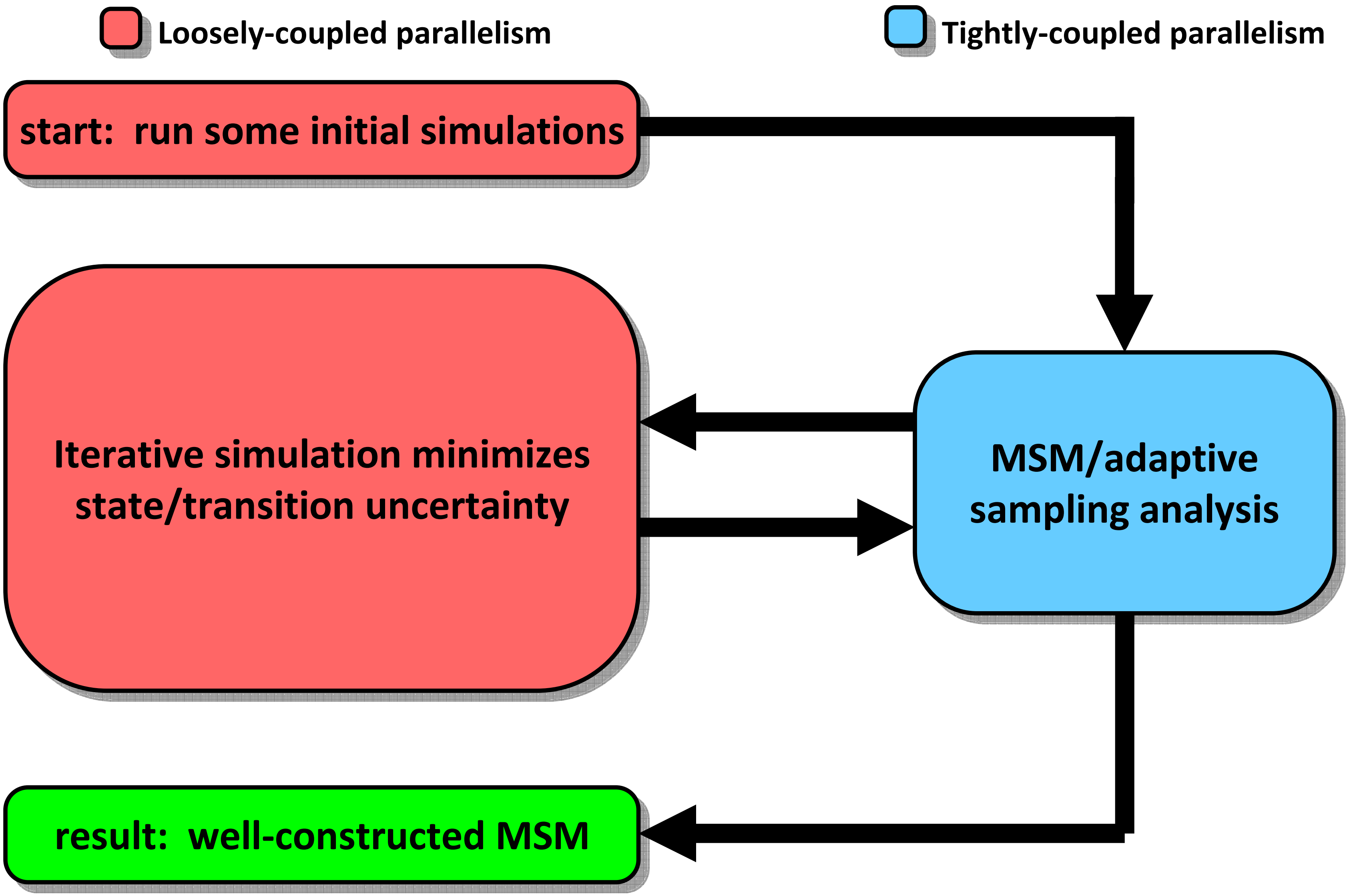
🟦 **Tightly-coupled parallelism**

🟥 **Loosely-coupled parallelism**

http://simtk.org/home/msmbuilder

(Singhal, Bowman, Haque, Pande)

Adaptive Sampling – Parallel + Resilient



🟥 Loosely-coupled parallelism

🟦 Tightly-coupled parallelism

start: run some initial simulations

Iterative simulation minimizes state/transition uncertainty

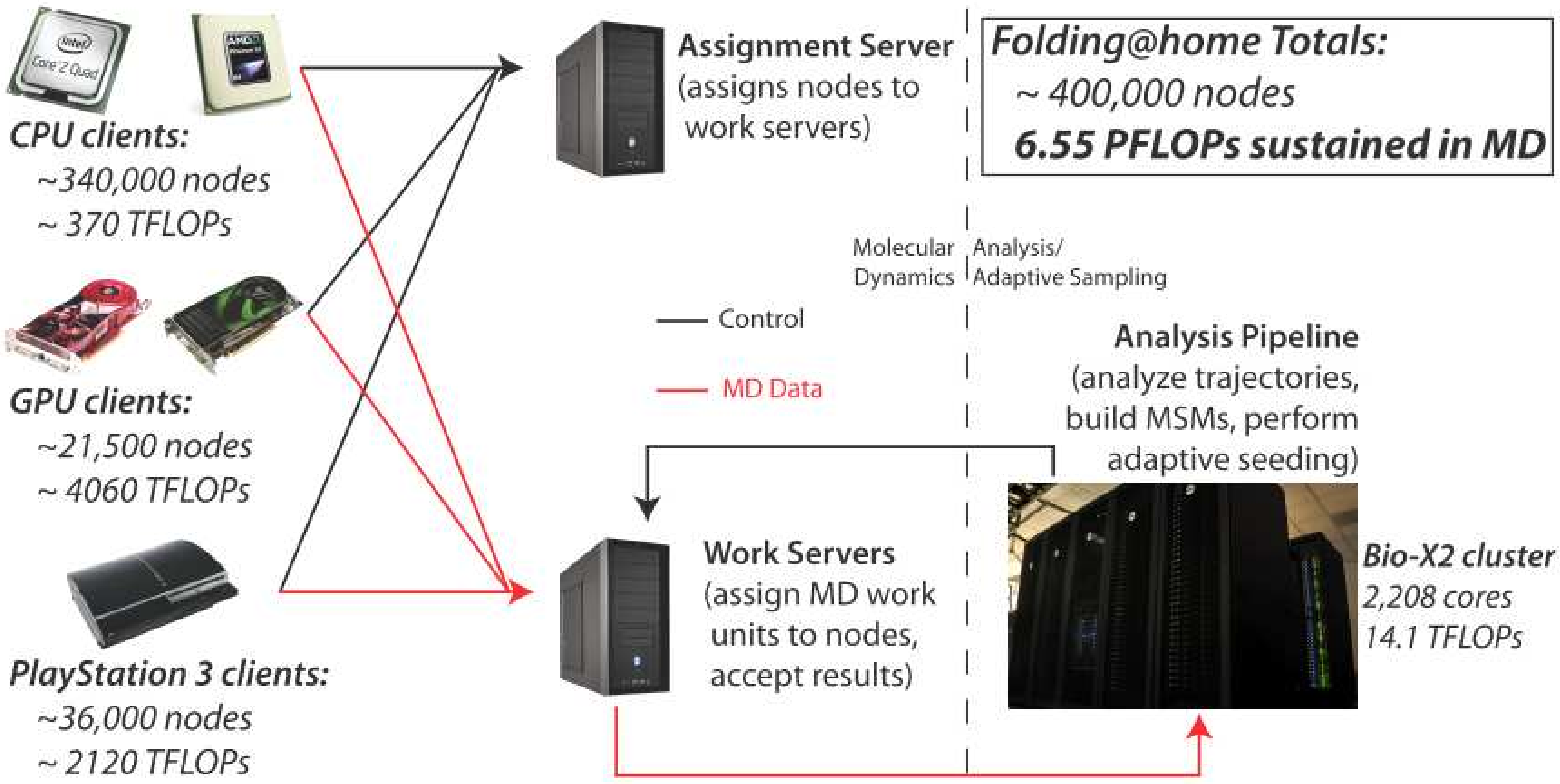
MSM/adaptive sampling analysis

result: well-constructed MSM

http://folding.stanford.edu (Pande Group)

Folding@home – Parallel + Resilient

🟥 **Loosely-coupled parallelism** 🟦 **Tightly-coupled parallelism**



CPU clients:
~340,000 nodes
~ 370 TFLOPs

GPU clients:
~21,500 nodes
~ 4060 TFLOPs

PlayStation 3 clients:
~36,000 nodes
~ 2120 TFLOPs

Assignment Server
(assigns nodes to work servers)

Folding@home Totals:
~ 400,000 nodes
6.55 PFLOPs sustained in MD

Molecular Dynamics | Analysis/ Adaptive Sampling

— Control
— MD Data

Analysis Pipeline
(analyze trajectories, build MSMs, perform adaptive seeding)

Work Servers
(assign MD work units to nodes, accept results)

Bio-X2 cluster
2,208 cores
14.1 TFLOPs

“Real” Chemistry: States and Rates

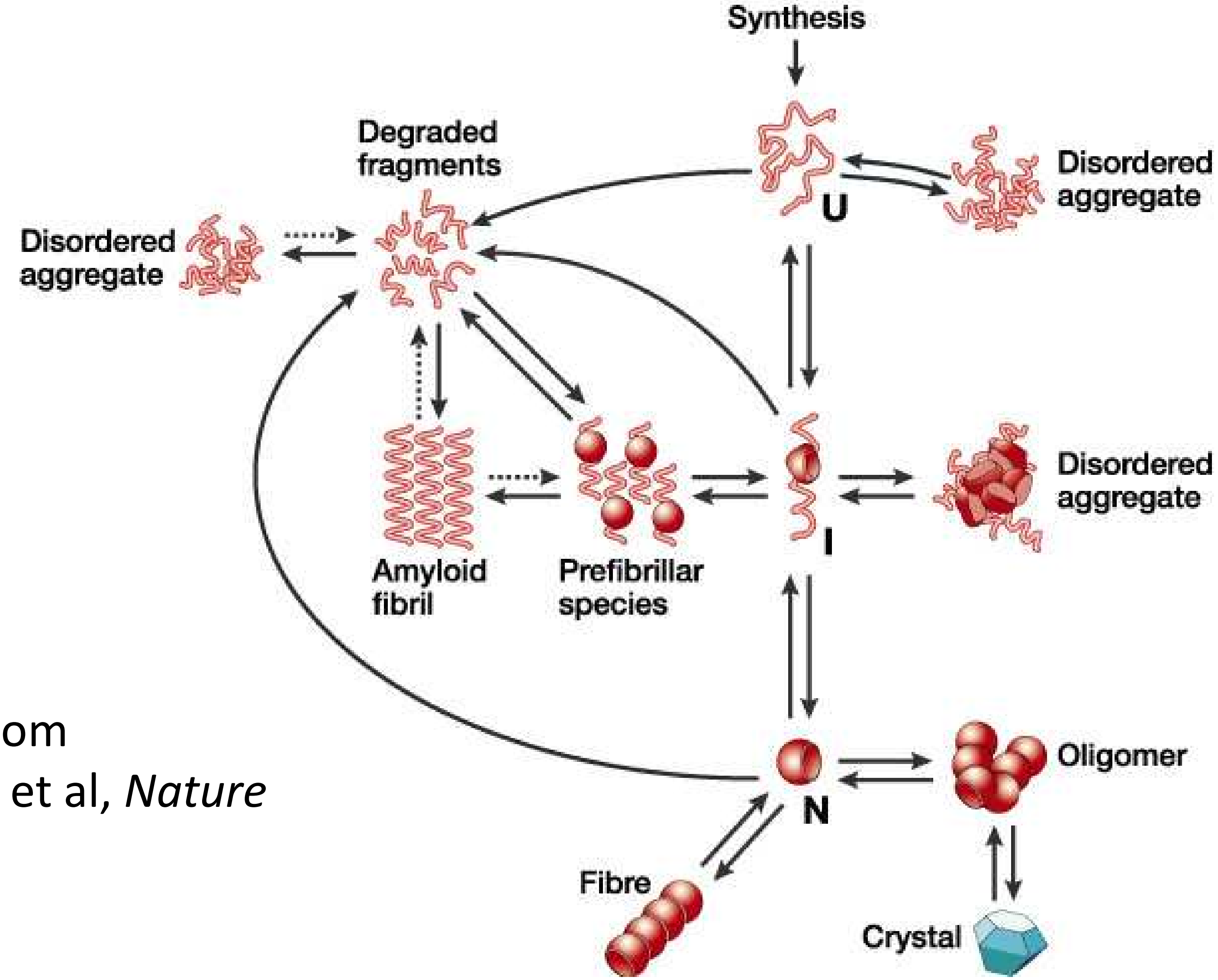
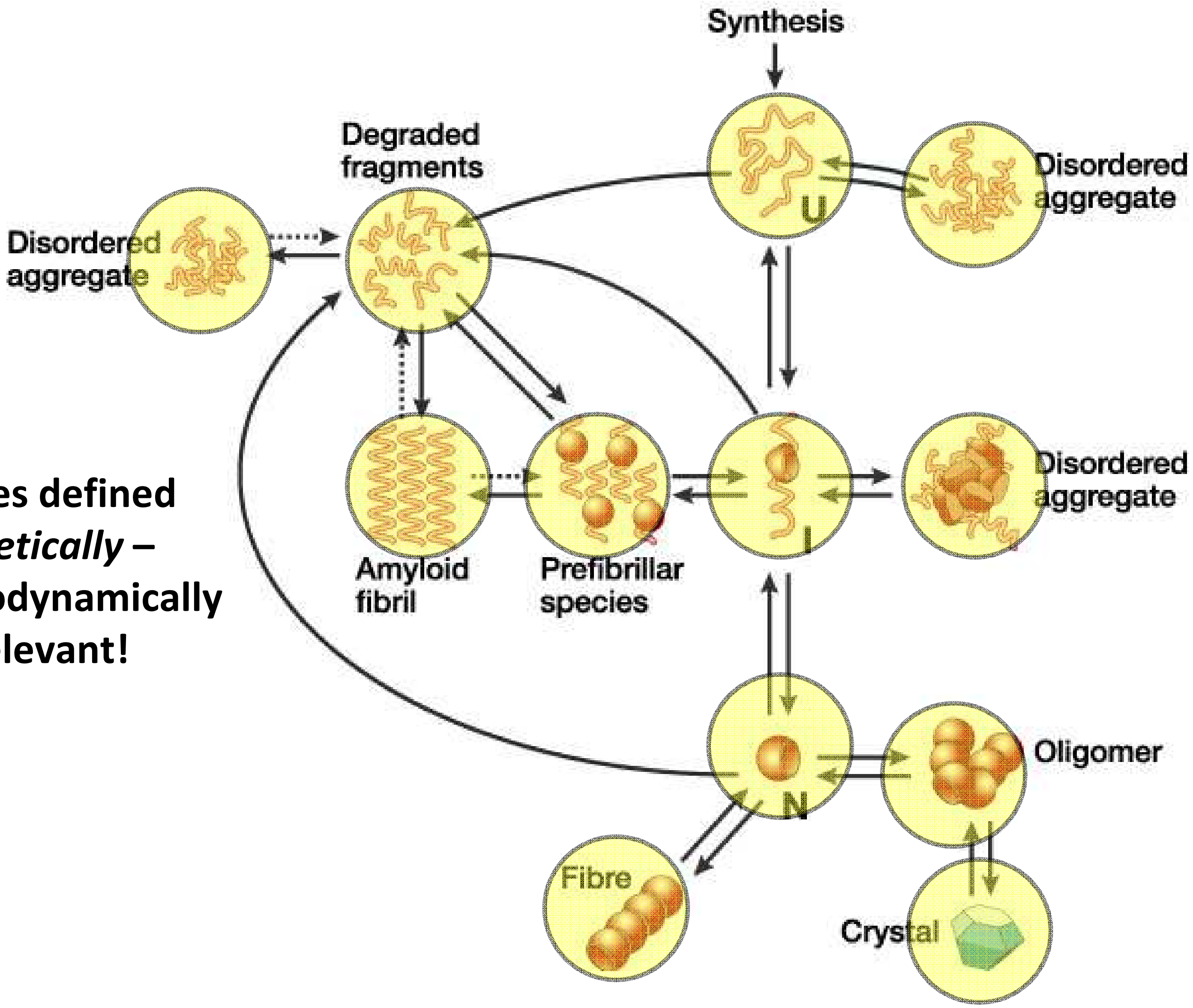


Figure from
Dobson, et al, *Nature*

http://simtk.org/home/msmbuilder

MSMs let us compute states and rates



States defined *kinetically* – thermodynamically relevant!

Chemical Biology - Methods

- Experimental assays: expensive, labor-intensive

- Physical simulation: expensive, slow, questionably accurate

- **Is there an alternative to giant molecular dynamics simulations for large-scale/high-throughput work?**

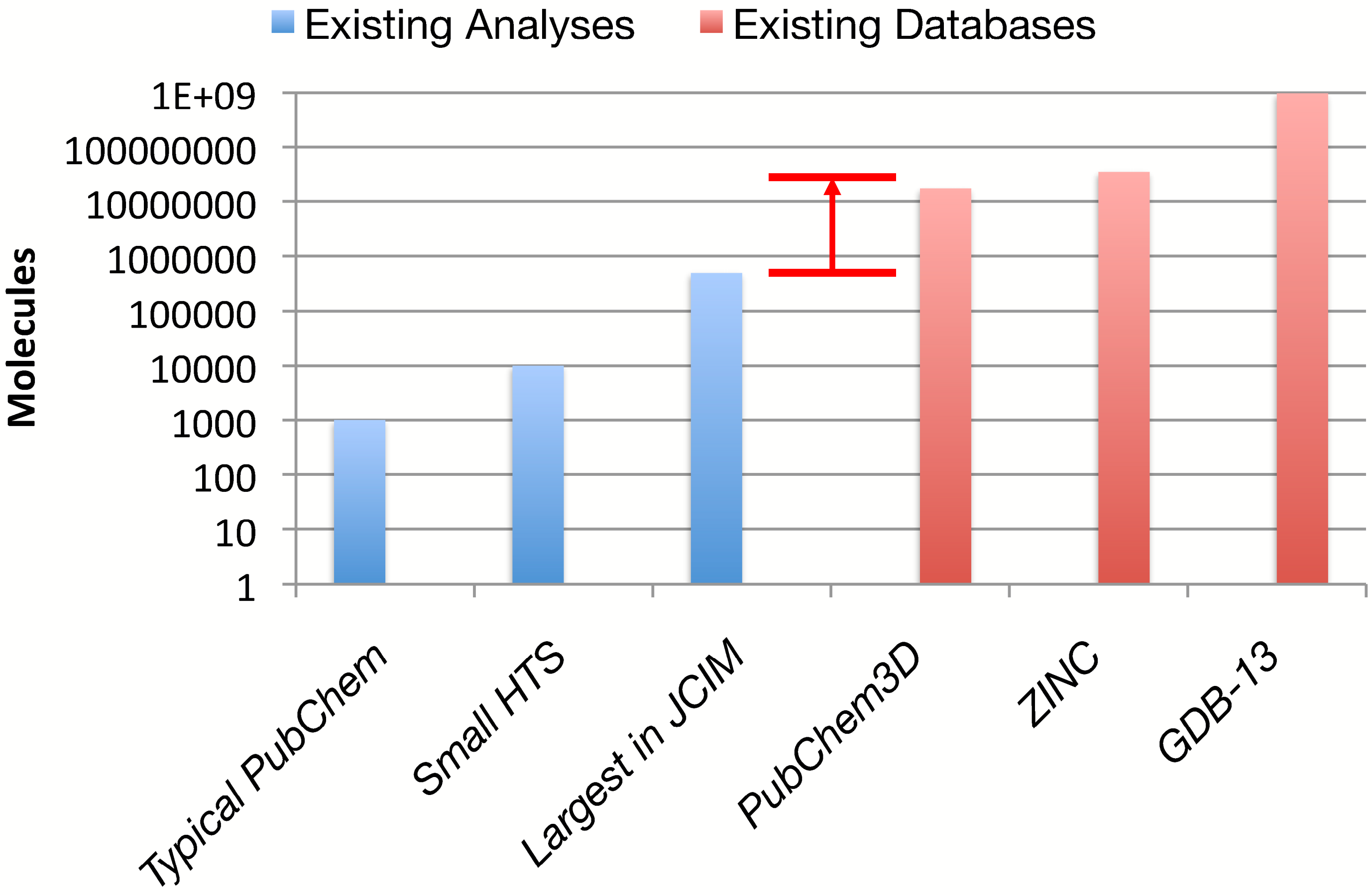
Chemical Databases

- A modern trend – giant **public** databases of chemical assay data
  - NCBI PubChem: 34,340 assays; 965,730 compounds
  - EBI ChEMBLdb: 8,054 targets; 600,625 compounds

- Companies releasing their internal databases
  - **GlaxoSmithKline**: Gamo et al. Thousands of chemical starting points for antimalarial lead identification. *Nature* **465,** 305-310 (20 May 2010).

- **Let's learn from this data and make predictions – chemical informatics or data mining!**

The Cheminformatics Gap



Computational analysis has not kept up with growth in chemical databases: the ***cheminformatics gap.***

Not just a linear gap

- Chemical similarity comparison is a common bottleneck in chemical algorithms
- How many similarities for N molecules?
  - **Virtual screening, k-means clustering: O(N)**
  - **Hierarchical clustering, network analysis: O(N^2)**
  - **LM hierarchical: O(N^3)**

The gap is not just 10x-100x…

more like 100x – 1 million x!

The storage challenge

- Making an O(N^2) method faster is not enough:

Problem size	CPU time	Storage needed
10 mols	1 ms	1 kB
10K mols	1 min	1 GB
100K mols	1 day	1 TB
10M mols	**3 yr**	**1 PB**
1B mols	**30K yr**	**10K PB**

- Computing on existing-scale datasets requires entire datacenters’ worth of storage.

A Modest Proposal

- Let’s calculate all the pairwise similarities for compounds in PubChem3D (N = 17M) based on 3D shape and 2D chemical similarity

- 3D: OpenEye ROCS: 150/sec/core = *30K cpu-yr*
  2D: OpenEye LINGO: 1M/sec/core = *4.5 cpu-yr*
  - **1 PB <u>per matrix</u>**

A Modest Proposal

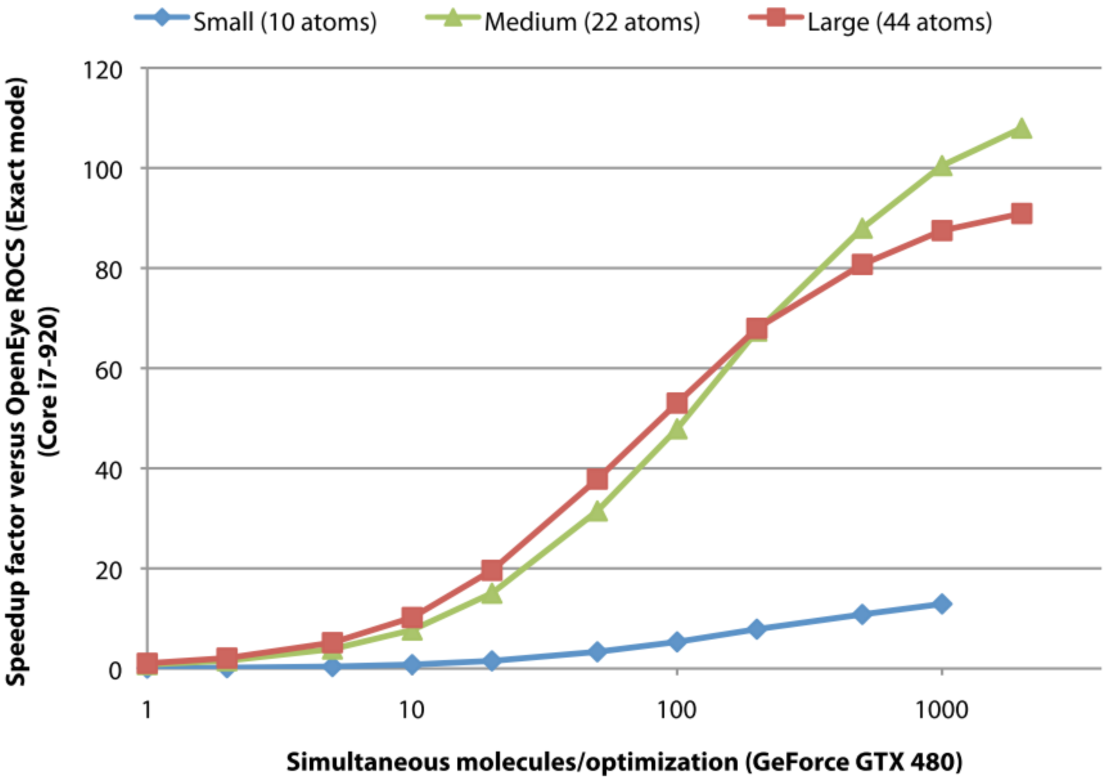
- Let's calculate all the pairwise similarities for compounds in PubChem3D (N = 17M) based on 3D shape and 2D chemical similarity

- 3D: OpenEye ROCS: 150/sec/core = *1.5 Jaguar-mth*
  2D: OpenEye LINGO: 1M/sec/core = *30 Jaguar-sec*
  - **13% of NCCS HPSS <u>per matrix</u>**

- Let's accelerate this with **heterogeneous** HPC!
  - High speed + high efficiency
  - Reliability? (See MemtestG80)

PAPER: GPU-Accelerated 3D Sim

- Use GPUs to accelerate 3D shape-only comparison: **100x speedup**

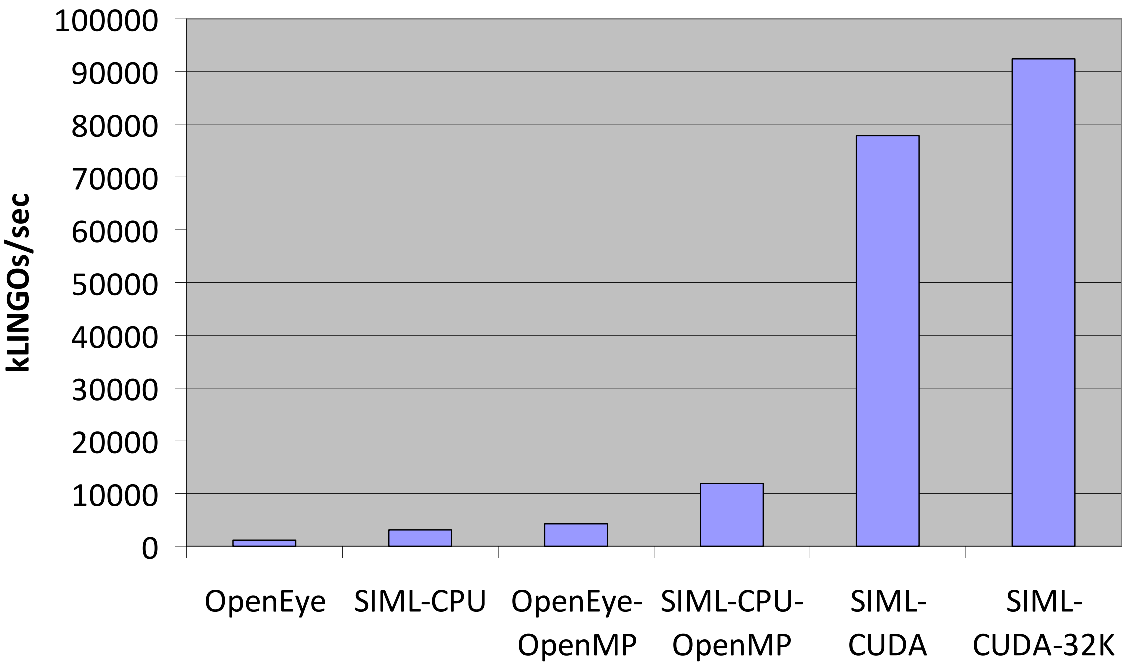


http://simtk.org/home/paper

Haque IS and Pande VS. *J. Comput. Chem.*, **2010,** *31*(1), pp 117-132

SIML: GPU-Accelerated 2D Sim

- 2D similarity has poor internal parallelism
- Invented new GPU-appropriate algorithm for LINGO
- Run one LINGO per compute unit (>200/GPU)



3x speedup with new algorithm on CPU

82x speedup on GPU

http://simtk.org/home/siml

Haque IS, Pande VS, Walters WP. *J. Chem. Inf. Model.*, **2010** *50*(4), pp 560-564

A Humble Proposal

- Let’s calculate all the pairwise similarities for compounds in PubChem3D (N = 17M) based on 3D shape and 2D chemical similarity

- 3D: PAPER: 15K/sec/gpu = ~ *300 gpu-years*
2D: SIML: 91M/sec/gpu = ~ *4 gpu-weeks*
  - **2D: 1 GPU is faster than reading solution from disk!**

- We’re not quite there yet for 3D…

SCISSORS: Math for Fun and Profit

- Many molecular similarity methods report similarity as a Tanimoto score
- How can we use the mathematical structure of Tanimotos to gain insight into the metrics and **calculate them faster**?

Classical vector Tanimoto returns value in [-1/3, 1] for a pair of vectors A, B in terms of their inner products

$$T_{AB} = \frac{\langle A, B \rangle}{\langle A, A \rangle + \langle B, B \rangle - \langle A, B \rangle}$$

Tanimoto equation can be rearranged to get inner product in terms of Tanimoto and vector magnitudes

$$\langle A, B \rangle = \frac{T_{AB}}{1 + T_{AB}} \left(\langle A, A \rangle + \langle B, B \rangle \right)$$

Haque IS and Pande VS. *J. Chem. Inf. Model.,* **2010** *50*(6), pp1075-1088.

SCISSORS: Derivation

- Assume molecules can be represented as vectors in $\mathbf{R}^N$
- Simple assumptions on <A,A> and <B,B> get us <A,B>

$$\langle A, B \rangle = \frac{2T_{AB}}{1 + T_{AB}}$$

- Given a matrix G of inner products, want matrix M with molecule vectors along rows

$$MM^T = G$$

- G is real-symmetric, so use eigenvalue decomposition

$$G = MM^T = VDV^T$$
$$M = VD^{\frac{1}{2}}$$

Haque IS and Pande VS. *J. Chem. Inf. Model.,* **2010** *50*(6), pp1075-1088.

SCISSORS: The key

- Select a small number **k** of molecules (**k** << **N**) to act as a "basis set"
- Do all-pairs comparison on basis set and decompose to molecule matrix **M**
- For each new "library" molecule **x**, run slow method only against basis set. Place inner products in a vector and solve for vector rep of **x** by least-squares:

$$M\vec{x} = T$$

- All-pairs: now only O(**kN**) slow computations!

Haque IS and Pande VS. *J. Chem. Inf. Model.,* **2010** *50*(6), pp1075-1088.

Hardly Even a Request…

- 3D: Using PAPER+SCISSORS (basis size=2700)

  17M * 2700 / 15000 = 35 gpu-day +

  17M * 17M / 600M = 5 gpu-day

  274,000x speedup (vs 30 000 cpu-yr)

- 2D: Using SIML

  17M * 17M / 91M = 36 gpu-day

  40x speedup (vs 4.5 cpu-yr)

- Storage: 200M for SIML, 17GB for SCISSORS

  33,000 x reduction (3D)

  2.8M x reduction (2D)

Doing it Faster *and* Better

- Intensive reparameterization of chemical similarity “forcefields”: 14-20D derivative-free optimization
- High-speed similarity allows exhaustive calculation of all similarities -> explicit significance estimates
- Future work: integration of biological data into similarity networks to make predictions

Acknowledgments

Stanford

- Vijay Pande (PI)
- Paul Novick
- Greg Bowman
- Kyle Beauchamp
- Randy Radmer
- Mark Friedrichs
- Peter Eastman

Collaborators

- Del Lucent
- Pat Walters
- Kim Branson
- Erik Lindahl
- Anthony Nicholls
- Brian Cole
- Roger Sayle

LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891

CSIRO

VERTEX™

OpenEye
Scientific Software

Conclusions

- Statistical approach **extends scalability and resilience** of MD to the exascale and unifies simulation and analysis
- New hardware and software technologies allow us to **bridge the cheminformatics gap** and scale analysis to multi-million molecule datasets
- Large-scale methods enable **statistically-rigorous** analysis and new insights into chemical space

ihaque@cs.stanford.edu