Elchanan Mossel
U.C. Berkeley

Recent Progress in Combinatorial Statistics

At Penn Statistics, Sep 11

Combinatorial Statistics

"Combinatorial Statistics":

Rigorous Analysis of Inference Problems where:

Estimating a discrete parameter (e.g. graph, order)

Explicit Sampling Complexity Bounds

Explicit Computational Complexity Bounds

Interest in both: Positive Results & Negative Results

Interdisciplinary (Stat, ML, Applied Prob., TCS, PAC ..)

Why Negative Results?

"I am not interested in negative science" (academy member when understanding sampling lower bounds)

Bible Codes

"The Duke University Scandal"

"Retractions in the medical literature: How many patients are put at risk by flawed research", Steen 10.

Ioannidis' : *Why Most Published Research Findings Are False*

Lecture Plan

Negative results:

On the usefulness of MCMC diagnostics.

Is it possible to recover graphical models?

Bayesian MCMC in Phylogeny mixtures.

Positive Results:

Recovering Phylogenetic Mixtures.

Recovering the Mallows model

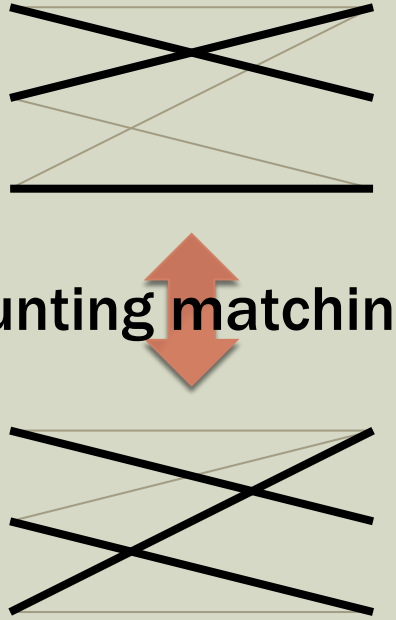
ON THE COMPLEXITY OF MARKOV CHAIN CONVERGENCE

With Nayantara Bhatnagar
UC Berkeley

Andrej Bogdanov
Chinese University of Hong Kong

Markov Chain Monte Carlo

volume estimation

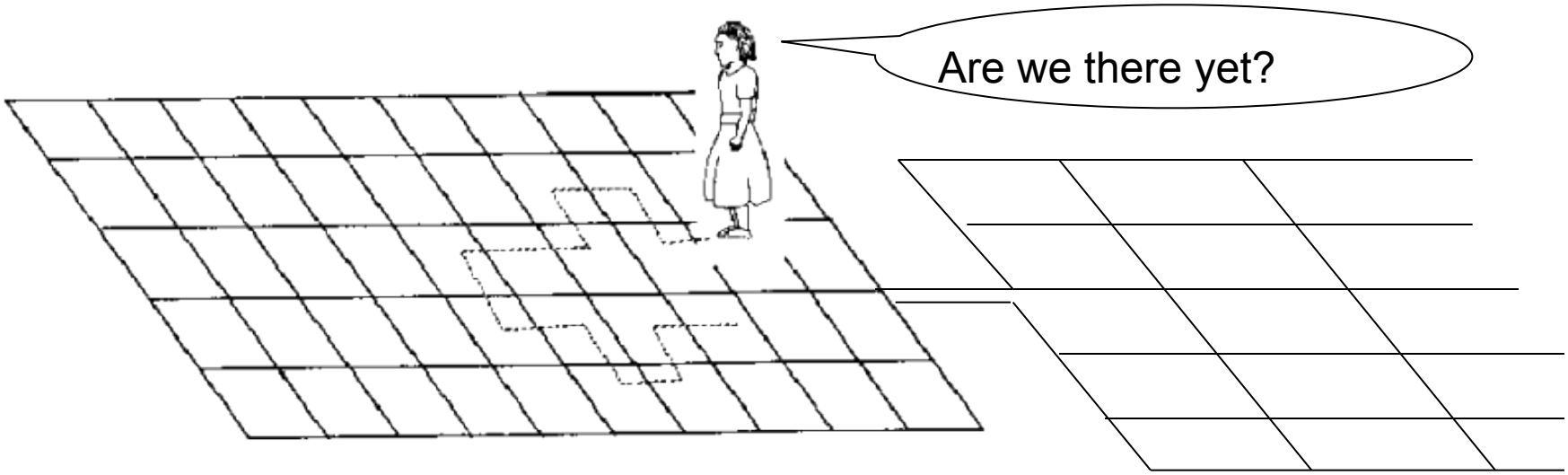


Markov Chains typically easy to design,

but guaranteeing convergence is difficult

Convergence Diagnostics

- A method to determine t such that $P^t(X_0, \cdot)$ is "sufficiently close" to π.
  - *MCMC in Practice, Gilks, Richardson & Spiegelhalter*



-

Convergence Diagnostics in the Literature

Visual inspection of functions of samples.
- *MCMC in Practice, Gilks, Richardson & Spiegelhalter*

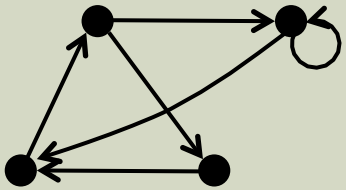
[Raftery-Lewis]
Bounding the variance of estimates of quantiles of functions of parameters.

[Gelman-Rubin]
Convergence achieved if the separate empirical distributions are approximately the same as the combined distribution.

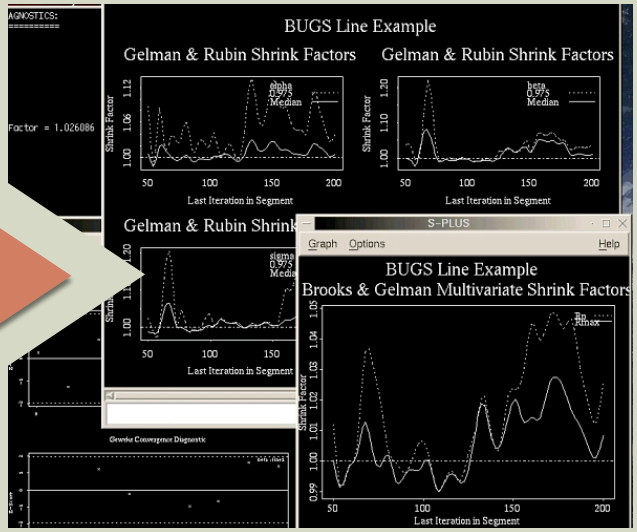
[Cowles-Carlin]

- Survey of 13 diagnostics designed to identify specific types of convergence failure.
- Recommend combinations of strategies – running parallel chains, computing autocorrelations.

Diagnostic tools



BOA



BOA is an R/S-PLUS program for carrying out convergence diagnostics and statistical and graphical analysis of Monte Carlo sampling output.

http://www.public-health.uiowa.edu/boa/

Our objective

- It is known that these diagnostics are not always accurate

- However, the inherent difficulty in designing such tools has not been studied

We point out some complexity theoretic limitations of Markov Chain Monte Carlo diagnostic tools

The model

input: a circuit C describing the transition function over state space $\{0,1\}^n$

$$C(\text{state}, \text{randomness}) = \text{new state}$$

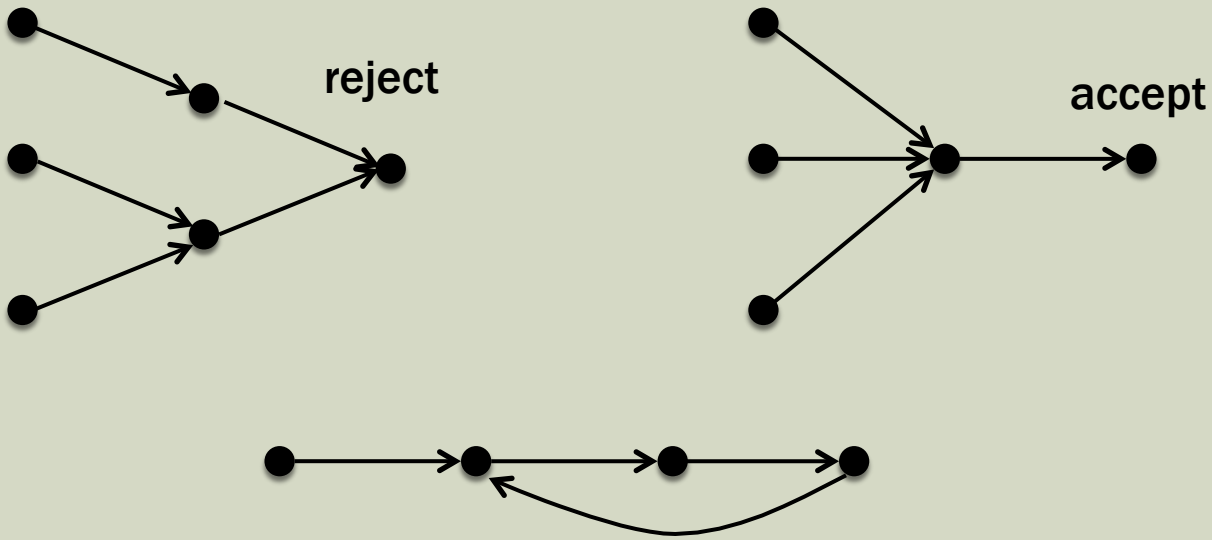
a convergence time bound t

a variation distance bound d

problem: does the MCMC C approximately reach variation distance d after t steps?

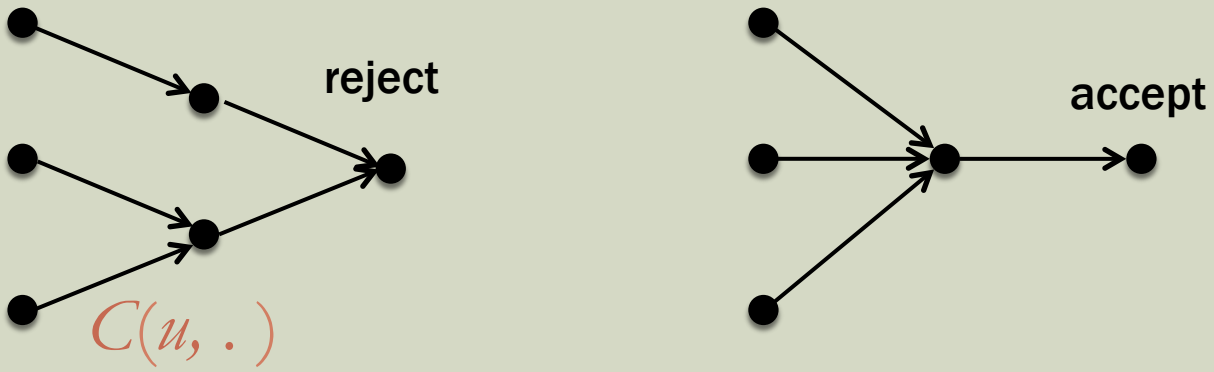
Note: Diagnostics can run C but also do many other things.

1. The PSPACE barrier



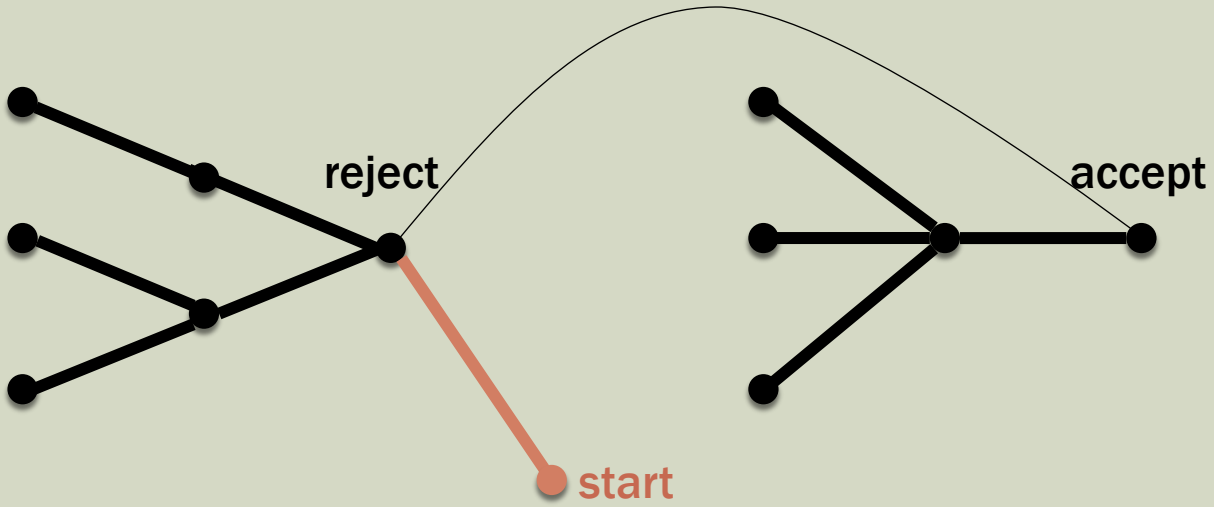
configuration graph of a Turing machine

1. The PSPACE barrier



configuration graph of a PSPACE Turing machine (that always halts)

1. The PSPACE barrier



In general it is PSPACE hard to get even
an exponential approximation of mixing time

tight as mixing time is computable in PSPACE

2. The SZK barrier

- What if we know the chain mixes somewhat fast, but we want a more precise estimate?

- This captures cases when provable bound is weak, but we suspect MCMC does better in practice

promise: mixing time is at most n^2

question: is it actually at most $2n$?

2. The SZK barrier

- **An SZK hard problem [Sahai and Vadhan]:**

input: two distributions D_0 and D_1 over $\{0,1\}^n$ described by sampling circuits

problem: are they close or far apart in variation distance?

Belief: Problem is computationally hard. Connections to Crypto.

2. The SZK barrier

- **The hard Markov Chain:**

  with probability $1 - \delta$**, retain previous state**

  with probability $\delta/2$**, resample from** D_0

  with probability $\delta/2$**, resample from** D_1

stationary distribution $= \frac{1}{2}(D_0 + D_1)$

distance at time $t = \frac{1}{2}(1-\delta)^t \operatorname{dist}(D_0, D_1)$

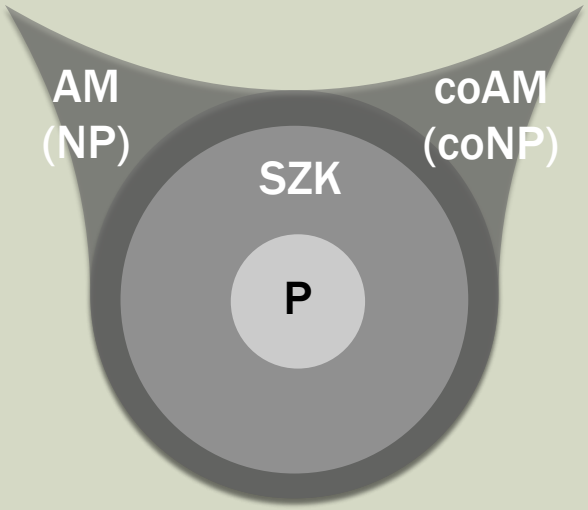
2. The SZK barrier

- For example, given a Markov Chain that is known to converge in time n^2, it is SZK-hard to tell if it is

  within $1/3$ of stationary at time n, or

  not within $2/3$ of stationary at time $2n$.

- Conversely, given a starting state s, detecting approximate convergence is an $\mathrm{AM} \cap \mathrm{coAM}$ problem



3. The coNP barrier

promise: mixing time is at most n^6
from any start state

- It is coNP-hard to tell if the Markov Chain is within distance to stationary

  at most $1/3$ at time n, from every start state, or

  at least $2/3$ at time $2n$, from some start state

... and it can be done in coAM

Conclusion

- Our constructions give evidence that detecting Markov Chain convergence cannot be automated
- To understand the realistic limitations, it could be good to have more examples and tighter bounds
- Open question: What if the stationary distribution is known (efficiently computable)?

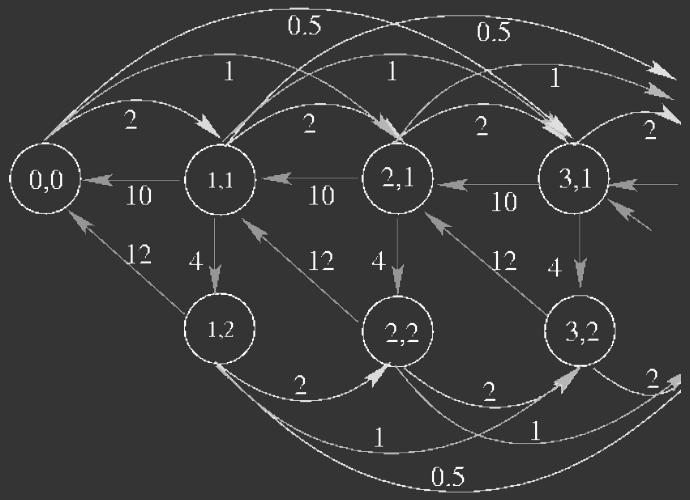
The complexity of distinguishing Markov random fields

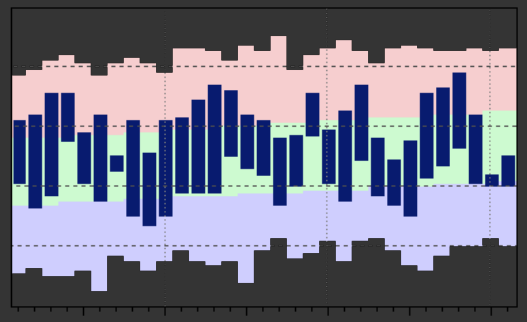
With

Andrej Bogdanov
Tsinghua University and Chinese University of Hong Kong

Salil Vadhan
Harvard University

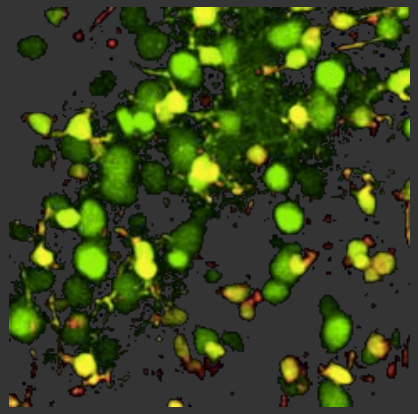
Model reconstruction



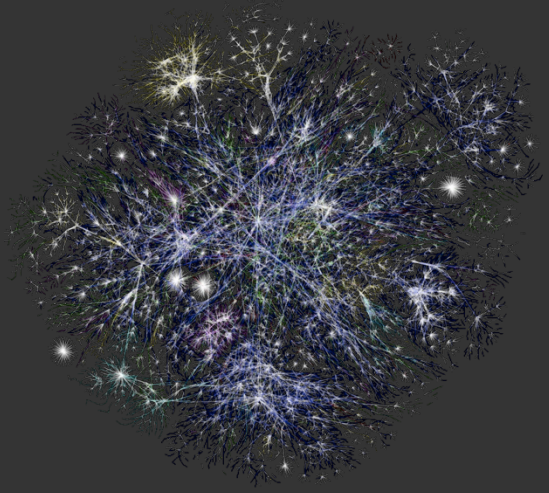


Combinatorial statistics on networks

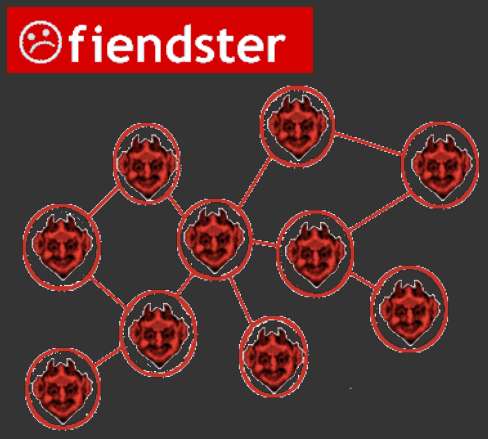
- **In many applications, the system represents interactions in a network**



biology



communications

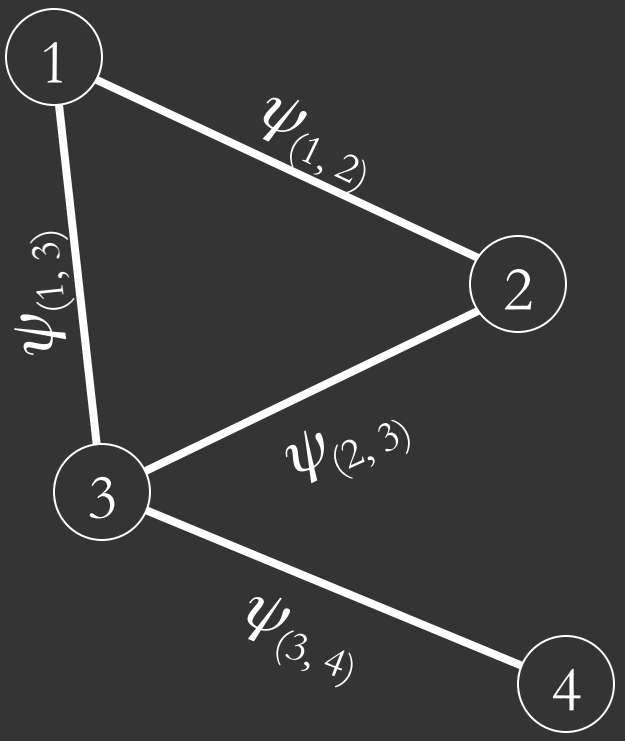


sociology

- **Can we reconstruct the network from observations at the nodes?**

Markov random fields

- **A common model for stochastic networks**

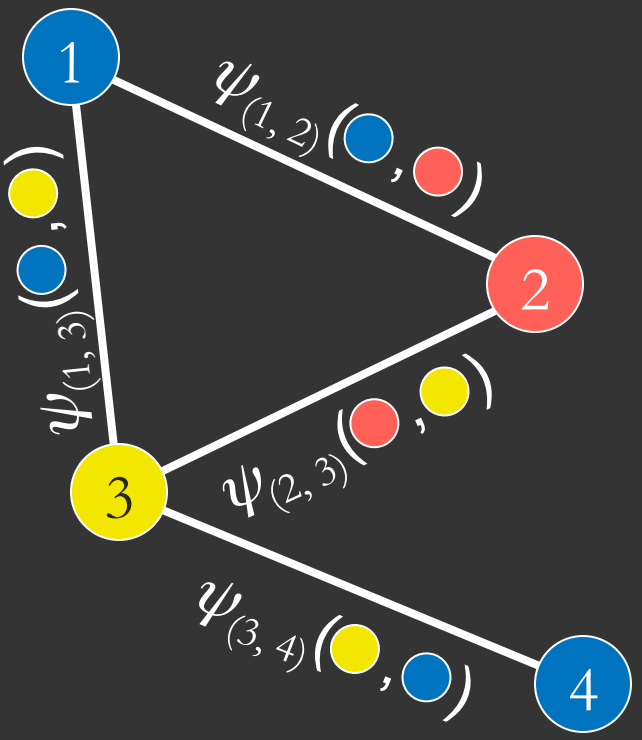


bounded degree graph $G = (V, E)$

weight functions ψ_e: $\Sigma^2 \rightarrow \mathbf{R}^{\geq 0}$
for every edge e

Markov random fields

- **A common model for stochastic networks**



bounded degree graph $G = (V, E)$

weight functions ψ_e: $\Sigma^2 \rightarrow \mathbf{R}^{\geq 0}$
for every edge e

nodes v are assigned
values a_v in alphabet Σ

distribution over states
$\sigma \in \Sigma^V$ given by

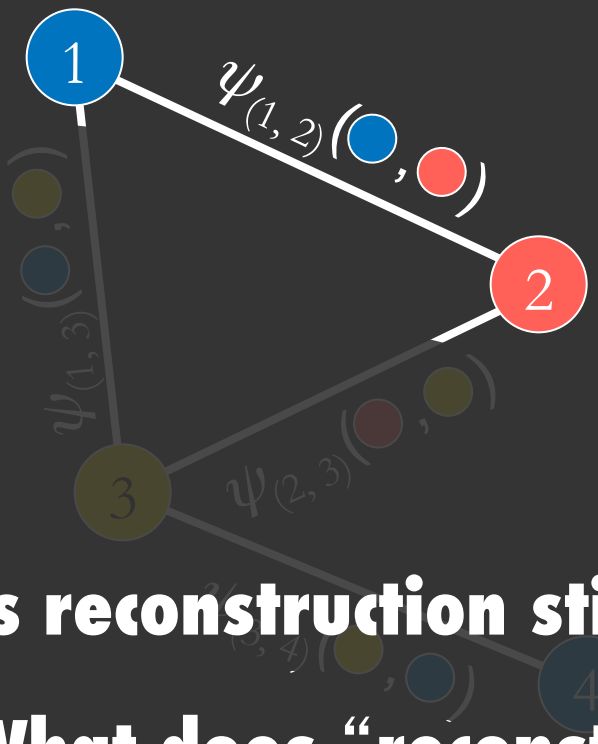
$$\Pr[\sigma] \sim \prod_{(u,v) \in E} \psi_{(u,v)}(a_u, a_v)$$

Reconstruction task for Markov random fields

- Suppose we can obtain independent samples from the Markov random field
- Given observed data at the nodes, is it possible to reconstruct the model (network)?

Hidden nodes

- **In some applications only some of the nodes can be observed**



visible nodes $\mathbb{W} \subseteq \mathbb{V}$

Markov random field over visible nodes is

$$\sigma_{\mathbb{W}} = (\sigma_w : w \in \mathbb{W})$$

- **Is reconstruction still possible?**
- **What does "reconstruction" even mean?**

Reconstruction versus distinguishing

- We are interested in computational obstacles for efficient reconstruction
- Reconstruction is related to learning.
- To provide evidence for hardness, we look at the easier problem of distinguishing models

Distinguishing problems

- Let M_1, M_2 be two models with hidden nodes

PROBLEM 1

- Can you tell if M_1 and M_2 are statistically close or far apart (on the visible nodes)?

PROBLEM 2

- Assuming M_1 and M_2 are statistically far apart and given access to samples from one of them, can you tell where the samples came from?

Main result

Problems 1 and 2 are intractable (in the worst case) unless $\mathrm{NP} = \mathrm{RP}$

- **Conversely, if** $\mathrm{NP} = \mathrm{RP}$ **then distinguishing (and other forms of reconstruction) are achievable**

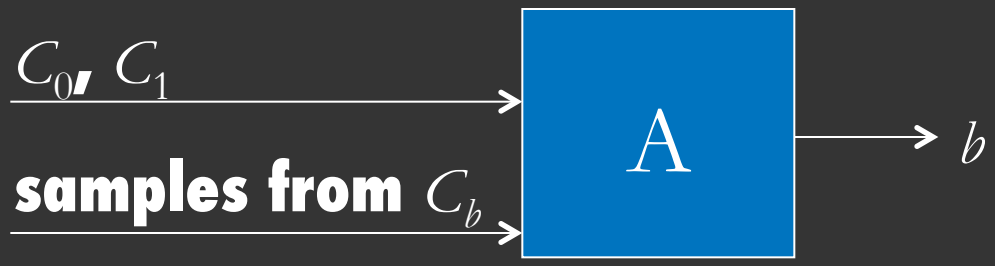
Reduction to circuits

- **Markov random fields can simulate the uniform distribution UC over satisfying assignments of a boolean circuit C**

$$\mathbf{pr}_{UC}(x) = \begin{cases} 1/\#\mathrm{SAT}(C), \text{ if } C(x) = \mathrm{TRUE} \\ 0, \text{ if } C(x) = \mathrm{FALSE} \end{cases}$$

Hardness of distinguishing circuits

- Assume you have an algorithm A such that



  - If the samples come from another distribution, A can behave arbitrarily

- We use A to find a satisfying assignment for any circuit $C: \{0,1\}^n \rightarrow \{0,1\}$

Hardness of distinguishing circuits

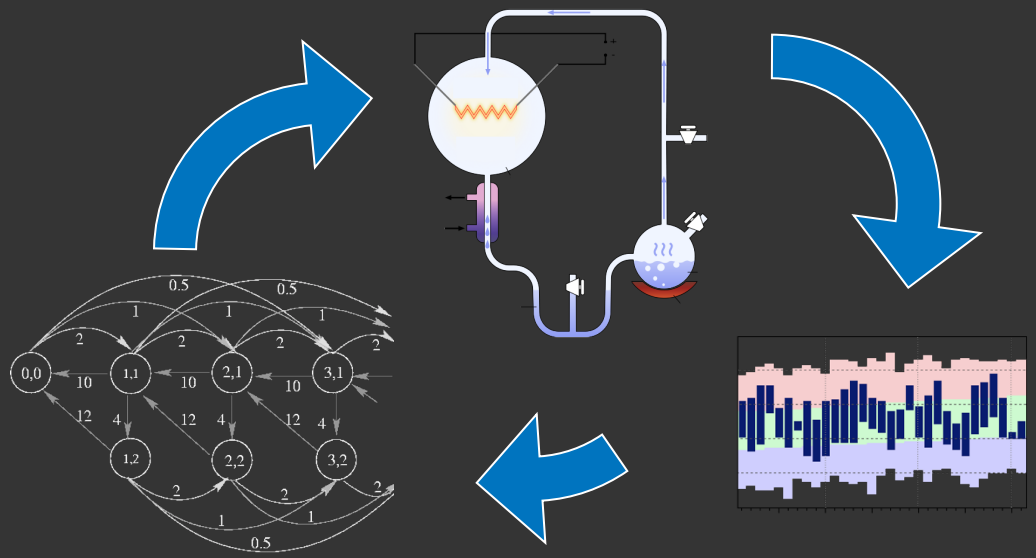
$$C_0(x_1, x_2, \ldots, x_n) = C(x_1, x_2, \ldots, x_n)$$
$$C_1(x_1, x_2, \ldots, x_n) = C(\overline{x_1}, x_2, \ldots, x_n)$$

visible inputs: x_1 **hidden inputs:** $x_2,\ldots,x_n$

– **Proof reminiscent of argument that** $\mathrm{NP} \cap \mathrm{coNP}$ **has** NP**-hard promise problems [Even-Selman-Yacobi]**

A possible objection

- The "hard" models M_1, M_2 describe distributions that are **not efficiently samplable**



- But if nature is efficient, we never need to worry about such distributions!

Two Models of a Biologist

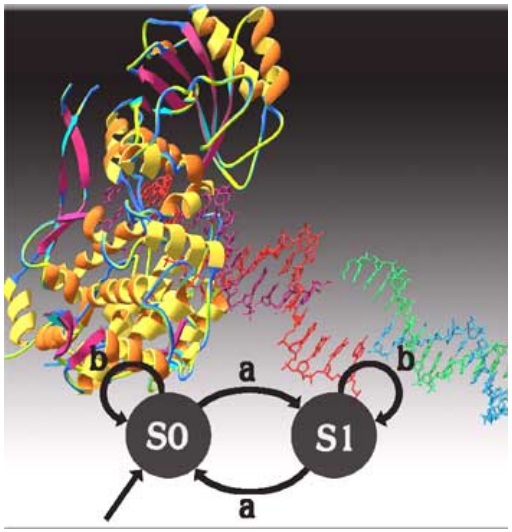
- The Computationally Limited Biologist: Cannot solve hard computational problems, in particular cannot sample from a general G-distributions.

- The Computationally Unlimited Biologist:

  Can sample from any distribution.

- Related to the following problem:

  Can nature solve computationally hard problems?



From Shapiro at Weizmann

Distinguishing problem for samplable distributions

PROBLEM 3

- If M_1 and M_2 are statistically far apart and given access to samples from one of them, can you tell where the samples came from, **assuming M_1 and M_2 are efficiently samplable**?
- Theorem

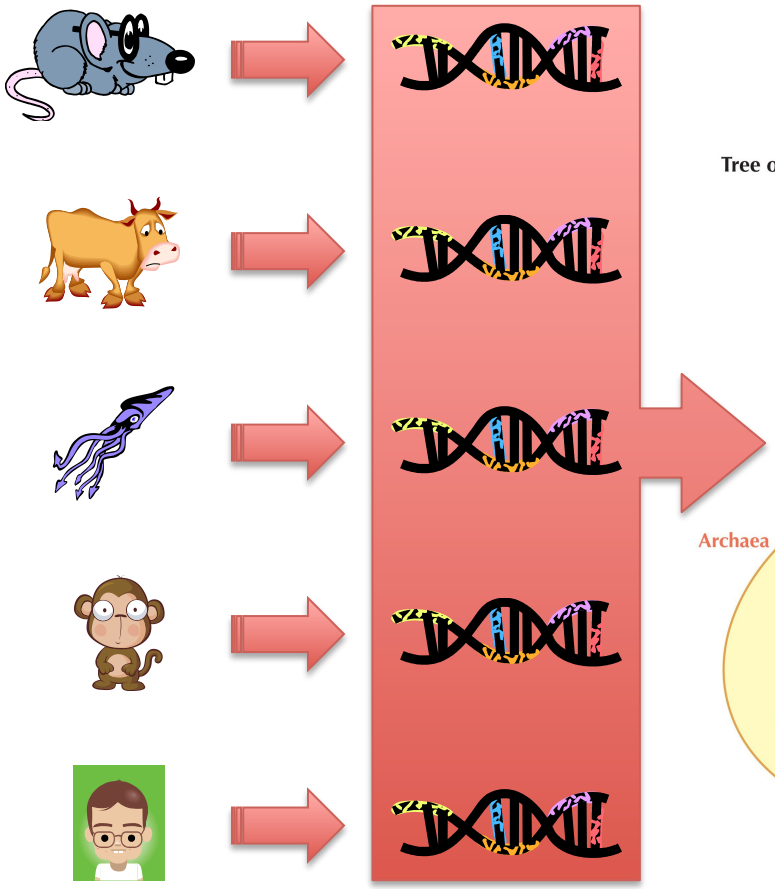
Problem 3 is intractable unless computational zero knowledge is trivial

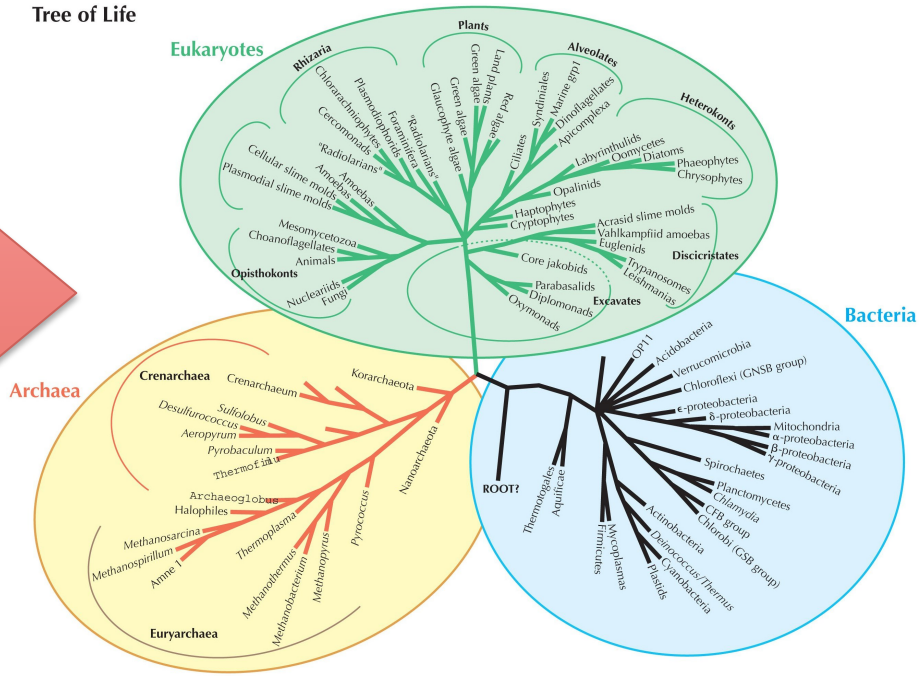
  - We don't know if this is tight

Phylogenetic Mixtures: The good & the bad

Based on joint work with
S, Roch (UCLA)

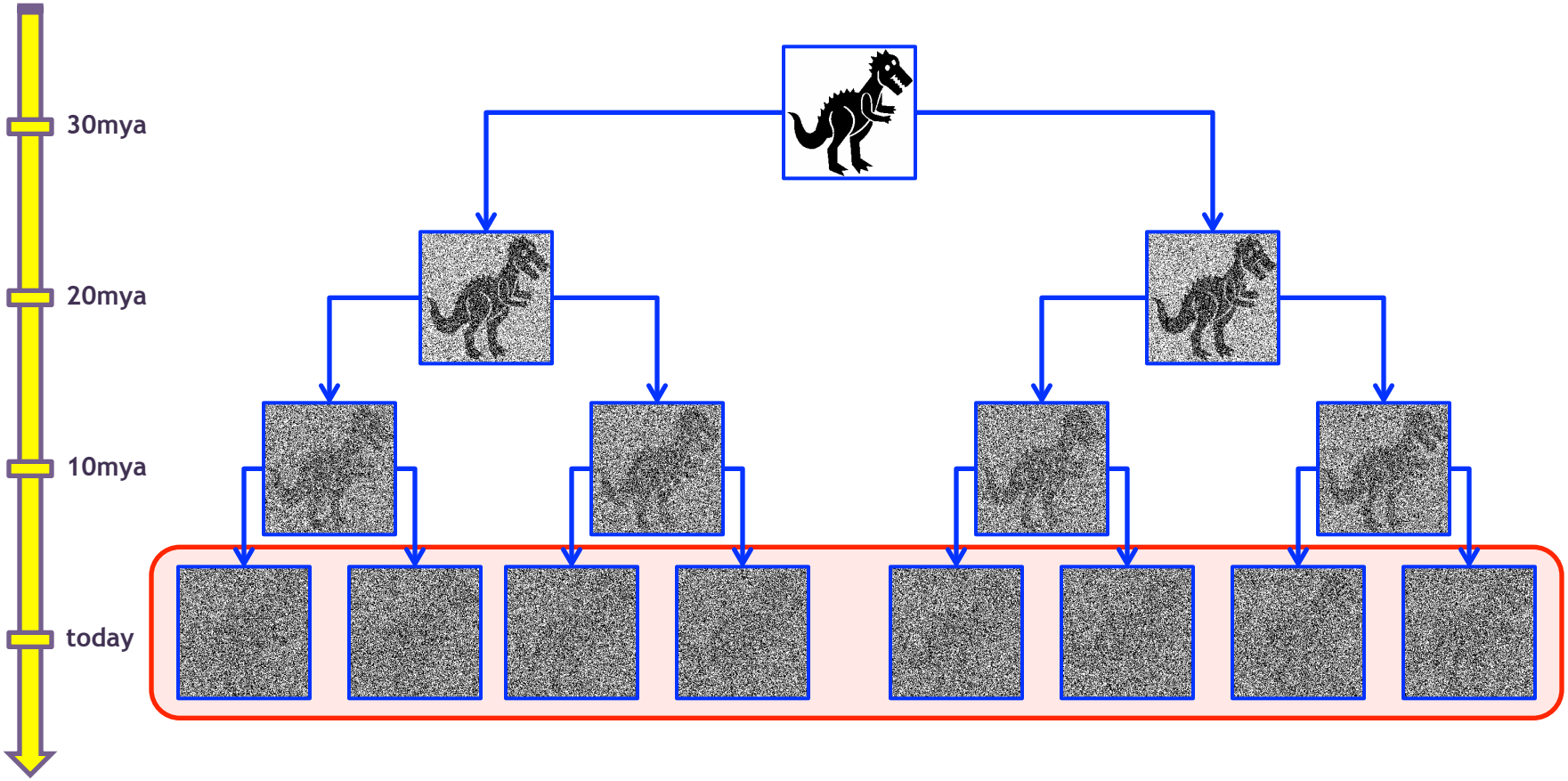
Of Mice and Men





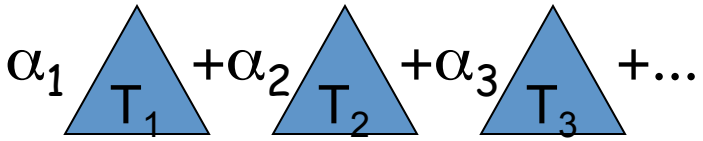
Broadcasting DNA

(Probabilistic Point of View)



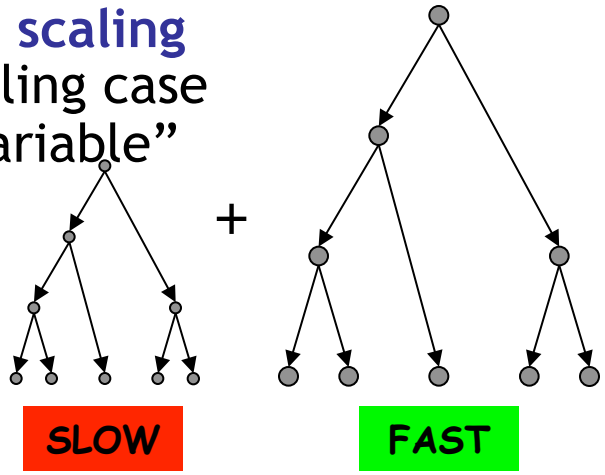
heterogeneous data

- **phylogenetic mixtures** – definition by picture:

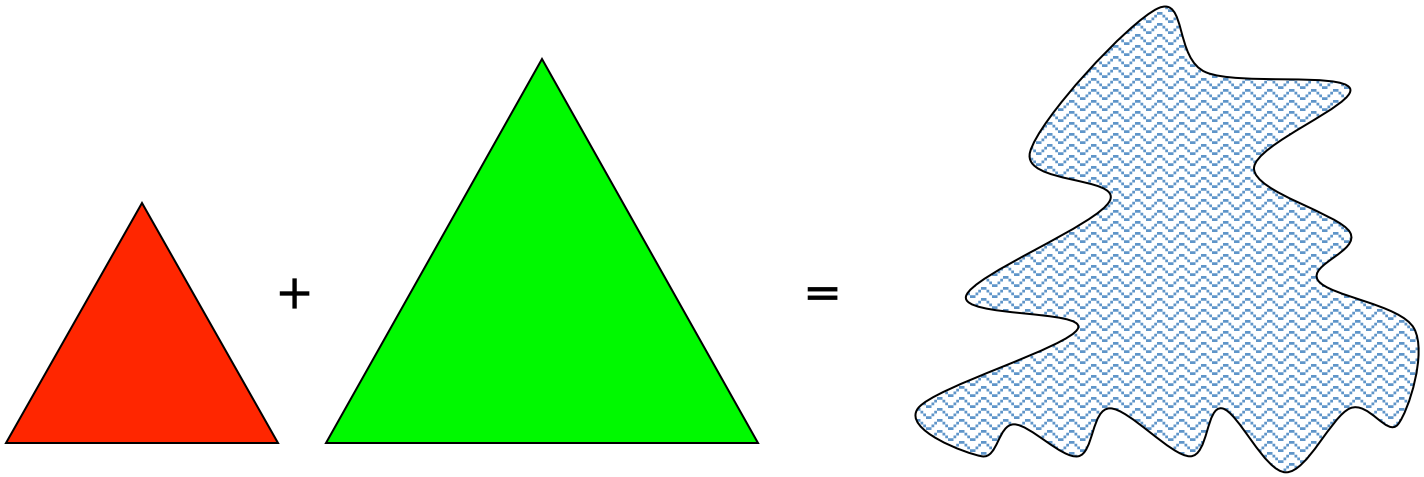


- **special case** – "rates-across-sites"
  - trees are the same up to **random scaling**
  - in this talk, will focus on two-scaling case
  - can think of scaling as "hidden variable"

- **biological motivation**
  - heterogeneous mutation rates
  - inconsistent lineage histories
  - hybrid speciation, gene transfer
  - corrupted data

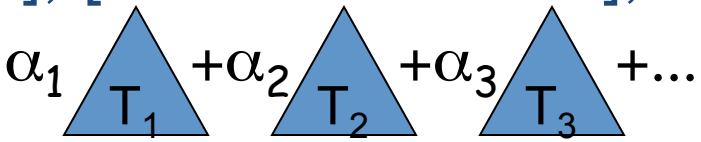


but, on a mixture…



why are mixtures problematic?

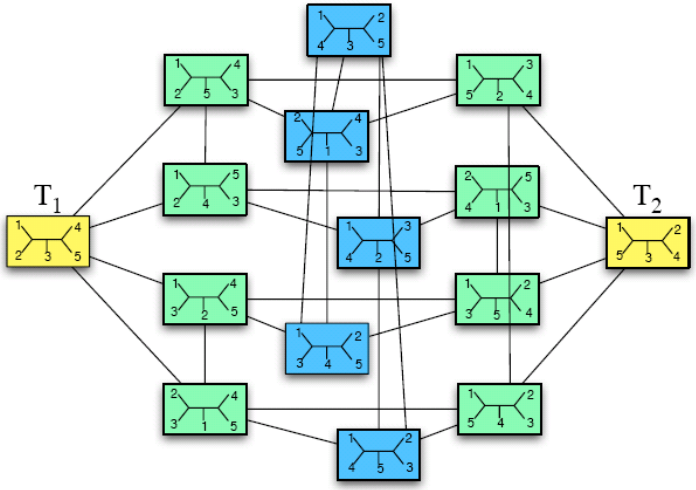
- **identifiability** – does the distribution at the leaves determine the α's and T's?
  - negative results: e.g. [Steel et al.'94], [Stefankovic-Vigoda'07], [Matsen-Steel'07], etc.
  - positive results: e.g. [Allman, Rhodes'06,'08], [Allman, Ane, Rhodes'08], [Chai-Housworth'10], etc.

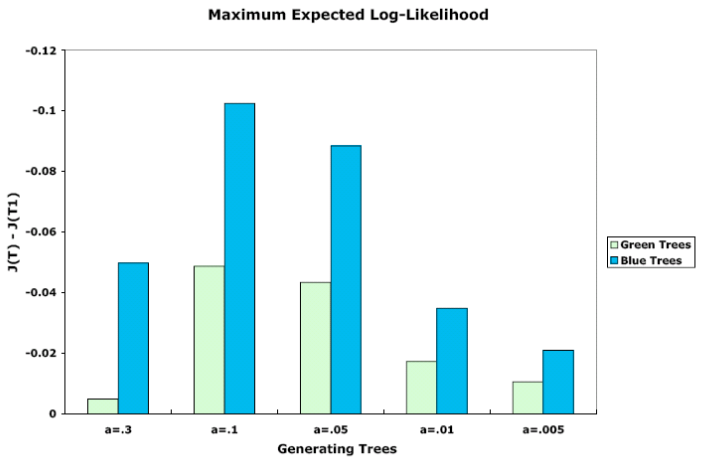


- **algorithmic** – assuming identifiability, can we reconstruct the topologies efficiently?
  - can mislead standard methods;
  - ML under the full model is consistent in identifiable cases;

The Pitfalls of Generic Techniques for Mixtures

- Note: Algorithm design is needed for guarantees with realistic sequence length and running time.
- Currently generic Techniques (Bayesian, ML, Parsimony) have no known guarantees in terms of running time / sequence length.
- In fact in [M-Vigoda' (Science 05, Ann. App. Prob. 06)]: Bayesian techniques are misleading for mixtures (assuming no-mixture).





a new site clustering approach

new results [M-Roch, 2011] – we give a simple way to **determine** which sites come from which component

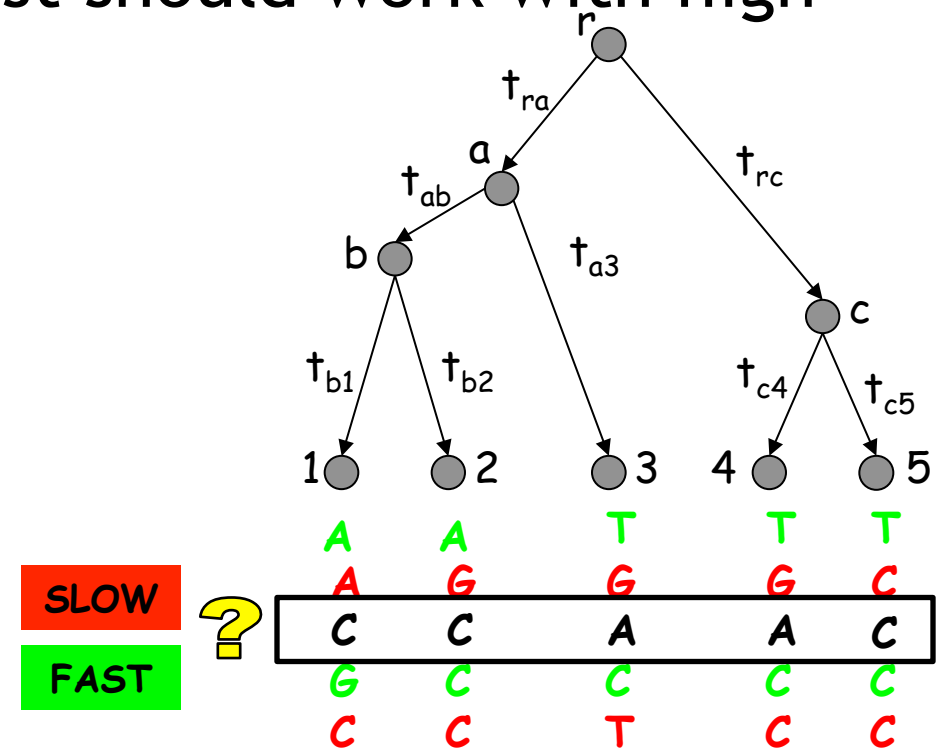
- based on concentration of measure in large-tree limit

site clustering

- ideally, guess **which sites** were produced by **each component**

  scaling is “hidden” but we can try to infer it

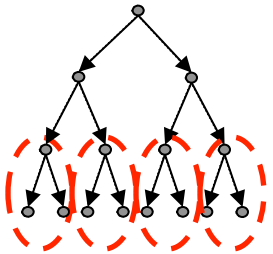
  – to be useful, a test should work with high confidence



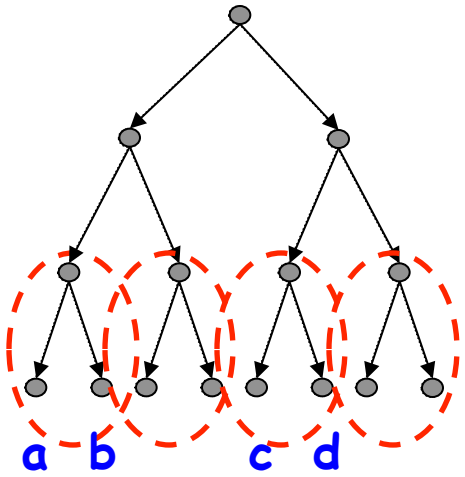
leaf agreement

- **a natural place to start** - impact of scaling on **leaf agreement**
  - one pair of leaves is not very informative
  - we can look at many pairs

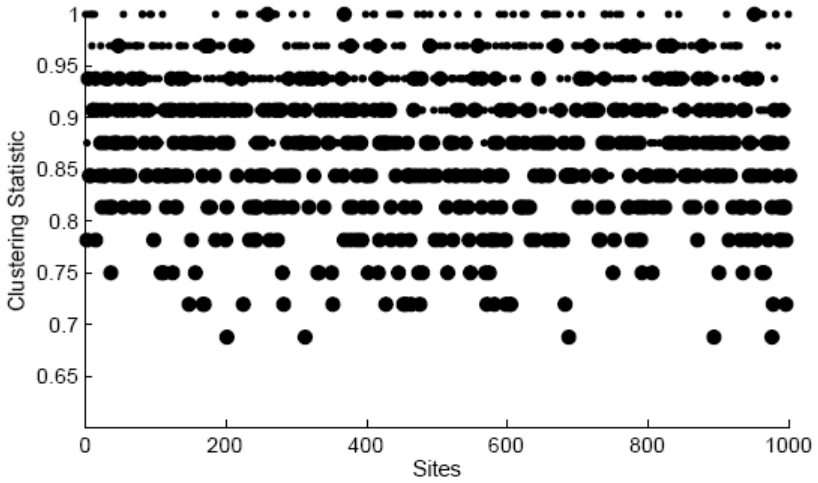
$$C = \sum_{(a,b)\in R\subseteq L^2} \mathrm{I}\{s_a = s_b\}$$

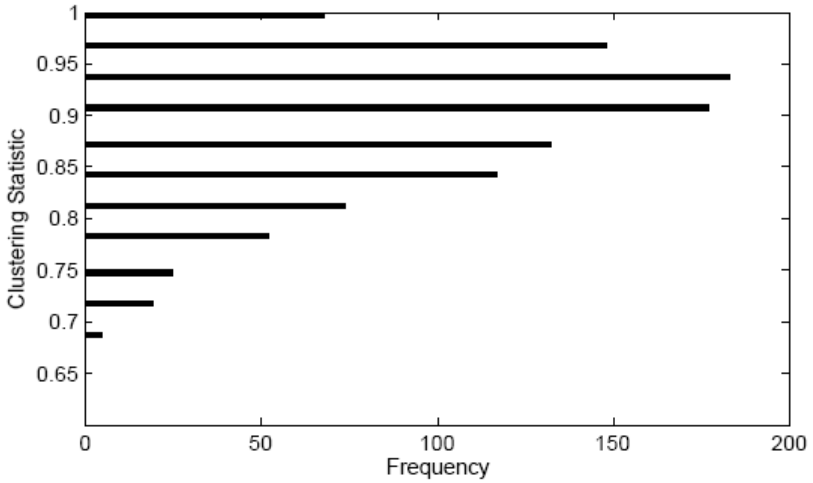


- we would like C to be **concentrated**:
  - large number of pairs
  - each pair has a small contribution
  - independent (or almost independent) pairs
  - nice separation between SLOW and FAST

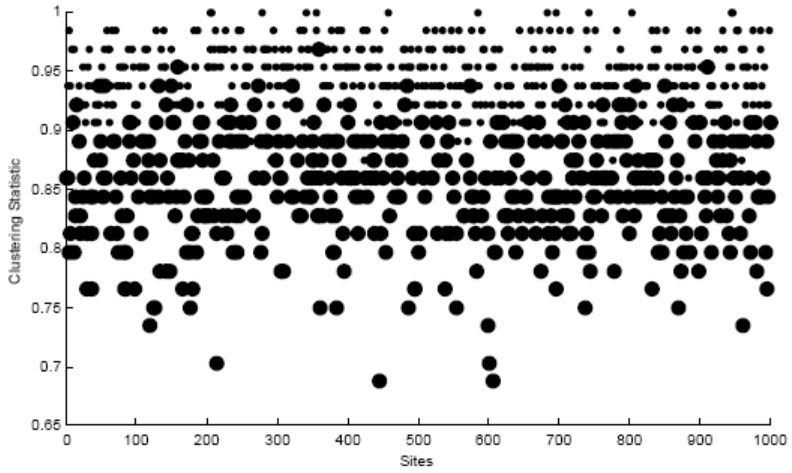


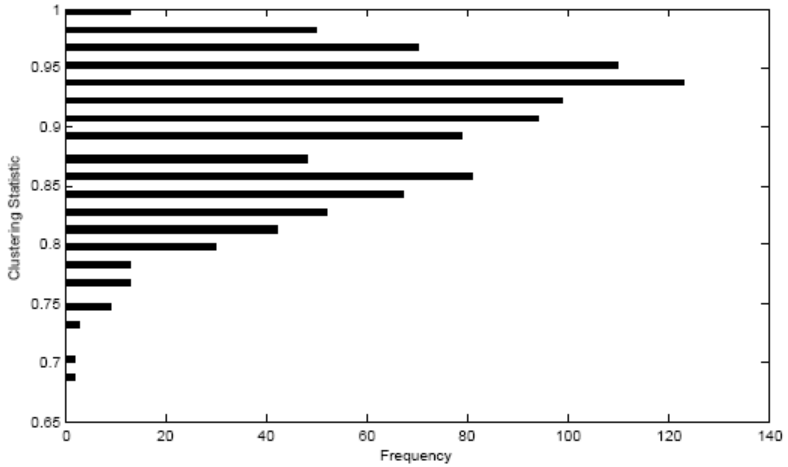
64 leaves


Clustering Statistic
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
200
400
600
800
1000
Sites

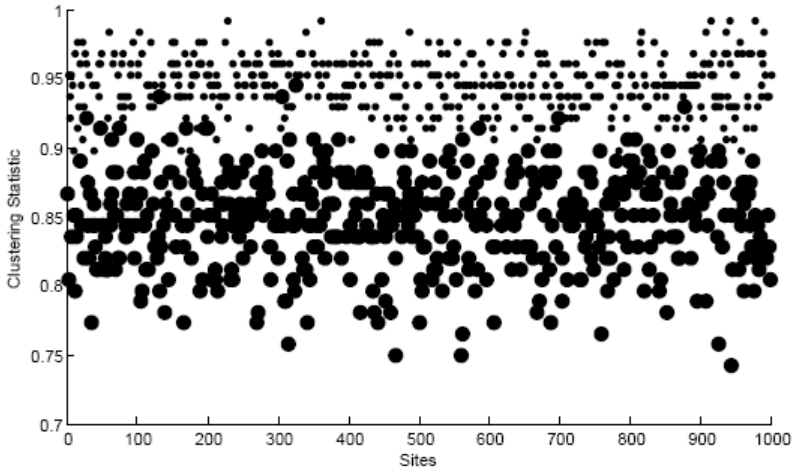

Clustering Statistic
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
50
100
150
200
Frequency

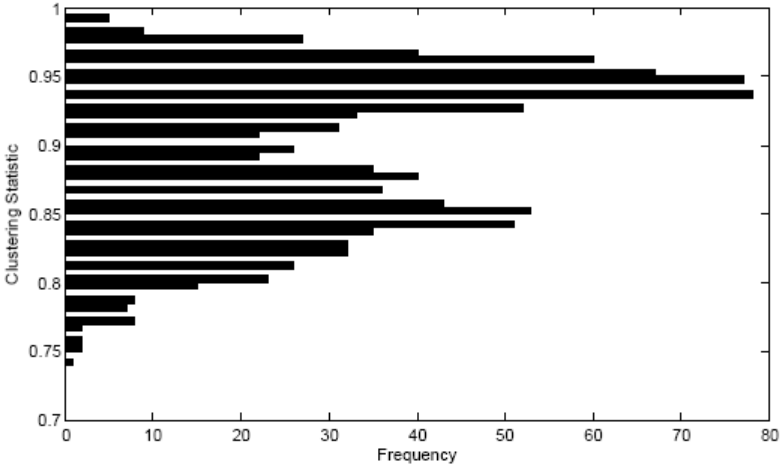
128 leaves


Clustering Statistic
Sites
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
100
200
300
400
500
600
700
800
900
1000

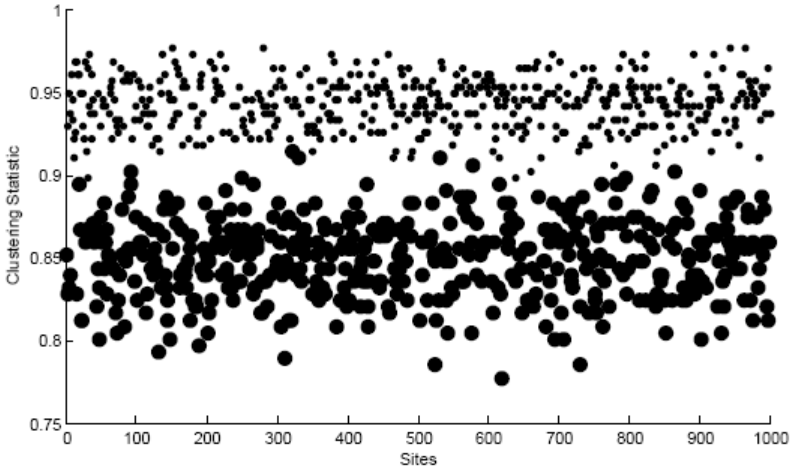

Clustering Statistic
Frequency
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
20
40
60
80
100
120
140

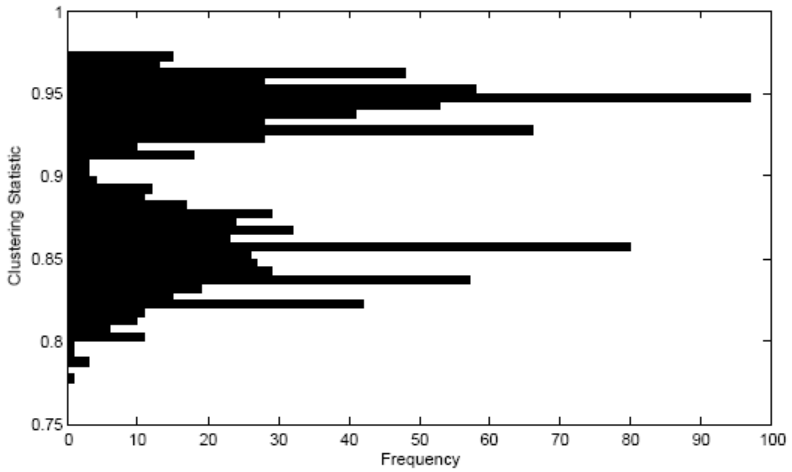
256 leaves


Clustering Statistic
Sites
1
0.95
0.9
0.85
0.8
0.75
0.7
0
100
200
300
400
500
600
700
800
900
1000


Clustering Statistic
Frequency
1
0.95
0.9
0.85
0.8
0.75
0.7
0
10
20
30
40
50
60
70
80

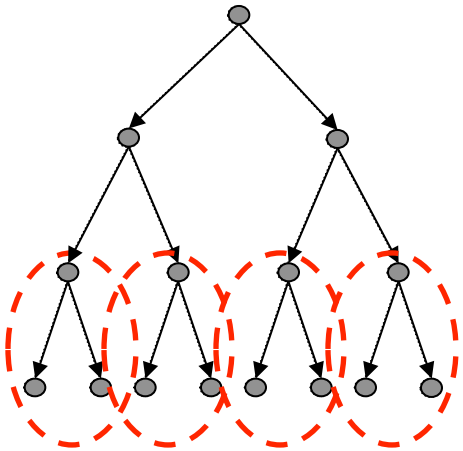
512 leaves


Clustering Statistic
1
0.95
0.9
0.85
0.8
0.75
0
100
200
300
400
500
600
700
800
900
1000
Sites


Clustering Statistic
1
0.95
0.9
0.85
0.8
0.75
0
10
20
30
40
50
60
70
80
90
100
Frequency

but the tree is not complete…

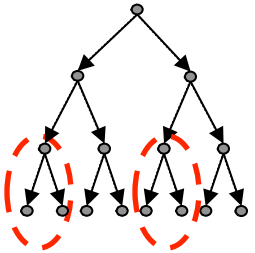
- **lemma 1** – on a general binary tree, the set of all pairs of leaves at distance at most 10 is linear in n
  - proof: count the number of leaves with no other leaves at distance 5
- **lemma 2** – in fact, can find a linear set of leaf pairs that are non-intersecting
  - proof: sparsify above $\hat{C} = \sum_{(a,b)\in\hat{R}\subseteq L^2} \mathrm{I}\{s_a = s_b\}$
- this is enough to build a concentrated statistic



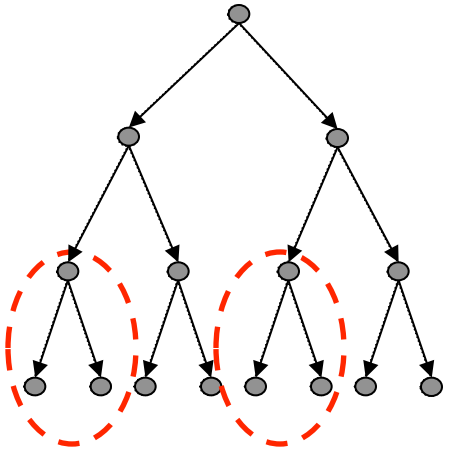
but we don't know the tree…

- **a simple algorithm** – cannot compute exact distances but can tell which pairs are more or less correlated
  - find "close" pairs
  - starting with one pair, remove all pairs that are too close
  - pick one of the remaining pairs and repeat

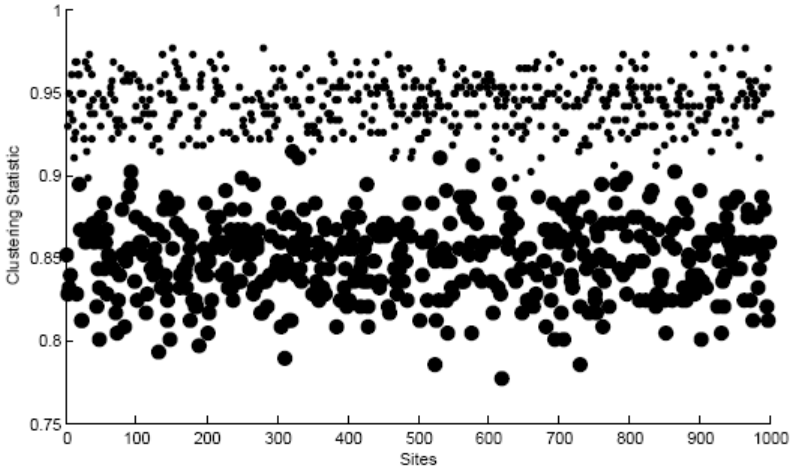
$$\hat{C} = \sum_{(a,b)\in\hat{R}\subseteq L^2} \mathrm{I}\{s_a = s_b\}$$

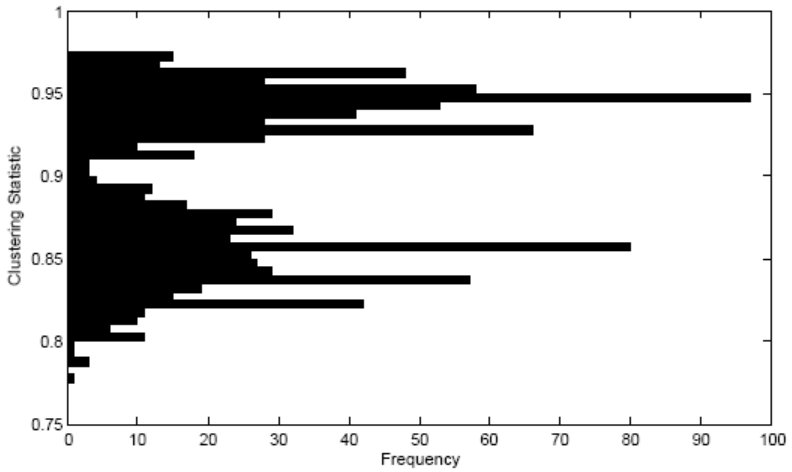


- **claim** – this gives a nicely concentrated variable (for large enough trees)
  - large number of pairs
  - independent (or almost independent) pairs
  - nice separation between SLOW and FAST



site clustering + reconstruction





summary

Proposition 4 (Site Clustering: RAS-JC Model) *Under the assumptions stated in Section 2 on the RAS-JC model, for any given tolerance on the mutation and mixture parameters, there exists a high-confidence site clustering algorithm.*

Proposition 5 (Full Reconstruction: RAS-JC Model) *Under the assumptions stated in Section 2 on the RAS-JC model, for any given tolerance on the mutation and mixture parameters, there exists a high-probability reconstruction algorithm using polynomial-length sequences and running in polynomial time.*

Efficient sorting of the Mallows model

Based on joint work with

Mark Braverman (Princeton)

Example Consensus Ranking, Rearrangements and the Mallows Model

- Problem 1: Consider a sorting problem where for each query comparing two elements x and y:
- Return correct answer with probability ½+ ε
- Independently for each query.
- Can query each pair as many times as we want.
- How many queries are needed to find correct order with probability 0.9999?
- Answer: Feige, Raghavan, Peled and Upfal.

- Problem 2: Consider a sorting problem where for each query comparing two elements x and y.
- Return correct answer with probability ½+ ε
- Independently for each query.
- Each pair of elements can be queried only once.
- What can we do?

Example Consensus Ranking, Rearrangements and the Mallows Model

- Problem 3 Given a set of permutations (rearrangements) $\{\pi_1, \pi_2, \ldots \pi_N\}$ find the consensus ranking (or central ranking)

$$\pi_0 = \text{argmin} \sum_{i=1}^{N} d(\pi_i, \pi_0)$$

for d = distance on the set of permutations of n objects
Most natural is d_K which is the Kendall distance.

$$d_K(\pi, \text{id}) = \sum_{i<j} 1_{j \prec_\pi i}$$

$$d_K(\pi, \pi') = d_K(\pi(\pi')^{-1}, \text{id}) = \sum_{i \prec_{\pi'} j} 1_{j \prec_\pi i}$$

The Mallows Model – A distribution on rearrangements

- Exponential family model in β:
  - $P(\pi \mid \pi_0) = Z(\beta)^{-1} \exp(-\beta\, d_K(\pi,\pi_0))$
- $\beta \equiv 0$: uniform distribution over permutations
- $\beta > 0$: π_0 is the unique mode of $P\beta_{,\pi 0}$
- This is a re-arrangement model with cost proportional to # of wrong inversions.
- ML estimation is exactly the same as consensus ranking!

- Theorem [Meila,Phadnis,Patterson,Bilmes 07]

  Consensus ranking (i.e ML estimation of π_0 for constant θ) can be solved exactly by a branch and bound (B&B) algorithm.
- The B&B algorithm can take (super-) exponential time in some cases
- Seem to perform well on simulated data.

Related work

$$P_{\theta,\pi_0}(\pi) = \frac{1}{Z(\theta)} e^{-\sum_j \theta_j V_j(\pi \pi_0^{-1})}$$

ML estimation

- [Fligner&Verducci 86] introduce generalized Mallows model, θ^{ML} estimation
- [Fligner&Verducci 88] (FV) heuristic for π_0 estimation

Consensus ranking

- [Cohen,Schapire,Singer 99] Greedy algorithm (CSS)
  - + improvement by finding strongly connected components
  - + missing data (not all π_i rank all n items)
- [Ailon,Newman,Charikar 05] Randomized algorithm
  - guaranteed 11/7 factor approximation (ANC)
- [Mathieu, 07] $(1+\varepsilon)$ approximation, time $O(n^6/\varepsilon^{+2\text{^}2O(1/\varepsilon)})$
- [Davenport,Kalagnanan 03] Heuristics based on edge-disjoint cycles
- [Conitzer,D,K 05] Exact algorithm based on integer programming, better bounds

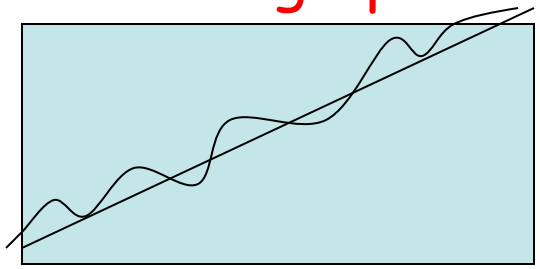
Efficient Sorting of Mallow's model of rearrangements (problem 3)

- [Braverman-Mossel-09]:

- Given r independent samples from the Mallows Model, find ML solution exactly! in time n^b, where
- $b = 1 + O((\beta r)^{-1})$,
- where r is the number of samples
- with high probability (say $\geq 1-n^{-100}$)

Sorting with nois (Problem 2)

- [Braverman-M-09]: In raking from ranking comparisons without repetitions find ML order in $O(n \log n)$ queries and time n^r where $r=O(1/\varepsilon^6)$ time.
- Proof Ingredient 1: "statistical properties" of generated permutations π_i in terms of the original order π_0 :
- With high probability: $\sum_x |\pi_i(x)-\pi(x)| = O(n)$,
  $\max |\pi_i(x)-\pi(x)| = O(\log n)$

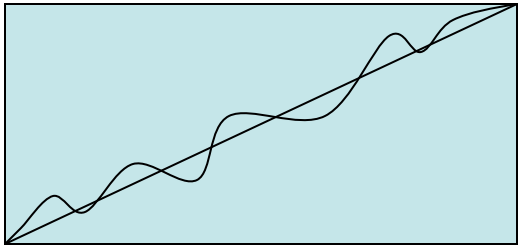


- Additional ingredient: A dynamic programming algorithm to find π given a starting point where each elements is at most k away with running time $O(n\, 2^{6k})$

Sorting the Mallows model (Problem 3)

- [Braverman-M-11]: Optimal order can be found in polynomial time and $O(n \log n)$ queries.
- Proof Ingredient 1: "statistical properties" of generated permutations π_i in terms of the original order π_0 :
- With high probability: $\sum_x |\pi_i(x)-\pi(x)| = O(n)$,

  $\max |\pi_i(x)-\pi(x)| = O(\log n)$



- Additional ingredient: A dynamic programming algorithm to find π given a starting point where each elements is at most k away with running time $O(n\, 2^{6k})$

The algorithm assuming small deviation

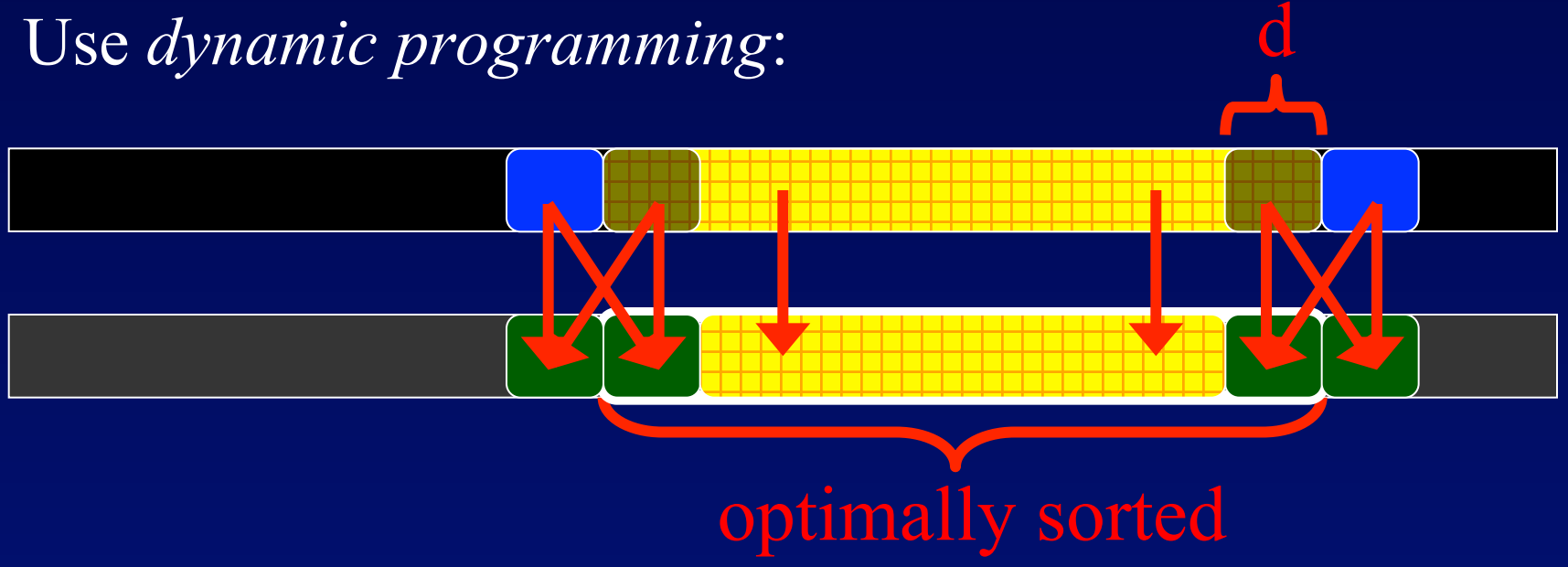
- Insert elements in some random order $a_1, a_2, \ldots, a_n$. For every $k \leq n$, there is an optimal order π_k and the original order I_k.
- Using union bound, for each k we have

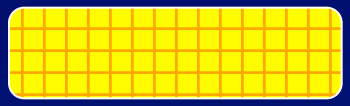
$$(*)\ \max |\, i - \pi_k(i)| = O(\log n).$$

- Find the π_k by inserting the elements one by one.

The algorithm assuming small deviation

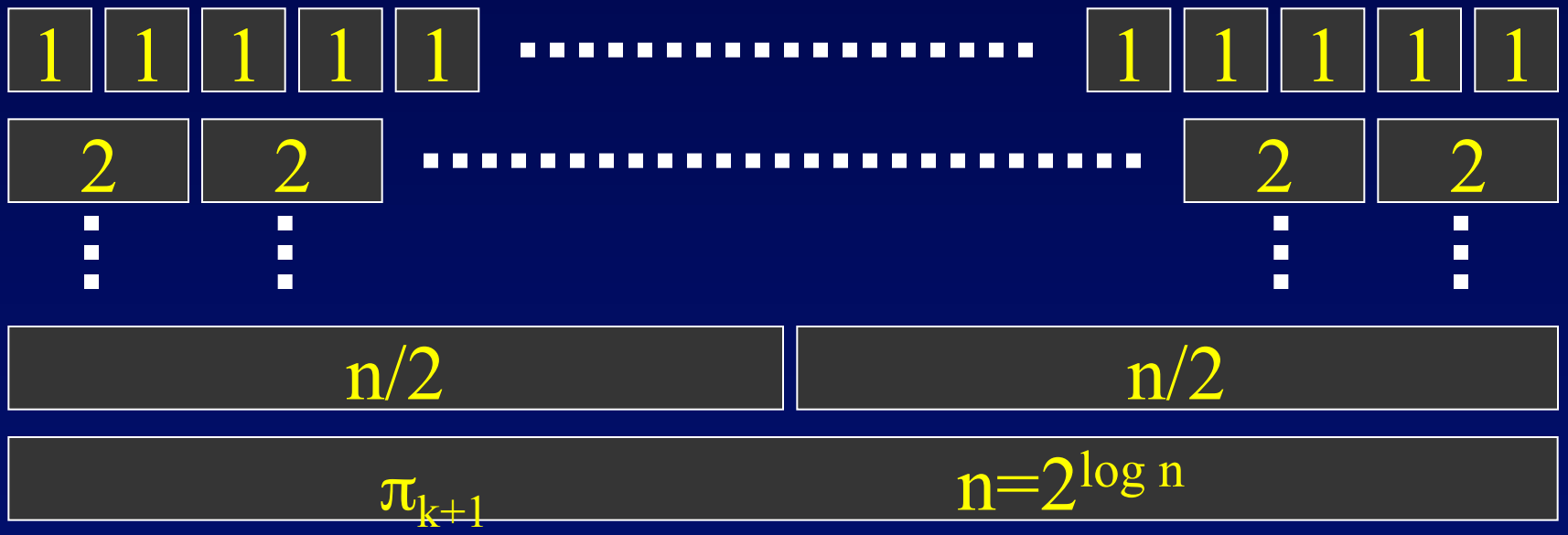
- The problem now: Find the optimal ordering π_{k+1} such that each element is at most $d = O(\log n)$ away from its position in π'.
- Use *dynamic programming*:



- For each interval  there are $<2^{4d}$ "variations".
- A total of poly(n) variations, can store all of them.

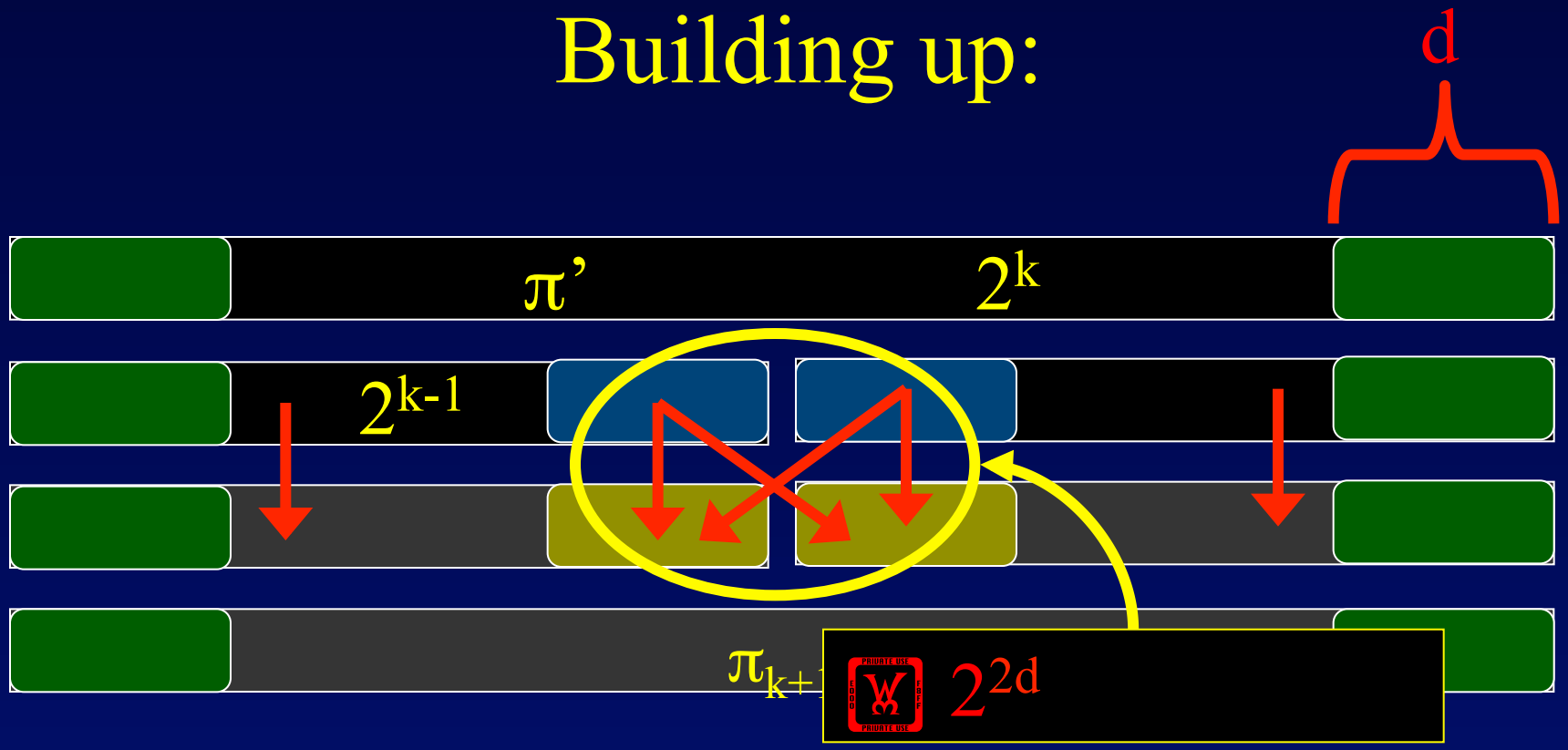
The algorithm assuming small deviation

- Store optimal permutations of all variations on the following intervals:



- A total of $\tilde{O}(2^{4d} n)$ storage.
- Work from shorter intervals to longer.

Building up:



- Each of the shorter intervals has been pre-sorted.
- Thus the cost of doing all intervals on level k is

  #intervals × #checks × #cost/check = $(n/2^k)\ 2^{4d} \times 2^{2d} \times 2^{2k}$.
- Thus, total running time is bounded by $O(2^{6d}\ n^2)$.

Conclusion

"Combinatorial Statistics":

Hard to prove results

But do prove them

Know what can and cannot be done

Many Other problems are waiting.

THANK YOU!!