

Applications of SMC to the analysis of partially observed jump processes and: the Entangled Monte Carlo algorithm

Alexandre Bouchard-Côté

University of British Columbia

Collaborators:

Seong-Hwan Jun, Bonnie Kirkpatrick, Liangliang Wang

Part I : Overview

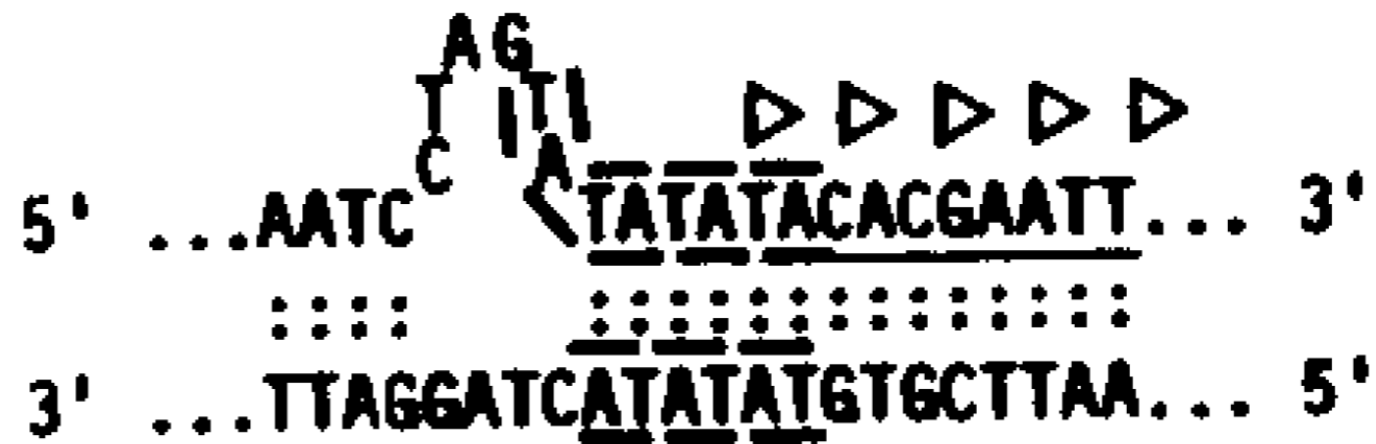
- Problem: posterior inference on countably infinite Continuous Time Markov Chains (CTMCs)
 - Motivations: phylogenetic inference under evolutionary models with random dependencies across sites
- Proposed method:
 - Proposals based on supermartingales on combinatorial potentials
 - Weights given by exponentiation of random matrices

Slipped strand mispairing (SSMs)

Normal pairing
during DNA
replication



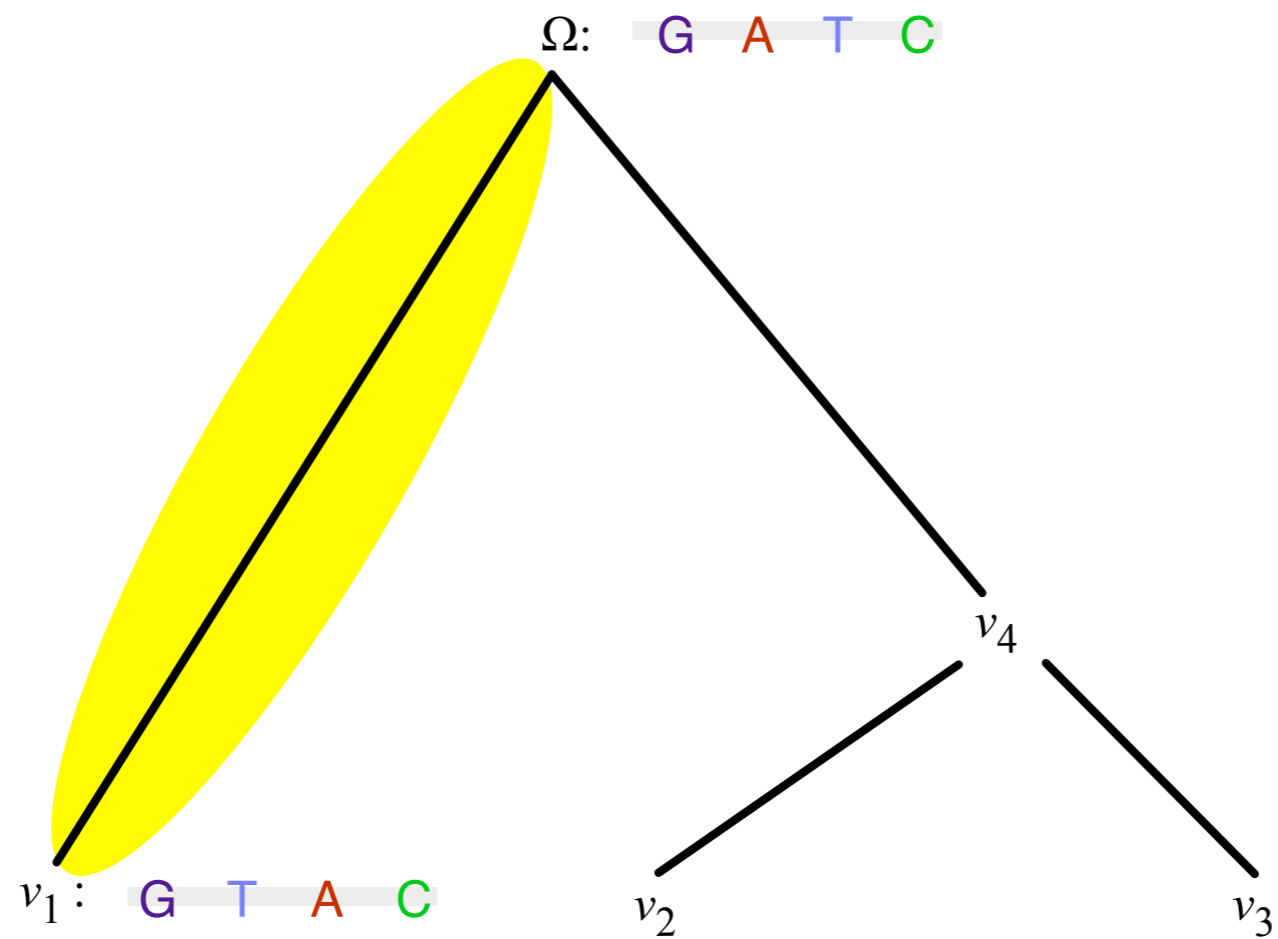
SSM:
Example of
insertion of an
extra TA
repeat



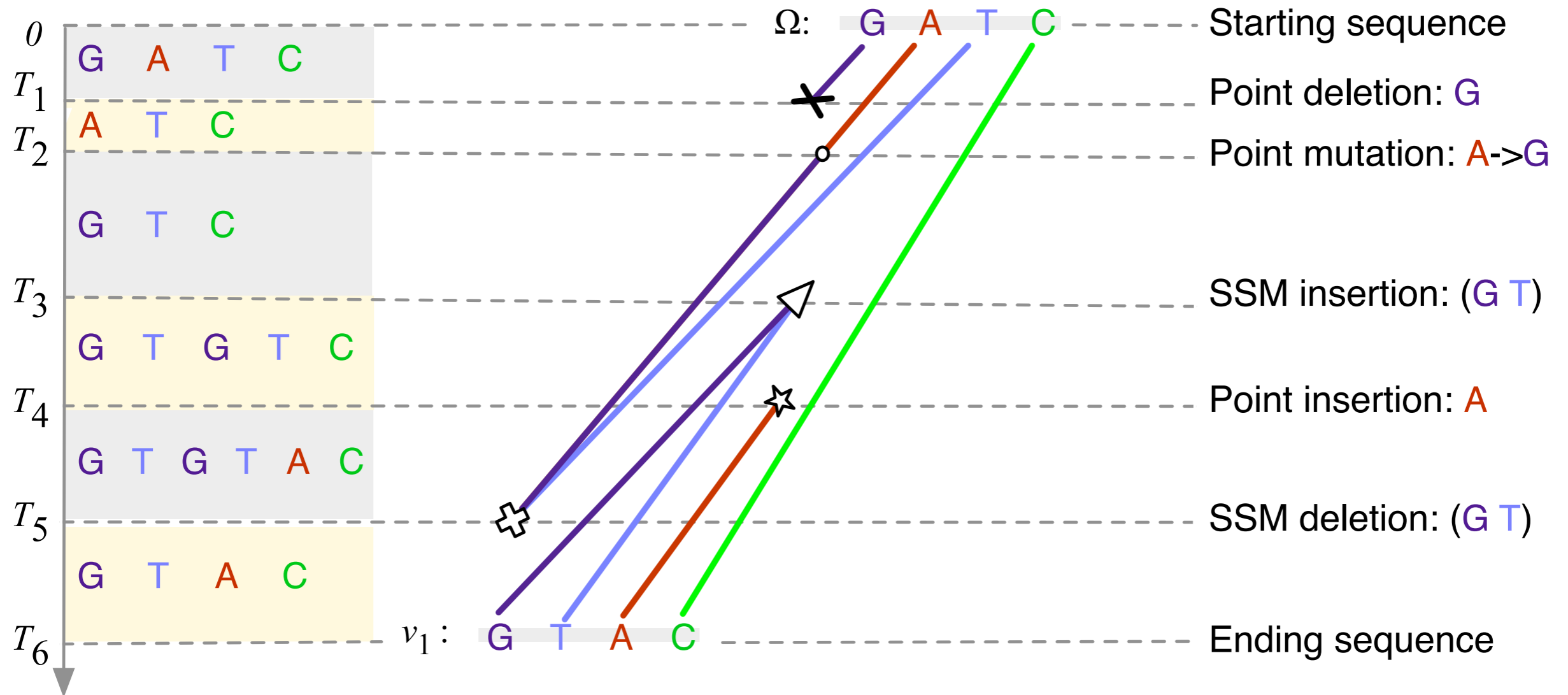
Levinson '87

SSMs on a tree

String-valued branching process:



SSMs on a branch



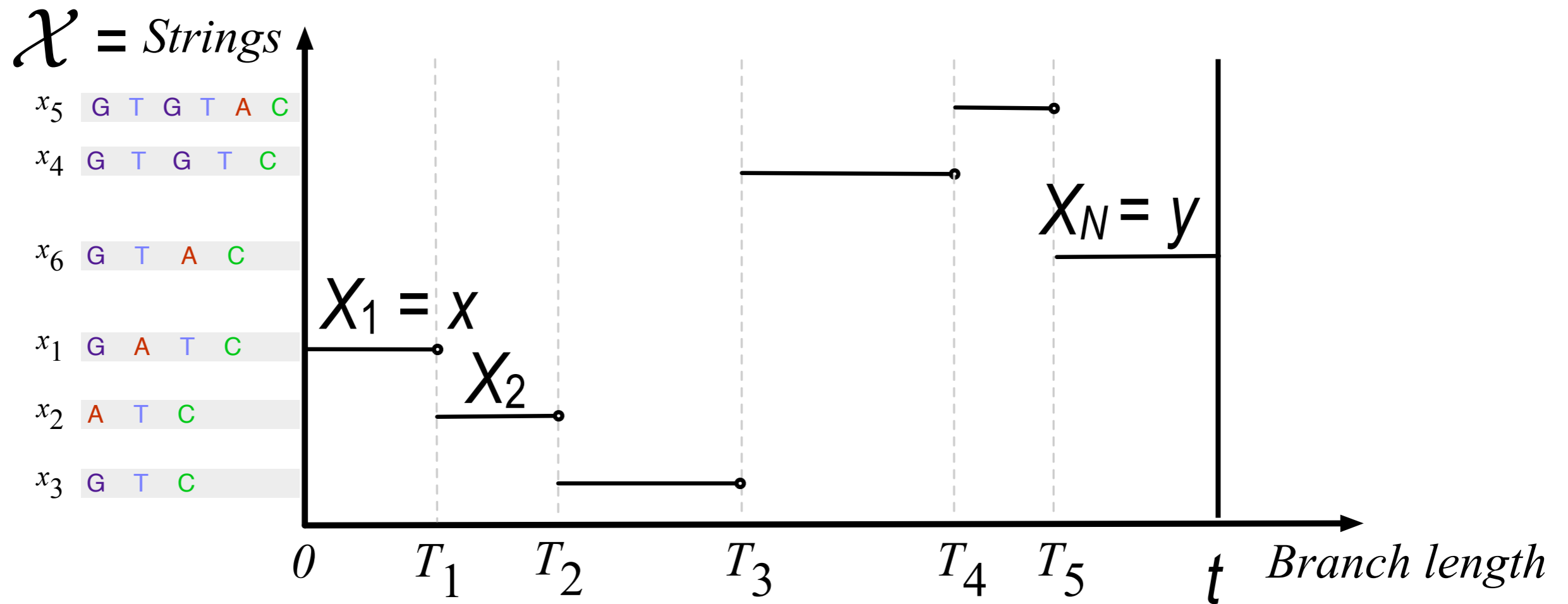
SSMs and phylogenetic inference

- Potential of SSM in phylogenetics:
 - Interactions between SSMs and point mutations adds constraints---this can help resolving trees and alignments
 - Very frequent in neutral regions (e.g. plant introns)
- This potential has not been exploited yet
 - Reason: inference is computationally challenging

Computational problem

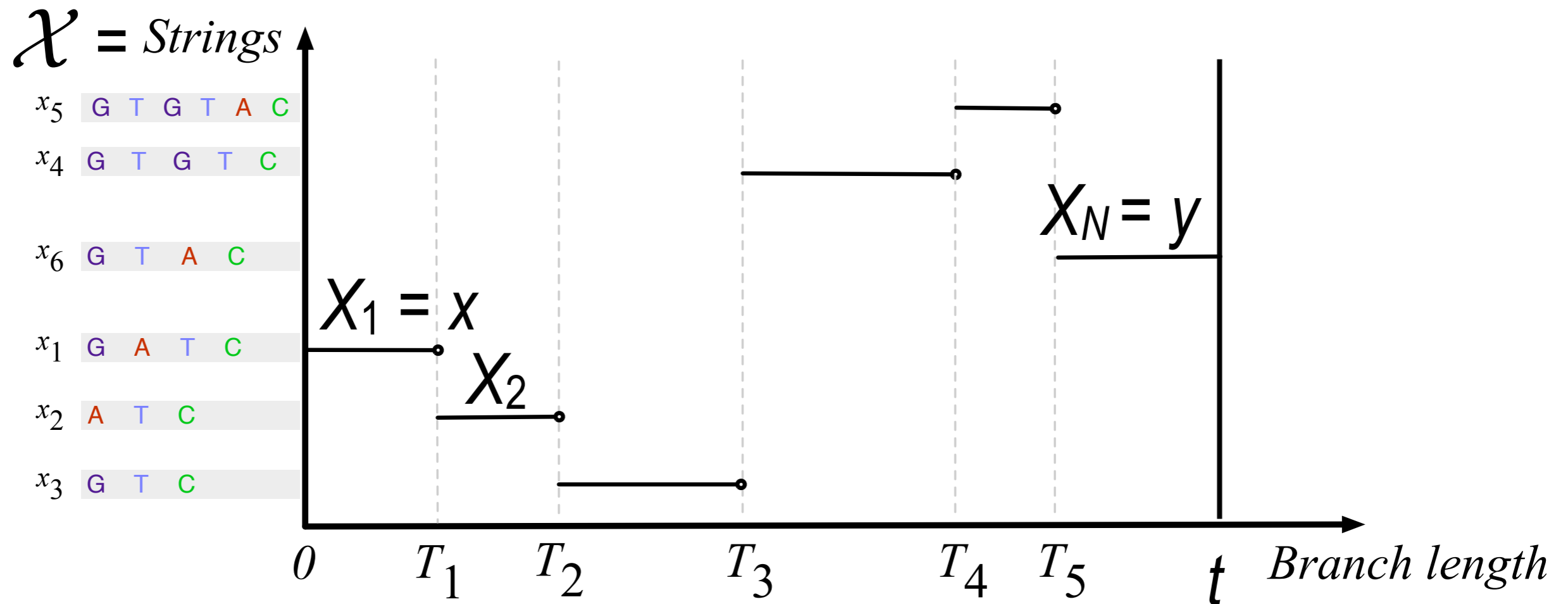
- Our application (phylogenetic tree inference) requires SMC/PMCMC samplers...
- but the main ideas can be explained in a simpler setup:
 - Computing a *marginal transition probability*,
 - using importance sampling

Marginal transition probability



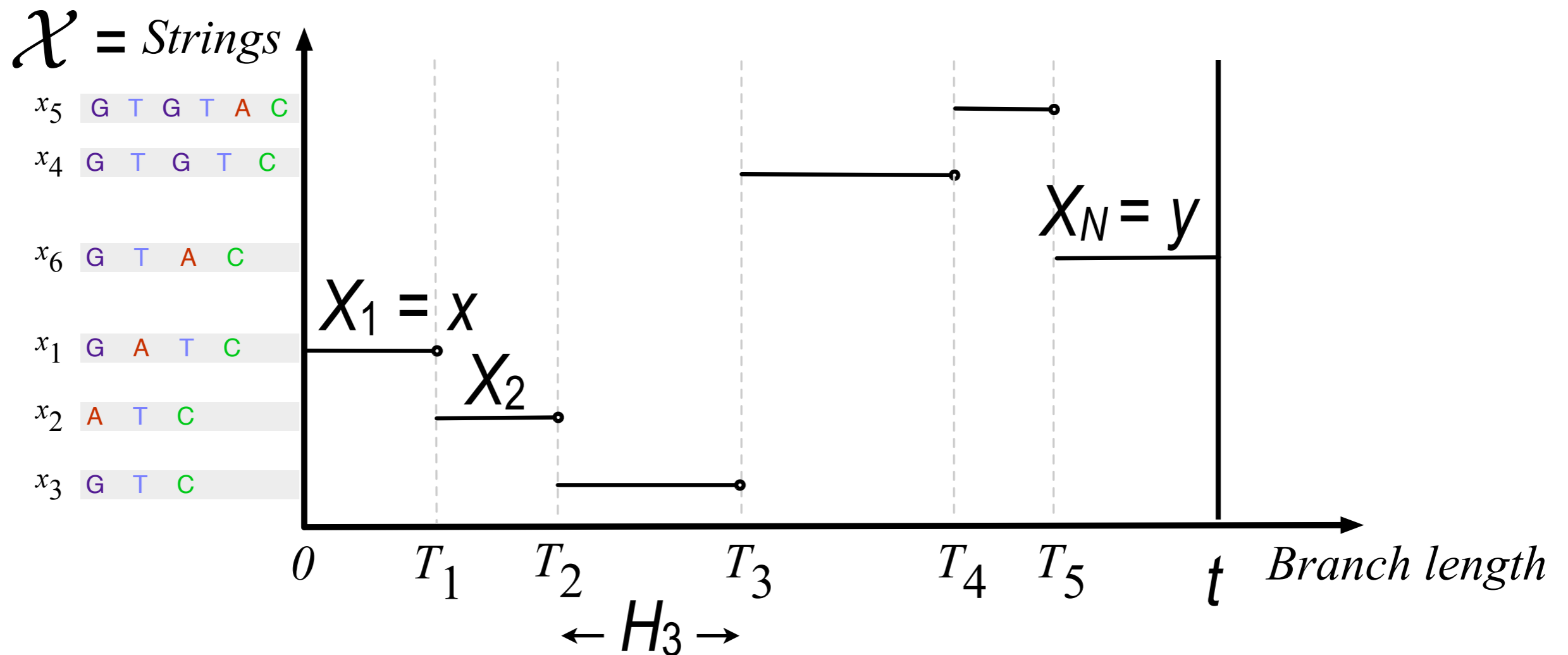
Marginal transition probability

$$\mathbb{P}(X_N = y | X_1 = x)$$



Model

- Jump distribution: $X_{i+1} | X_i \sim \nu_{X_i}$
- Hold times: $H_i \sim \text{Exp}(\cdot)$; $H_i = T_i - T_{i-1}$



Parameters: example

The rate of departing from x : $\lambda : \mathcal{X} \rightarrow (0, \infty)$

$$\lambda(x) = n\theta_{sub} + \lambda_{pt} + n\mu_{pt} + \lambda_{SSM} + f(x)\mu_{SSM}$$

n : length of x ;

$f(x)$: the number of valid SSM deletion locations.

The jumping distribution: $\nu : \mathcal{X} \times \mathcal{F}_\mathcal{X} \rightarrow [0, 1]$

Mutation type from x to x'	
θ_{sub}	Point substitution
$\frac{\lambda_{pt}}{n+1}$	Point insertion
μ_{pt}	Point deletion
$\frac{\lambda_{SSM}}{f(x)}$	SSM insertion
μ_{SSM}	SSM deletion,

$$\nu_x(\{x'\}) = \frac{1}{\lambda(x)}$$

Parameters: example

The rate of departing from x : $\lambda : \mathcal{X} \rightarrow (0, \infty)$

$$\lambda(x) = n\theta_{sub} + \lambda_{pt} + n\mu_{pt} + \lambda_{SSM} + f(x)\mu_{SSM}$$

n : length of x ;

$f(x)$: the number of valid SSM deletion locations.

The jumping distribution: $\nu : \mathcal{X} \times \mathcal{F}_{\mathcal{X}} \rightarrow [0, 1]$

		Mutation type from x to x'
$\nu_x(\{x'\}) = \frac{1}{\lambda(x)}$	{	θ_{sub} Point substitution
		$\frac{\lambda_{pt}}{n+1}$ Point insertion
		μ_{pt} Point deletion
		$\frac{\lambda_{SSM}}{f(x)}$ SSM insertion
		μ_{SSM} SSM deletion,

Note: this is explosion free (always assumed today)

Parameters: exam

Note:
unbounded rate
function

The rate of departing from x : $\lambda : \mathcal{X} \rightarrow (0, \infty)$

$$\lambda(x) = n\theta_{sub} + \lambda_{pt} + n\mu_{pt} + \lambda_{SSM} + f(x)\mu_{SSM}$$

n : length of x ;

$f(x)$: the number of valid SSM deletion locations.

The jumping distribution: $\nu : \mathcal{X} \times \mathcal{F}_x \rightarrow [0, 1]$

Mutation type from x to x'

$$\nu_x(\{x'\}) = \frac{1}{\lambda(x)} \begin{cases} \theta_{sub} & \text{Point substitution} \\ \frac{\lambda_{pt}}{n+1} & \text{Point insertion} \\ \mu_{pt} & \text{Point deletion} \\ \frac{\lambda_{SSM}}{f(x)} & \text{SSM insertion} \\ \mu_{SSM} & \text{SSM deletion,} \end{cases}$$

Note: this is
explosion free
(always assumed
today)

Related work

- Finite case: efficient exact and approximate exponentiation and estimation of rate matrices (Albert 1962; Asmussen *et al* 1996; Hobolth *et al.* 2005; Tataru *et al.* 2011; *inter alia*)
- When the rate function is bounded: Uniformization (Jensen 1953; Hobolth *et al.* 2009; *inter alia*), more recent jump-diffusion inference schemes using thinning for the discrete part (Casella *et al.* 2011, Murray Pollock's talk)
- MCMC approaches (Rao *et al.* 2011)
- Work on countable spaces based on forward simulation (Saeedi *et al.* 2011, Läubli 2011)
- Birth-death processes (Crawford *et al.* 2011; *inter alia*)

Proposed method: notation

State space: list of visited states between end points

$$X = (X_1, X_2, \dots, X_N)$$

Marginalized: transition times

$$T = (T_1, T_2, \dots, T_N)$$

Target distribution: $x^* \in \mathcal{X}^*$

$$\pi(\{x^*\}) = \mathbb{P}(X = x^* | X_1 = x, X_N = y)$$

Obtaining the marginal transition probability

$$\begin{aligned}\pi(\{x^*\}) &= \mathbb{P}(X = x^* | X_1 = x, X_N = y) \\ &= \frac{\mathbb{P}(X = x^* | X_1 = x)}{\mathbb{P}(X_N = y | X_1 = x)} \quad \left. \vphantom{\frac{\mathbb{P}(X = x^* | X_1 = x)}}{\mathbb{P}(X_N = y | X_1 = x)}} \right\} \begin{array}{l} \gamma(x^*) \\ Z \end{array}\end{aligned}$$

Marginal transition obtained from the estimator of Z

Proposal

Notation: $\tilde{\mathbb{P}}(X = x^*)$

Natural choice: Forward simulation

$$\tilde{\mathbb{P}}(X = x^*) = \mathbb{P}(X = x^* | X_1 = x)$$

The space is infinite \Rightarrow positive probability of not reaching y

Solution: introduce potentials ρ^y

- Functions on the state space $\rho^y : \mathcal{X} \rightarrow \mathbb{N}$
- Assume: $\rho^y(x) = 0$ iff $x = y$
- Dependency on the length to end point also possible

Example: Levenshtein edit distance

$$\begin{aligned} \rho^{\text{'ACTG'}}(\text{'CGG'}) &= \text{min number of point insertion, deletion, subst.} \\ &= 2 \end{aligned}$$

Using the potentials

- If for all $x \neq y$: $\mathbb{P} \left(\rho^y (X_{n+1}) < \rho^y (x) \mid X_n = x \right) > 0$
- For $\rho = \text{Levenshtein}$, this holds because for $x \neq y$ there is always a string z reached in one operation and closer (or equal) to y
- Then we can build $\tilde{\mathbb{P}}$ such that: $\tilde{\mathbb{P}}(N < \infty) = 1$

Construction

Notation: Proposal restricted on states decreasing the potential:

$$\nu_x^{\downarrow y}(A) = \nu_x(A \cap \{z : \rho^y(x) > \rho^y(z)\})$$

With α_x^y large enough, this yields a suitable $\tilde{\mathbb{P}}$:

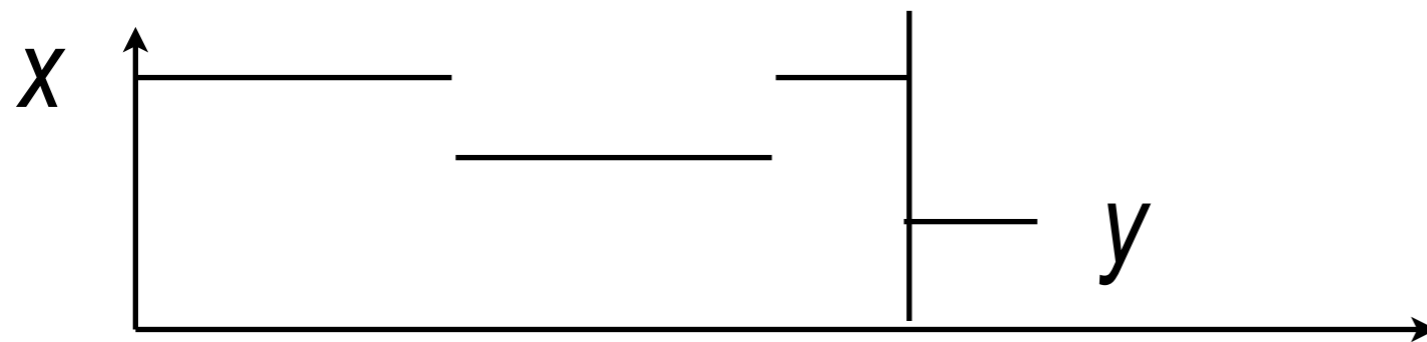
$$\tilde{\nu}_x = \alpha_x^y \frac{\nu_x^{\downarrow y}}{\nu_x^{\downarrow y}(\mathcal{X})} + (1 - \alpha_x^y) \frac{\nu_x - \nu_x^{\downarrow y}}{1 - \nu_x^{\downarrow y}(\mathcal{X})}$$

Example: for $\rho = \text{Levenshtein}$ can pick

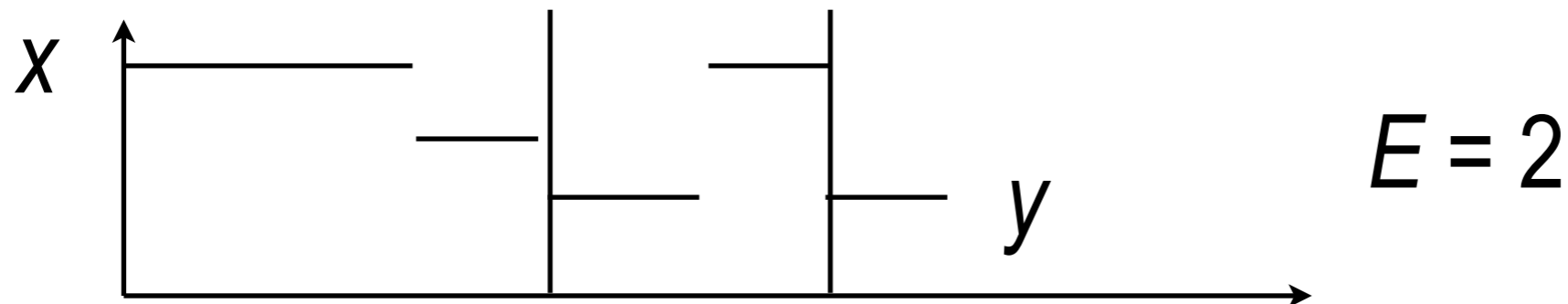
$$\alpha_x^y = \max\{\alpha, \nu_x^{\downarrow y}(\mathcal{X})\} \quad \alpha > \frac{1}{2}$$

Multiple excursions

Paths generated by $\tilde{\mathbb{P}}$ stop as soon as they hit y



This is not necessarily the case under \mathbb{P}



Solution: first sample a number of excursions E from a hyper-parameter distribution

$$E \sim \text{Geo}(\beta)$$

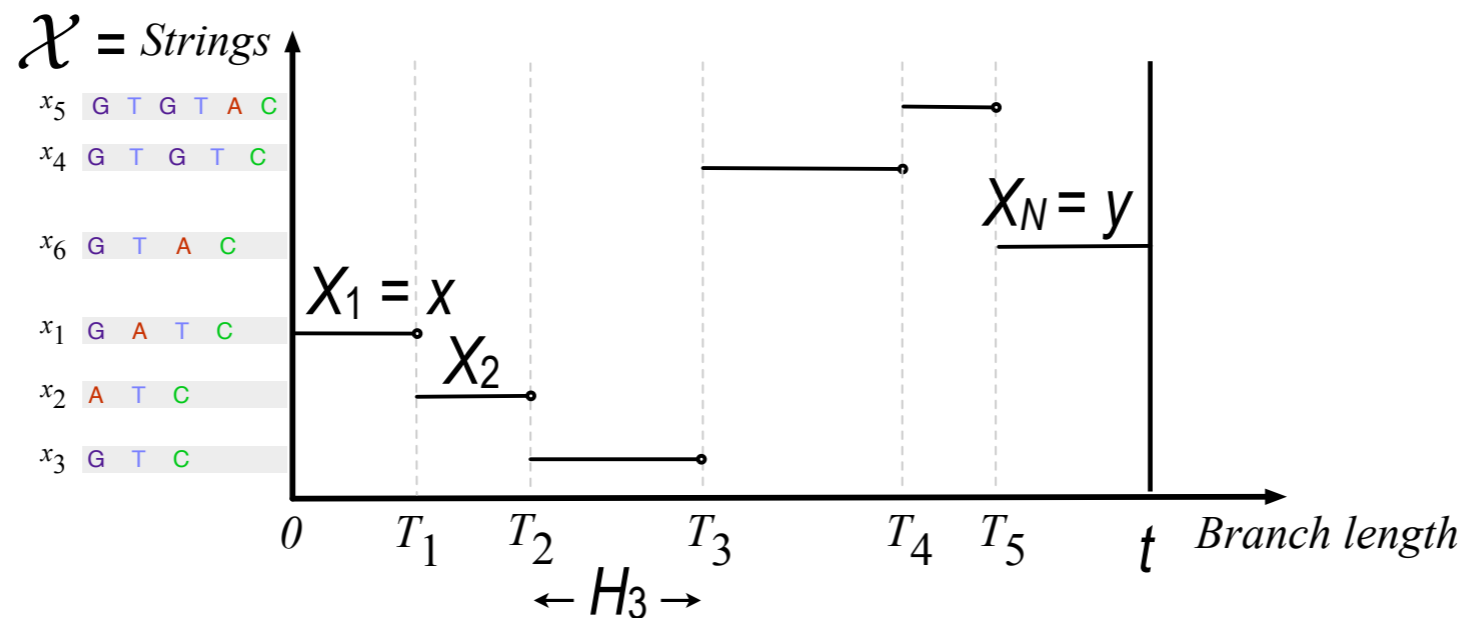
Proposal hyper-parameters

- How to set α, β ?
 - Optimal choice depends on the process and on t
- We use an ensemble of kernels with different combinations of α, β , ranging over several magnitudes
 - The particles produced by the members of this ensemble compete; the weights and resampling naturally do selection
 - Easy to justify with an auxiliary variable construction

Weights

Integrating the holding times:

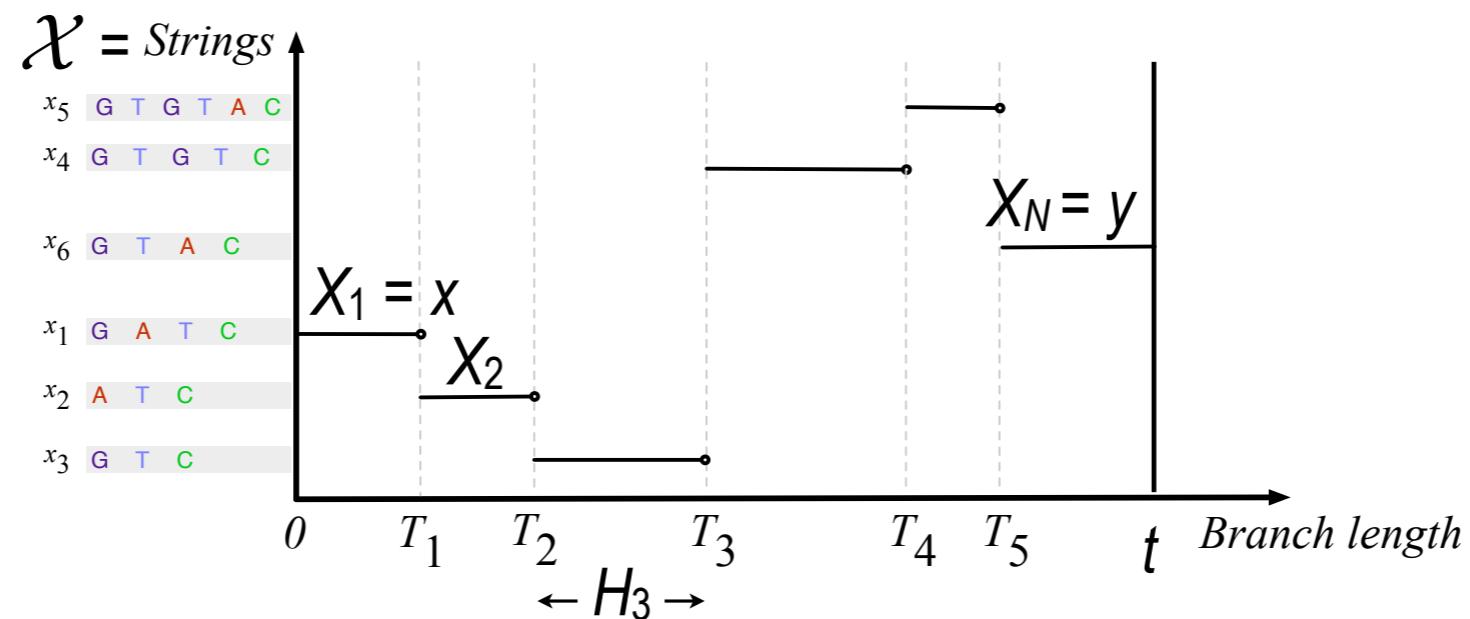
$$\underbrace{\int \dots \int}_{n=|x^*| \text{ times}} \lambda(x_1^*) \exp(-h_1 \lambda(x_1^*)) \dots \lambda(x_{n-1}^*) \exp(-h_{n-1} \lambda(x_{n-1}^*)) (1 - \exp(-h_n \lambda(x_n^*))) dh_1 \dots dh_n$$



- High dimensional integral
- Results on convolution of exponential?
 - Not directly applicable
 - Expensive when rates have multiplicities

Reduction to a matrix exponential

Idea: construct a finite rate matrix Q on the fly, for each particle

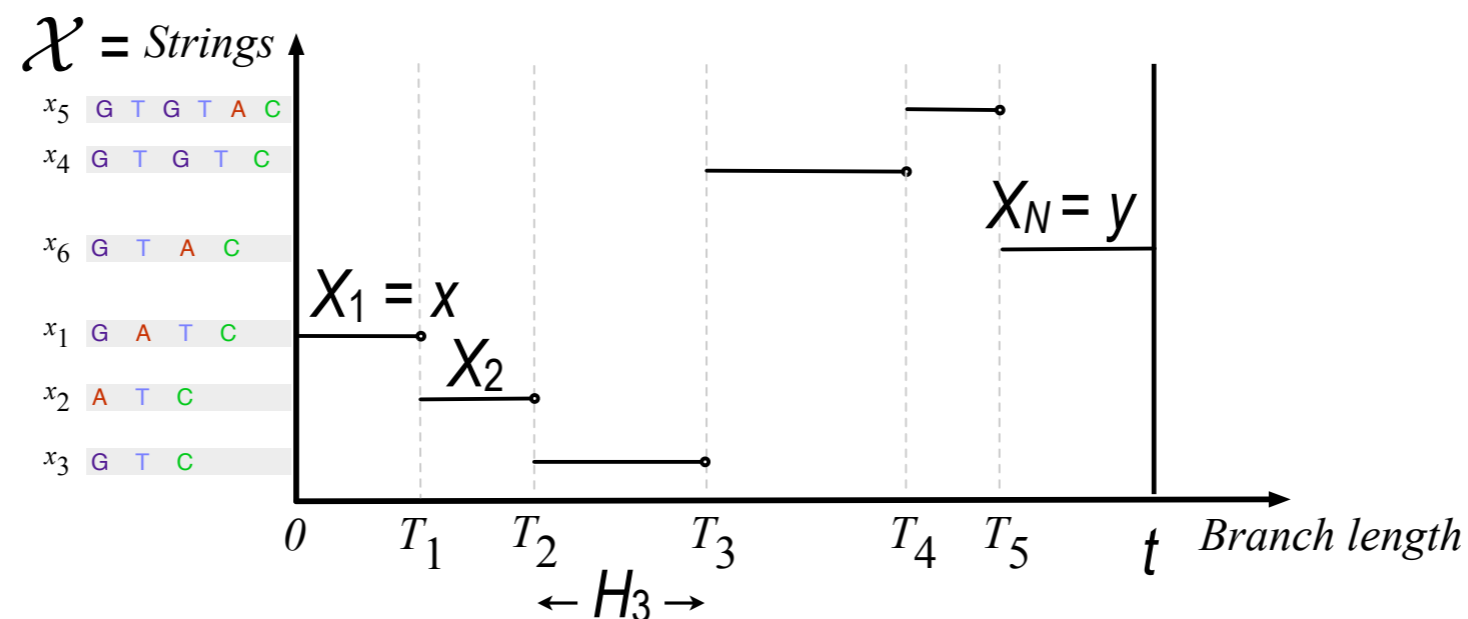


For each state visited in X , build an artificial state (with multiplicities)

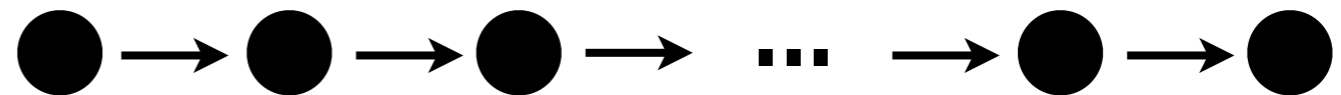


Reduction to a matrix exponential

Idea: construct a finite rate matrix Q on the fly, for each particle



For each state visited in X , build an artificial state (with multiplicities)



There will be positive rates only between consecutive artificial states

Numerical issues

- If all rates are distinct (in particular, same state not visited twice): exponentiation through diagonalisation is possible and fast
 - Using sparsity: inversion is quadratic
 - Can do the computation only for one entry of M
- If rates are not distinct: above method fails (Q does not have a complete set of linearly indep. eigenvectors)
 - Can use Jordan-Chevalley decomposition ($Q = A + N$, A diag., N nilpotent)
 - Simpler: Padé series + scaling & squaring method

Experiments

- Numerical validations of consistency in # of particles
 - All the ideas presented today tested on 2 (of the rare) countably infinite CTMCs with closed form for the marginals
 - Linear birth death process
 - Poisson Indel Process
- Experiments on phylogenetic inference for the proposal presented today but without integrated holding times

Experiments

- Task: reconstruction of tree topologies and branch lengths (error measured using tree metrics)
- 10 taxa at the leaves
- Example of simulated data:
- Alignment not given

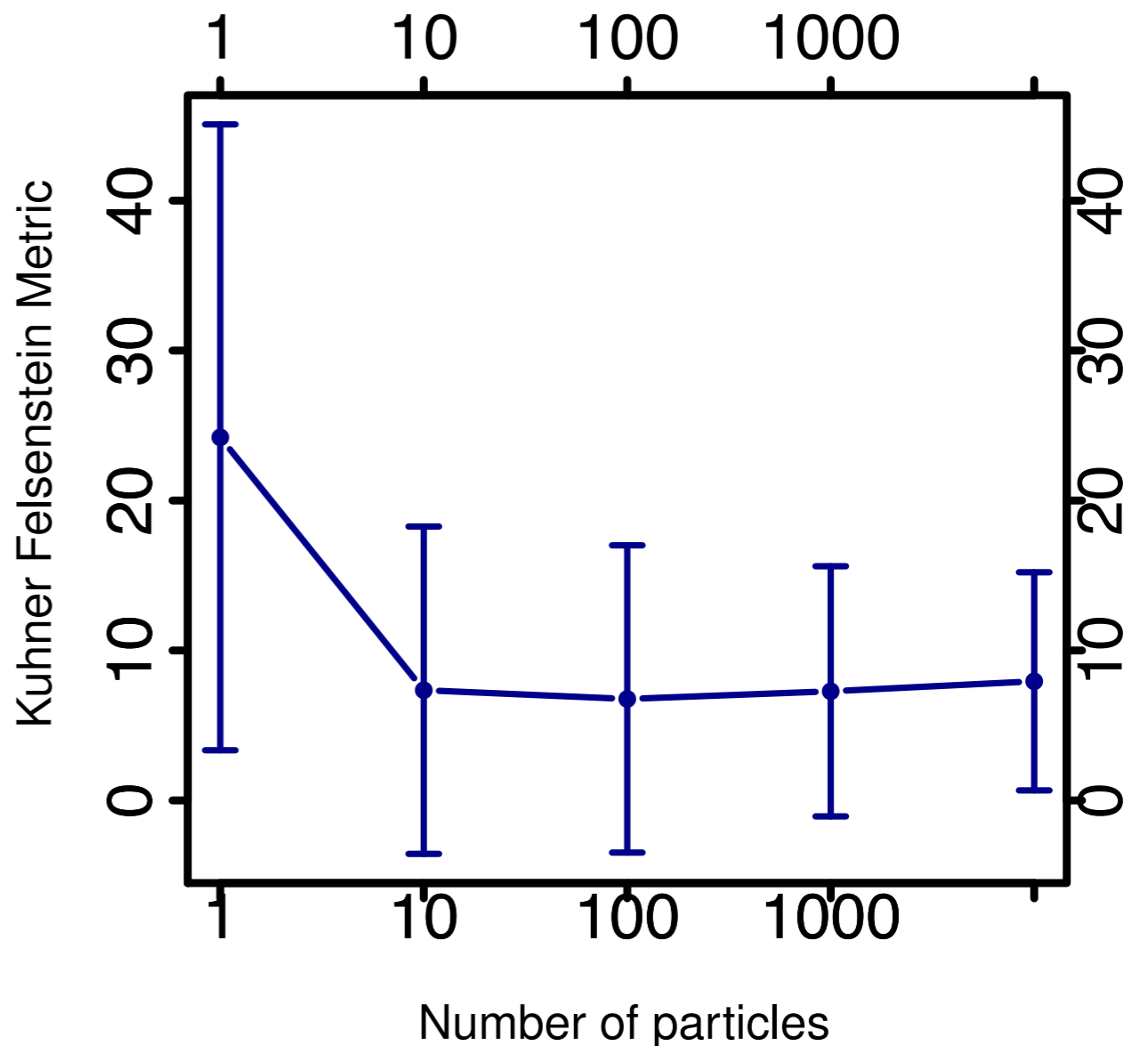
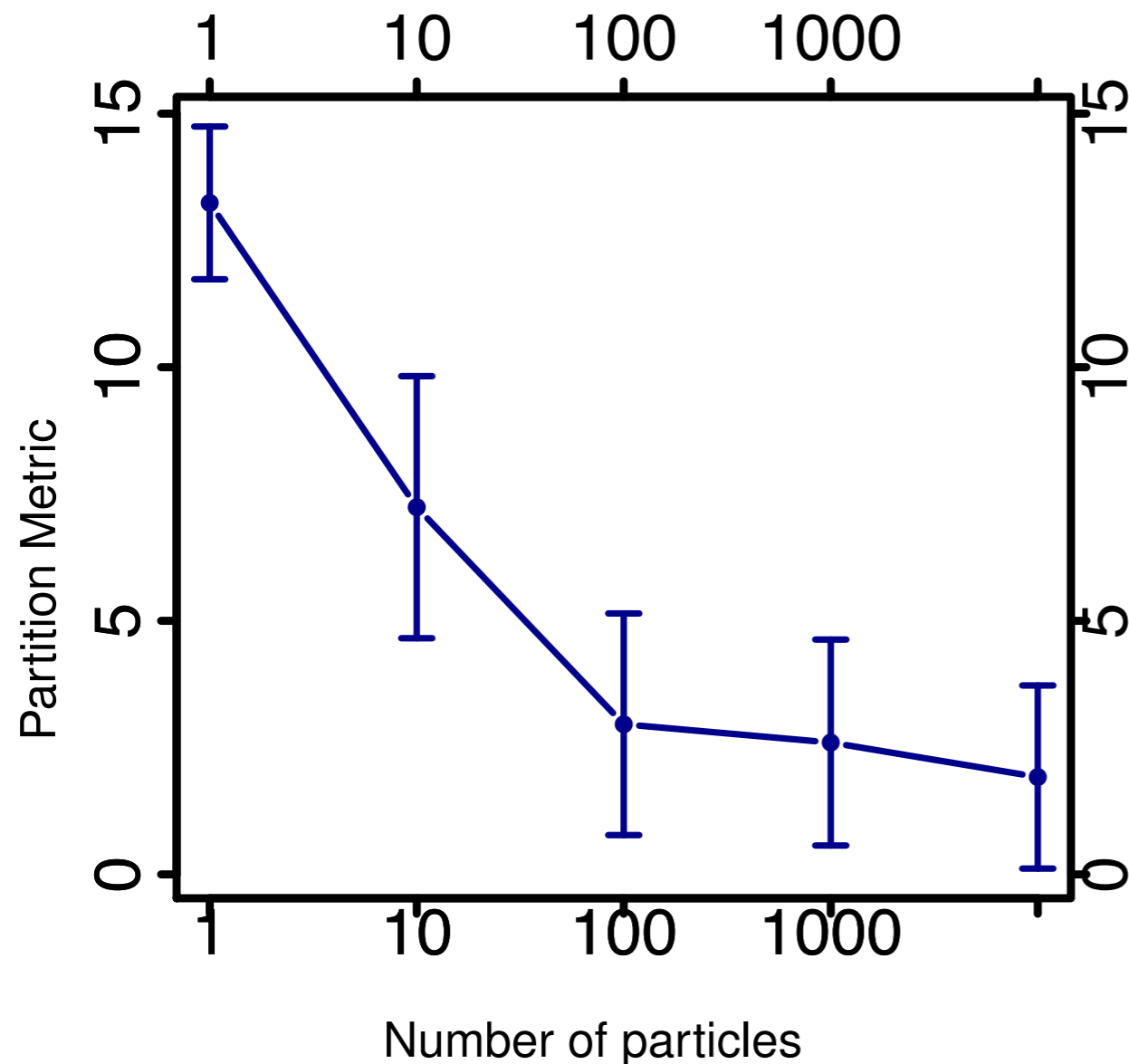
```

internal_0|CA--G---C---A--G-----TG--A---
internal_1|-GAG-C---G-G-----AA---GA---TGC-TGC
internal_2|--AG-CAG--CC-----CG--C-GAC---TG-----
internal_3|-GAG-C---G-G-----AA---GA---TGC-----
internal_4|-GAG-C---G-G-----AA---GA---TGC-----
internal_5|-GAG-C---G-G-----AA---GA---TGC-----
internal_6|-GAG-C---G-G-----AA---GA---TGC-----
internal_7|-TAG-C---G-C-----CA--C-GAC---TGC-----
internal_8|ATAG-C---G-----C---A---G-C-GGCA---
leaf_0    |CA--G---C---A--G---C--A---G-TG--A---
leaf_1    |-GAG-C---G-G-----AA---GT---TGC-TGC
leaf_2    |-GAG-C---G-G-----AA---GA---TGC-TGC
leaf_3    |-GAG-C---G-G-----AA---GA---TGC-----
leaf_4    |CA--G---C---A--G-----TG--A---
leaf_5    |-GAT-C---G-G-----AA---GA---TGC-----
leaf_6    |--AG-CAG--CC-----CG--C-GAC---TG-----
leaf_7    |-GAG-C---G-G-----AA---GA---TGC-----
leaf_8    |--AG-CAG--CC-GC--CCG--C-GAC---CG-----
leaf_9    |-GAT-----G-G-----GA---GT-----GC-----
    
```

Setting: SSM length is 3; $\theta_{sub} = 0.03$; $\lambda_{pt} = 0.05$; $\mu_{pt} = 0.2$; $\lambda_{SSM} = 2.0$; $\mu_{SSM} = 2.0$

Preliminary results

Tree inference using correct parameters:



(replications on 10 random trees & datasets)

Scaling up to large datasets

- Large number of particles needed
 - Large phylogenetic trees
 - Mixing proposals with different hyper-parameter values α , β
- Motivation for parallel architectures
 - Revised Moore's law: parallel architectures
 - Each particle is large
 - particles are forests
 - need to keep one string for each tree in forest
 - 'worst' case: one string = one genome

Part II : Entangled Monte Carlo (EMC)

- Goal:
 - Do parallelization in such a way that the result is equivalent to running everything on a (hypothetical) single machine
- Complementary approach: modify SMC
 - Éric Moulines' talk on Island models from yesterday
 - Pierre Jacob's talk on pairwise resampling scheme; this afternoon

Stochastic maps

- A way to decouple randomness and state dependencies
 - Consider an arbitrary kernel: $T : \mathcal{S} \times \mathcal{F}_{\mathcal{S}} \rightarrow [0, 1]$
 - Stochastic map: $(\mathcal{S} \rightarrow \mathcal{S})$ -valued r.v. F such that

$$T(s, A) = \mathbb{P}(F(s) \in A)$$

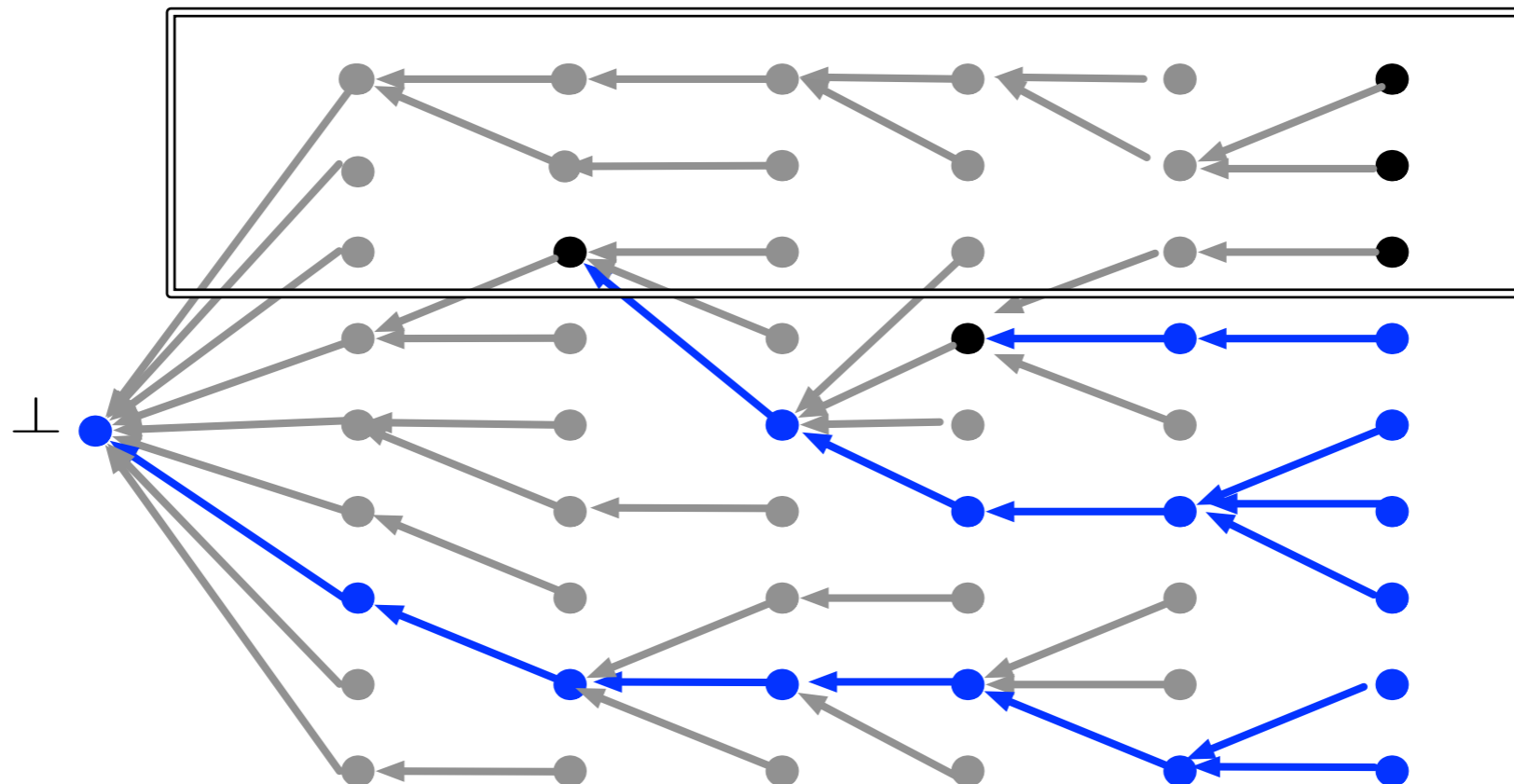
- Example: alternate view on MCMC
 - Sample F_1, F_2, \dots **i.i.d.**
 - Pick x_0 arbitrarily
 - Return:

$$F_1 \circ \dots \circ F_n(x_0)$$

Overview

Sample a global collection of i.i.d. stochastic maps for both the proposal $\{ F_i \}$ and resampling steps $\{ G_i \}$

Assume the global collection is transmitted to all machines ($O(1)$ if pseudo-random)

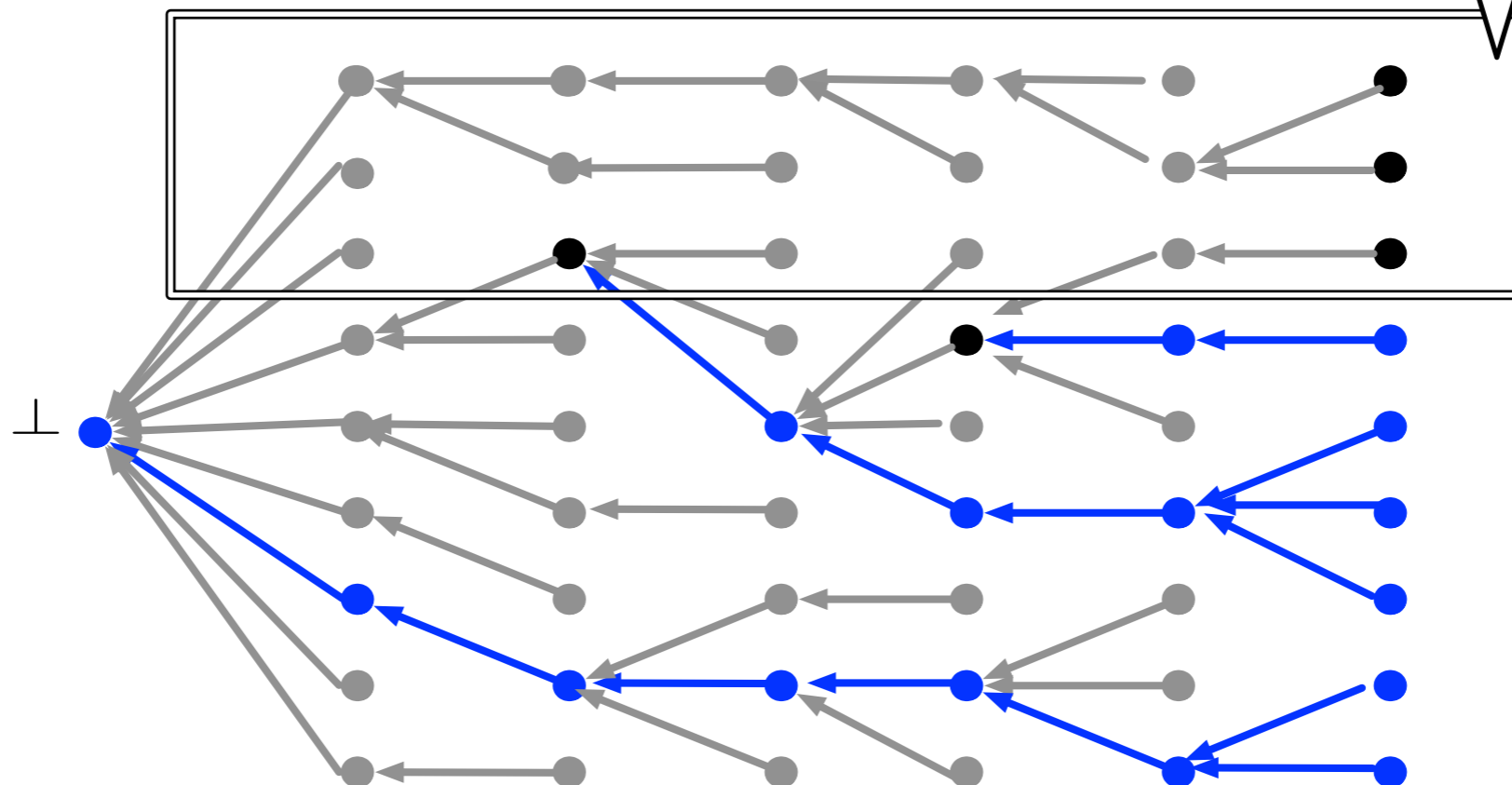


Overview

Sample a global collection of i.i.d. stochastic maps for both the proposal $\{ F_i \}$ and resampling steps $\{ G_i \}$

Assume the global collection is transmitted to all ($O(1)$ if pseudo-random)

Each machine m is responsible of a subset of the particles

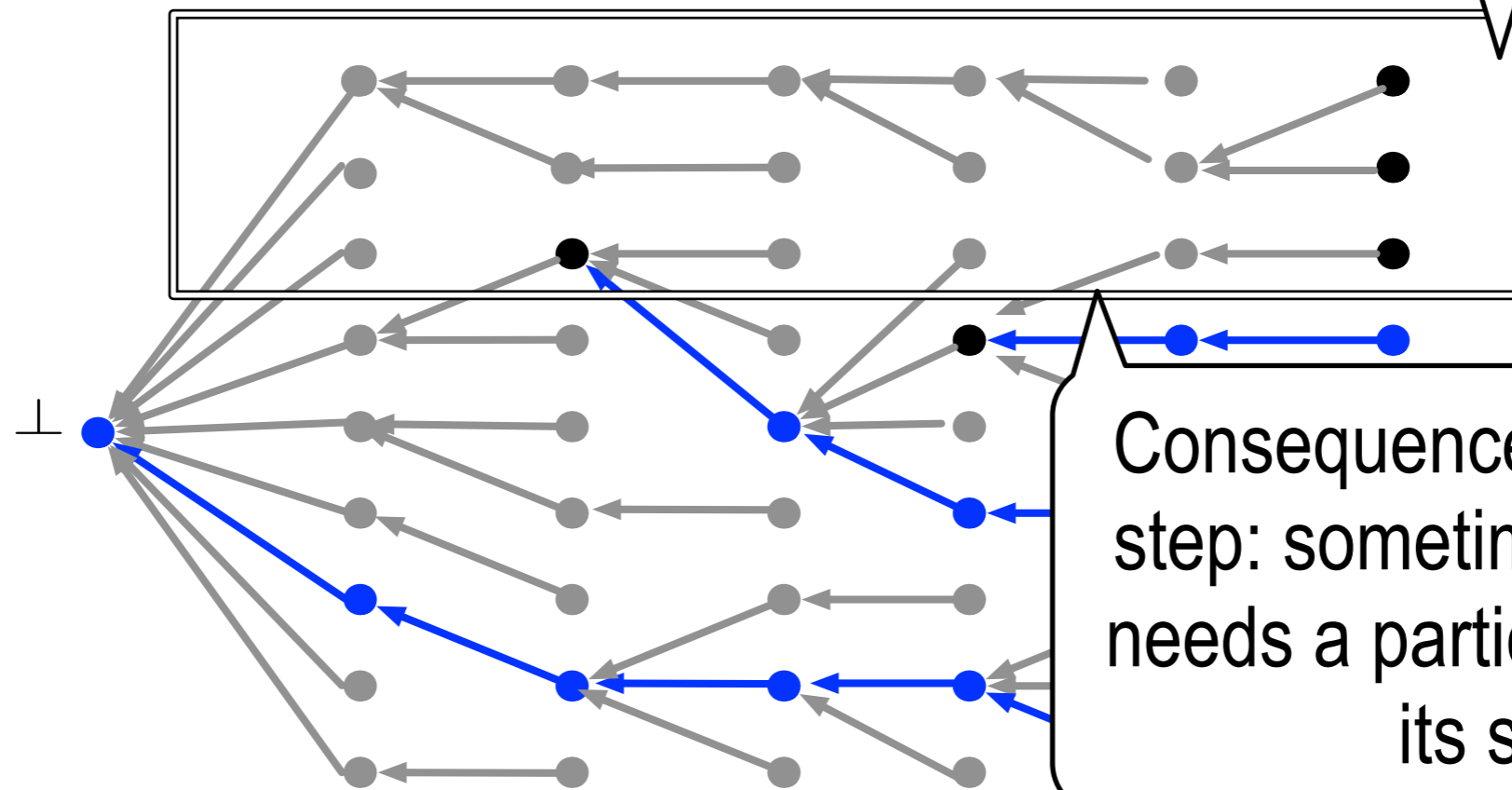


Overview

Sample a global collection of i.i.d. stochastic maps for both the proposal $\{ F_i \}$ and resampling steps $\{ G_i \}$

Assume the global collection is transmitted to all ($O(1)$ if pseudo-random)

Each machine m is responsible of a subset of the particles



Consequence of resampling step: sometimes machine m needs a particles i outside of its subset

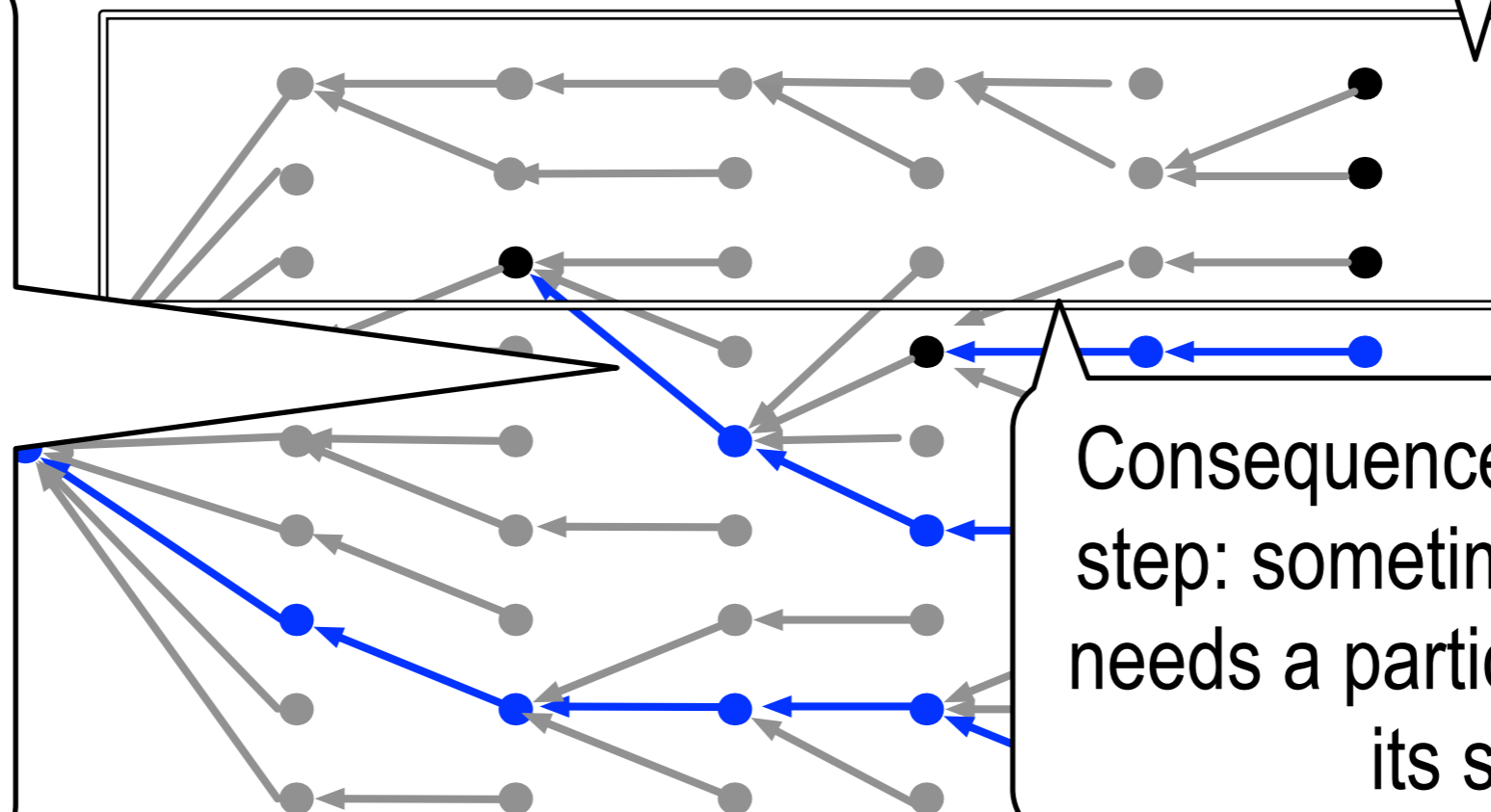
Overview

Sample a global collection of i.i.d. stochastic maps for both the proposal $\{ F_i \}$ and resampling steps $\{ G_i \}$

Assume the global collection is transmitted to all ($O(1)$ if pseudo-random)

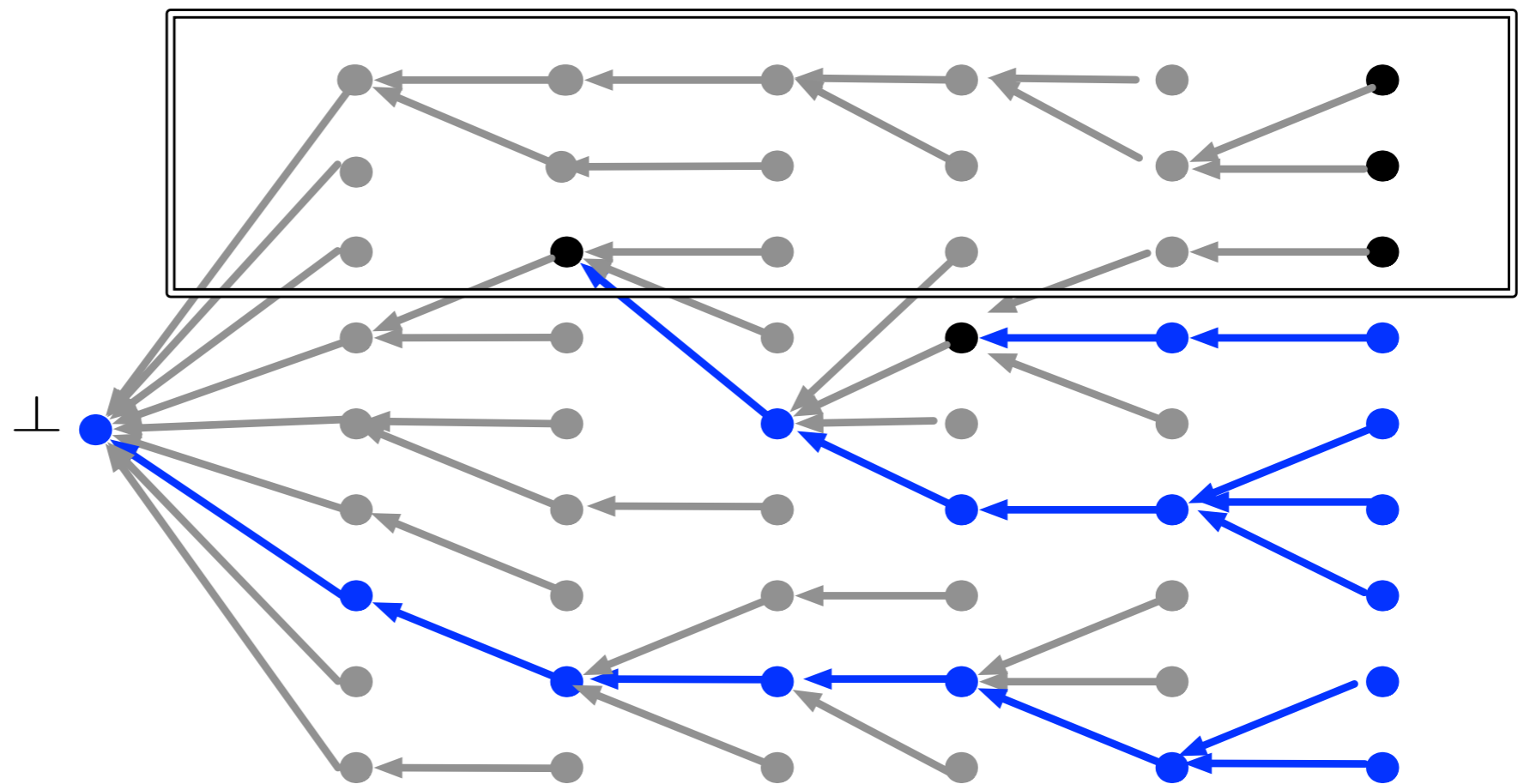
Each machine m is responsible of a subset of the particles

Idea:
Reconstruct particle i using the stochastic maps



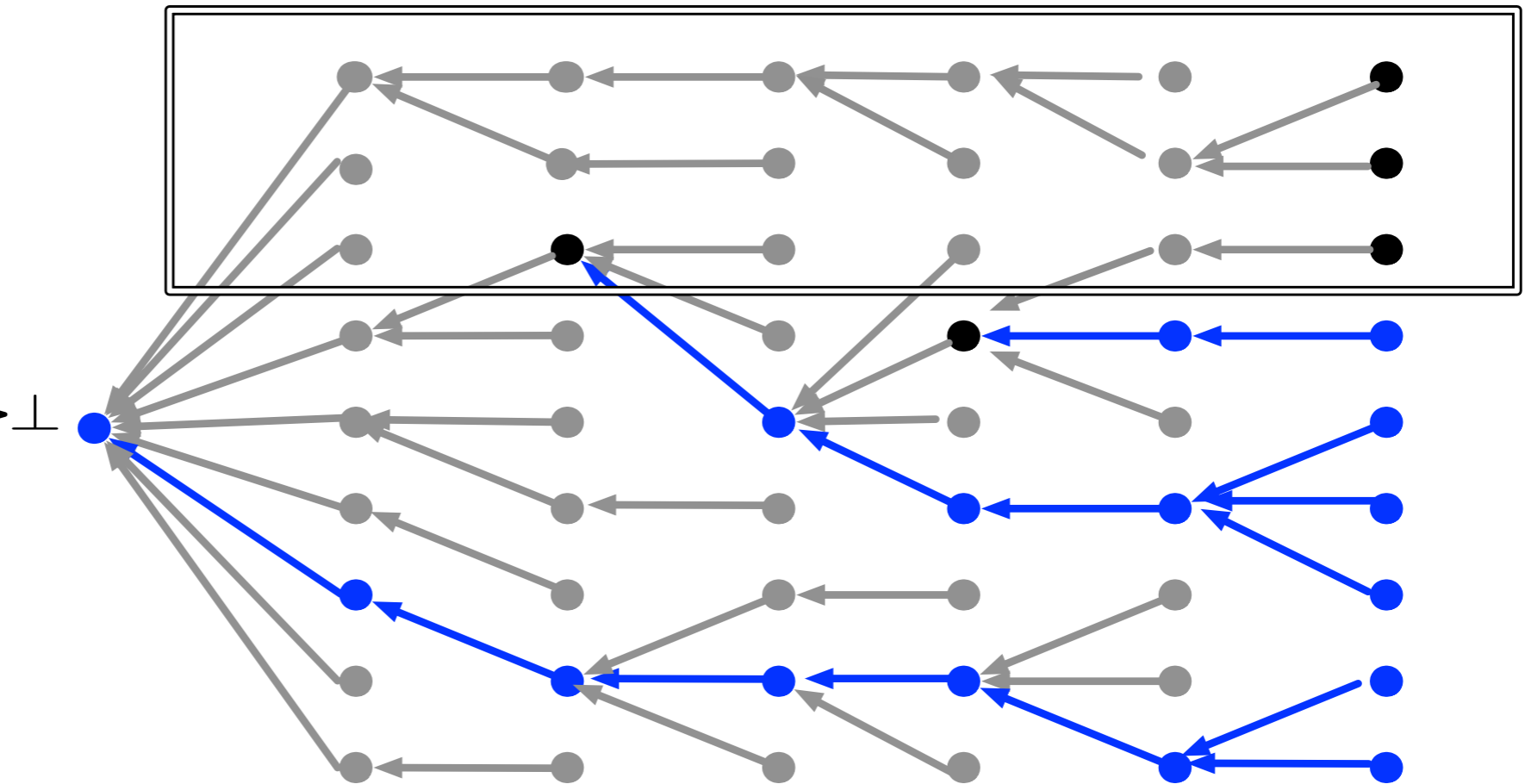
Consequence of resampling step: sometimes machine m needs a particles i outside of its subset

Distributed genealogy $s(i)$, $\rho(i)$



Distributed genealogy $s(i)$, $\rho(i)$

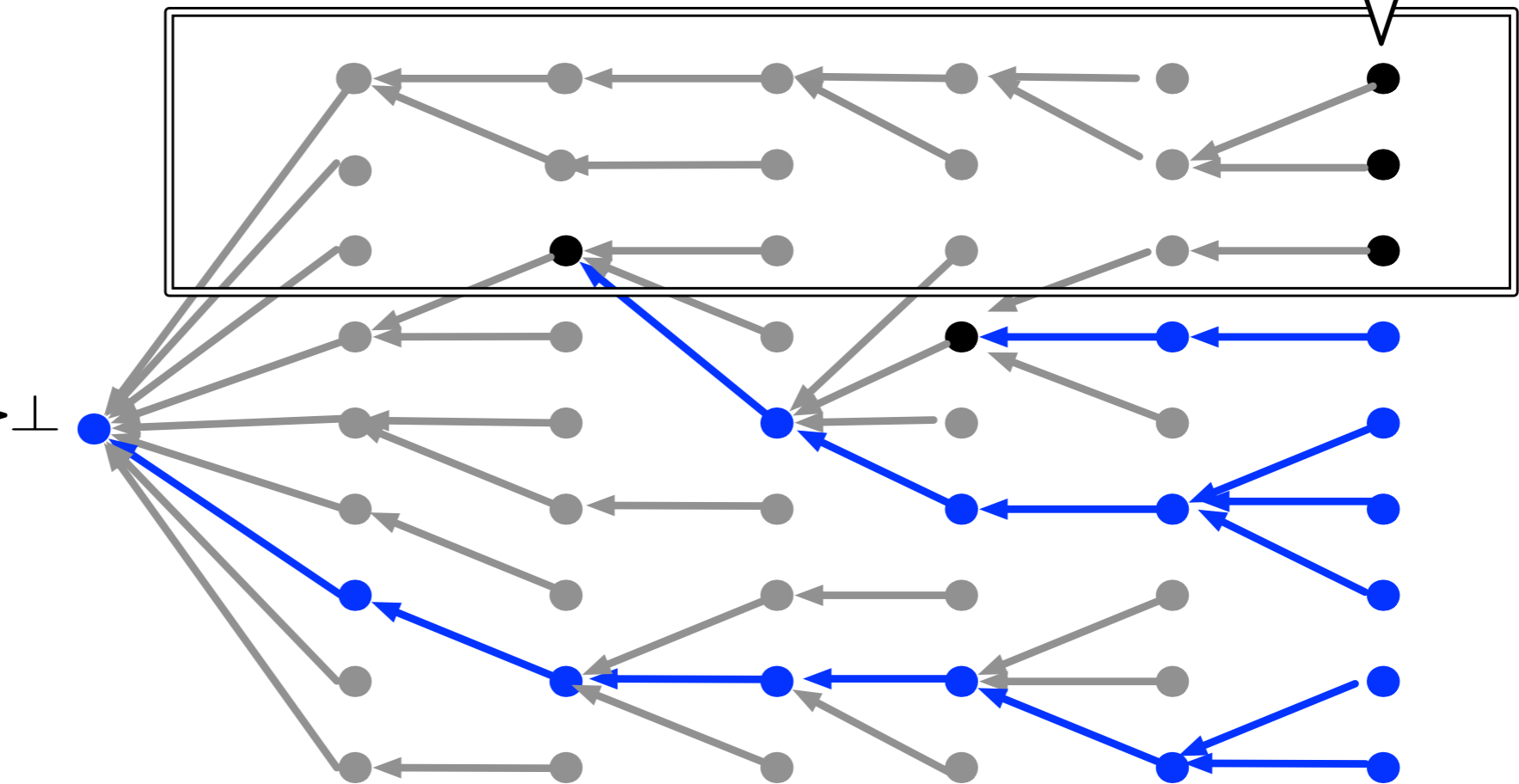
Assume
w.l.o.g. there
is always a
shared
common
ancestor



Distributed genealogy $s(i)$, $\rho(i)$

Current **concrete** particles: those explicitly stored in machine m
 $s(i) \neq \text{nil}$

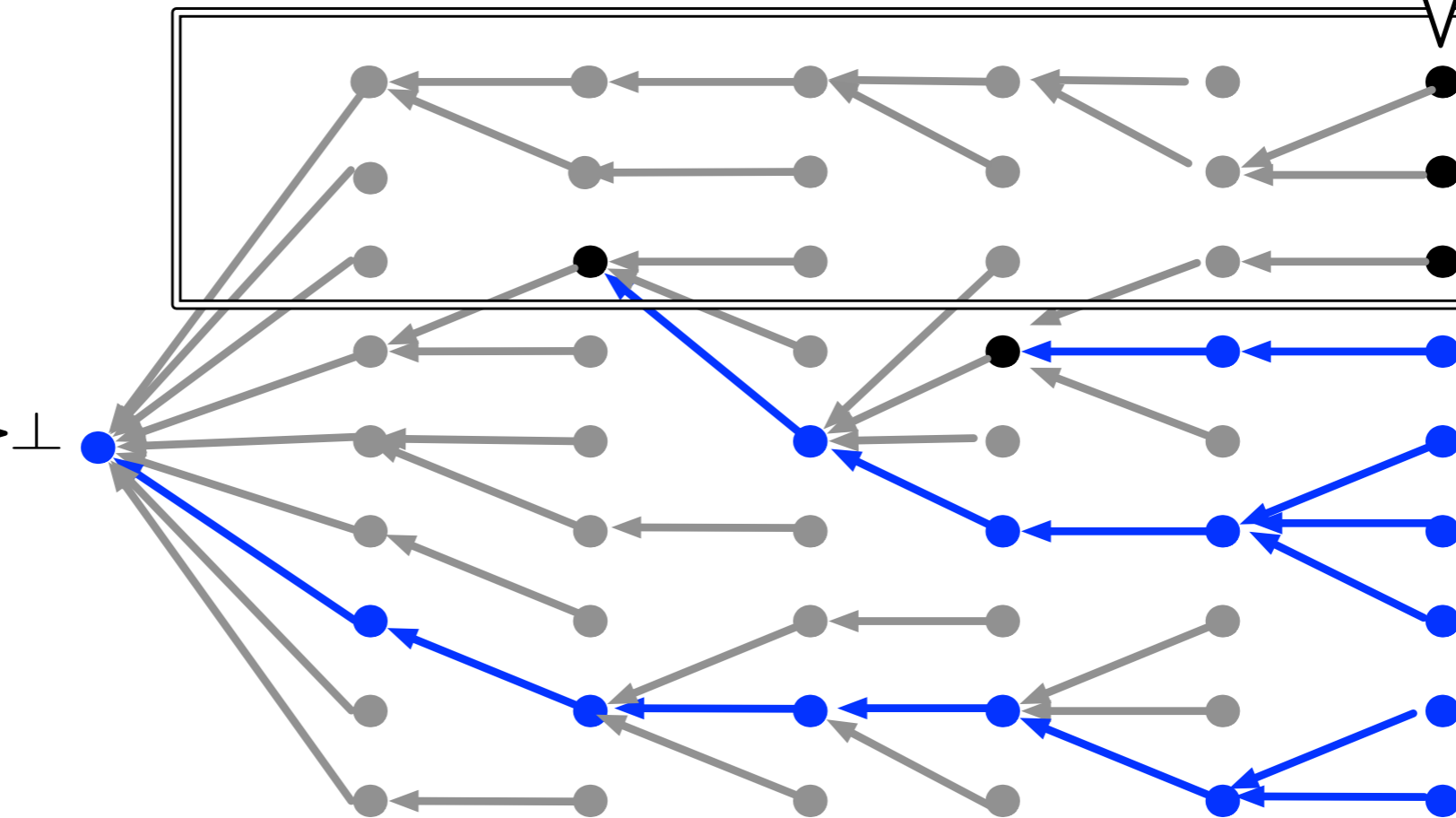
Assume
w.l.o.g. there
is always a
shared
common
ancestor



Distributed genealogy $s(i)$, $\rho(i)$

Current **concrete** particles: those explicitly stored in machine m
 $s(i) \neq \text{nil}$

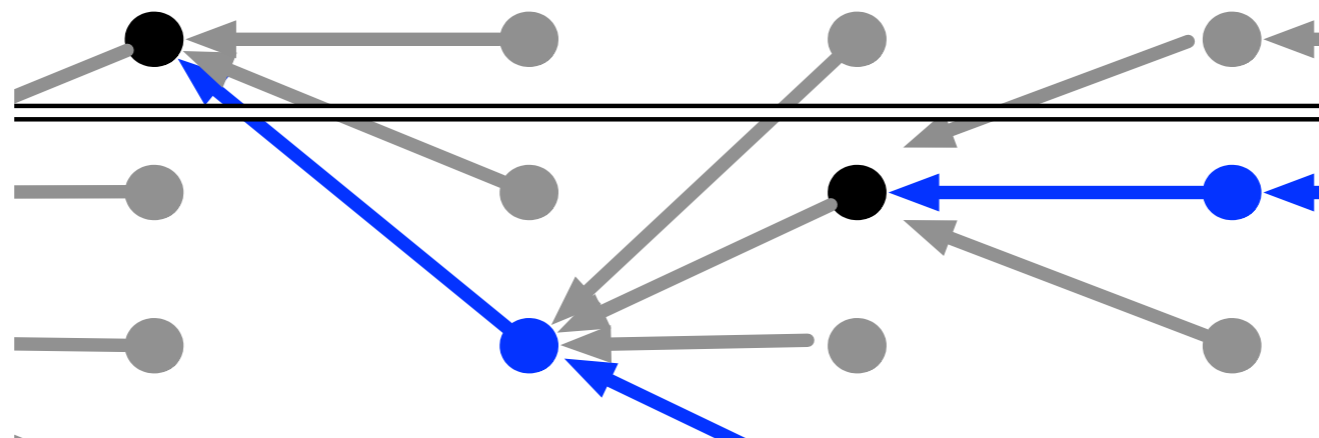
Assume
w.l.o.g. there
is always a
shared
common
ancestor



Current **compact** particles: only store id of the parent $\rho(i)$
 $s(i) = \text{nil}$

Reconstruction of particle i

- 1: $F \leftarrow I$
- 2: **while** ($s(i) = \text{nil}$) **do**
- 3: $F \leftarrow F \circ F_i$
- 4: $i \leftarrow \rho(i)$
- 5: **end while**
- 6: **return** $F(s(i))$



Resampling

- At resampling, only transmit particle weights
- Genealogy can be updated efficiently from this information

Details

- See NIPS paper:
 - Jun, Wang, Bouchard-Côté (2012) NIPS.
- Datastructures the stochastic maps
 - Constant storage using pseudo-randomness
 - Need random access to the random number: binary trees of xor'ing 2 streams of random numbers
- Allocations schemes: heuristics to minimize the amount of particle transmission

Experiments

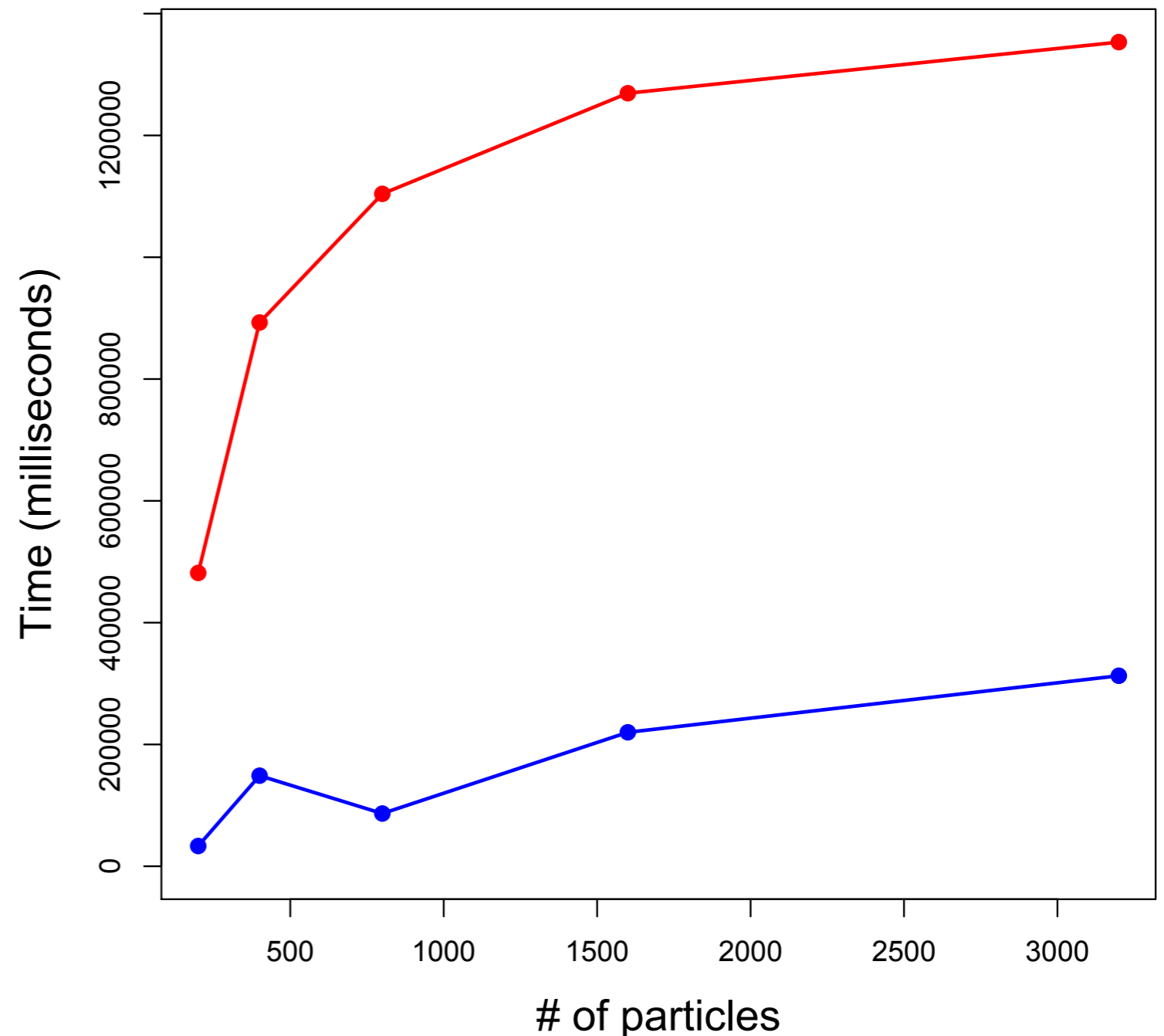
■ Setup:

- Phylogenetic inference
- 100 particles/EC2 instance

■ Comparison:

- Particle transmission over network (red)
- EMC (blue)

Total run time of EMC versus Particle transfer

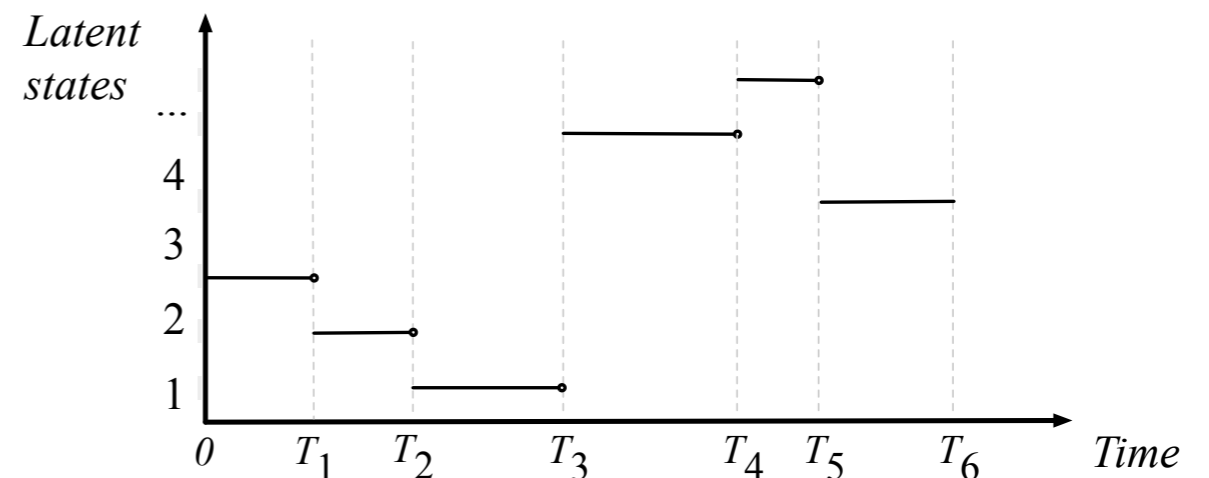
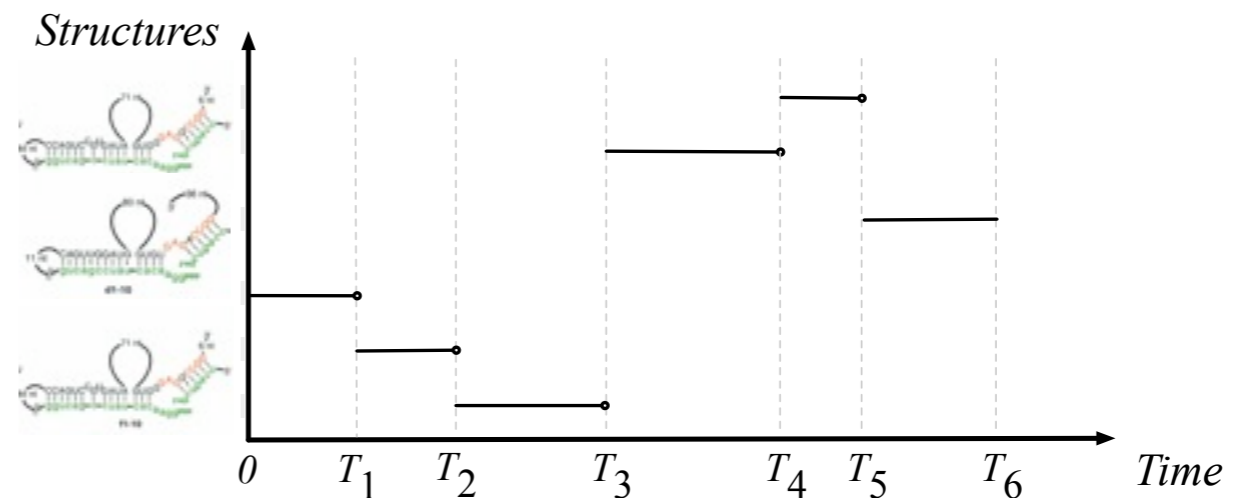


Future directions

- SMC algorithms for inference over countably infinite / combinatorial CTMCs
 - Using these techniques to remove the bounded jump rate assumption in jump-diffusion methods
 - New applications:

RNA strand following its folding pathway

Bayesian non-parametric model



Future directions

- EMC
 - Working on another version where only the sum of the particle weights is transmitted (using DHT methods)
 - Better understanding of when and why the method works well