Bayesian Phylogenetic Inference using Sequential Monte Carlo Algorithms

Alexandre Bouchard-Côté*, Sriram Sankararaman*, and Michael I. Jordan*,†

* Computer Science Division, University of California Berkeley

† Department of Statistics, University of California Berkeley

Phylogenetic tree inference

Topic of this talk: **integration** over the space of trees using Sequential Monte Carlo (SMC)

Motivation: Bayesian approach to phylogenetic inference

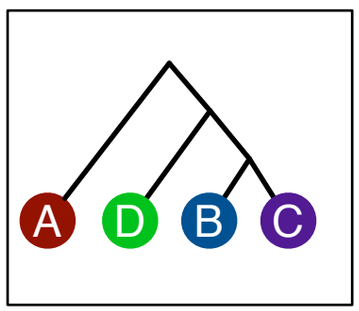
- Put a prior on trees, use the posterior for reconstruction
- Heavy use of integrals over the space of trees: e.g. for handling nuisance parameters, computing minimum risk estimators, Bayes factors, etc.

Prelude: a parallel with the simpler problem of **maximization** over the space of trees

Maximization over phylogenies

Two strategies: *Local* and *sequential* search

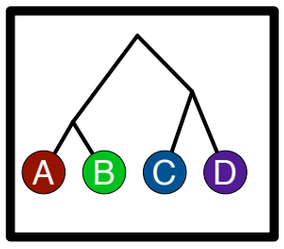
Key difference: representation

Local	Sequential
 **State** t	
Trees over the observed species	

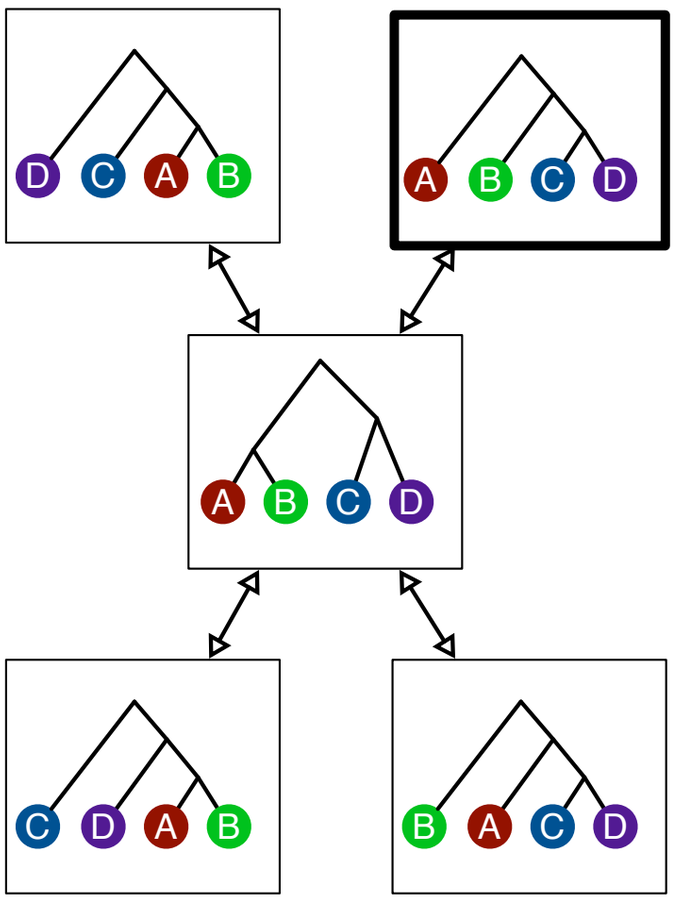
Maximization: *local* strategy

Meta-algorithm:

1. Start at arbitrary *state*



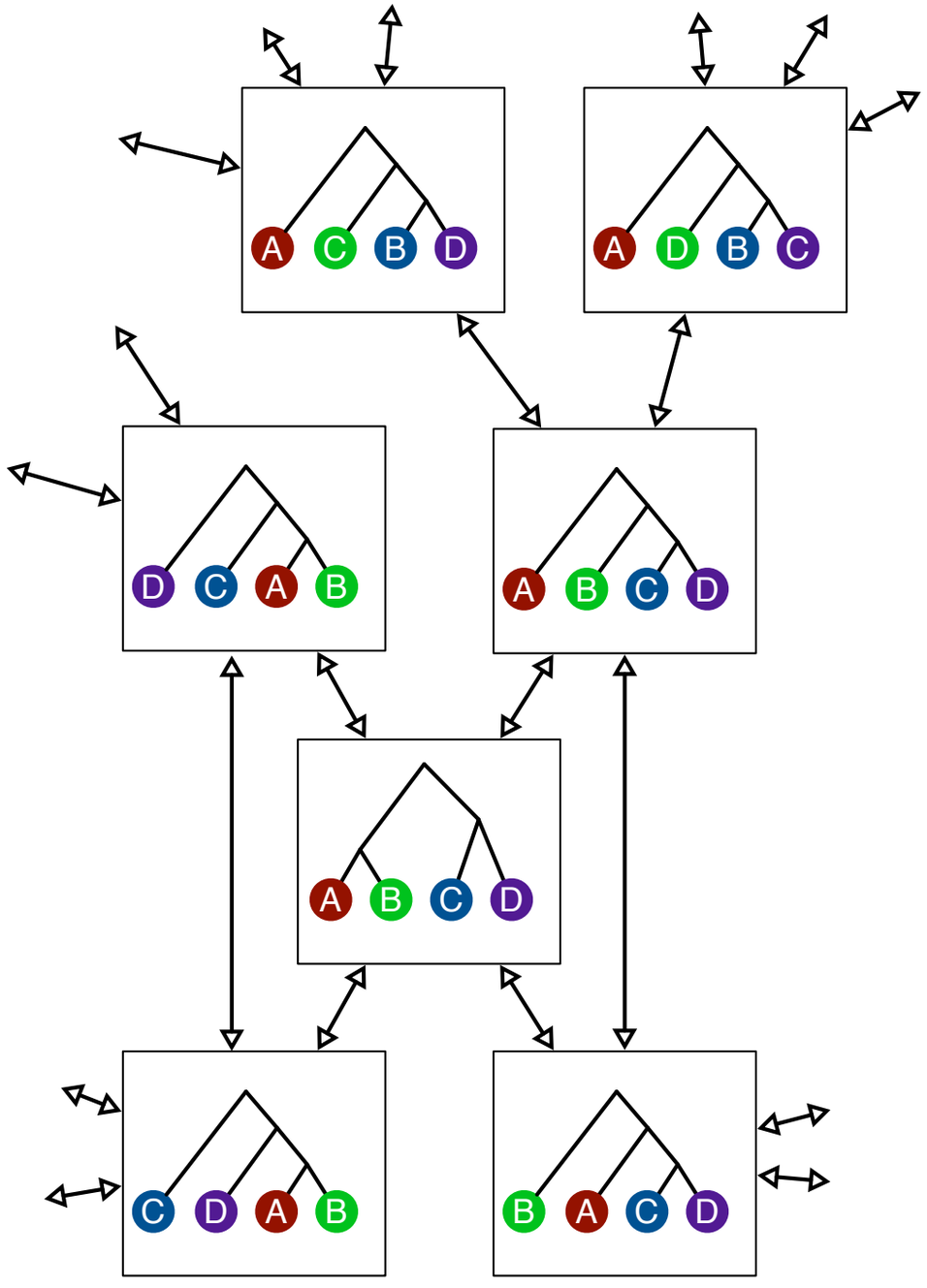
Maximization: *local* strategy



Meta-algorithm:

1. Start at arbitrary *state*
2. Iterate:
   i. Evaluate neighbors
   ii. Move to a nearby tree

Maximization: *local* strategy



Meta-algorithm:

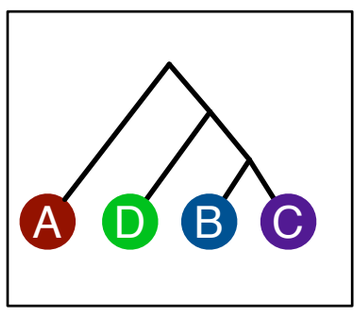
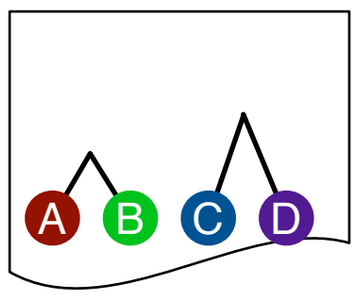
1. Start at arbitrary *state*
2. Iterate:
   i. Evaluate neighbors
   ii. Move to a nearby tree
3. Return best state visited

Example: stochastic annealing

Maximization over phylogenies

Two strategies: *Local* and *sequential* search

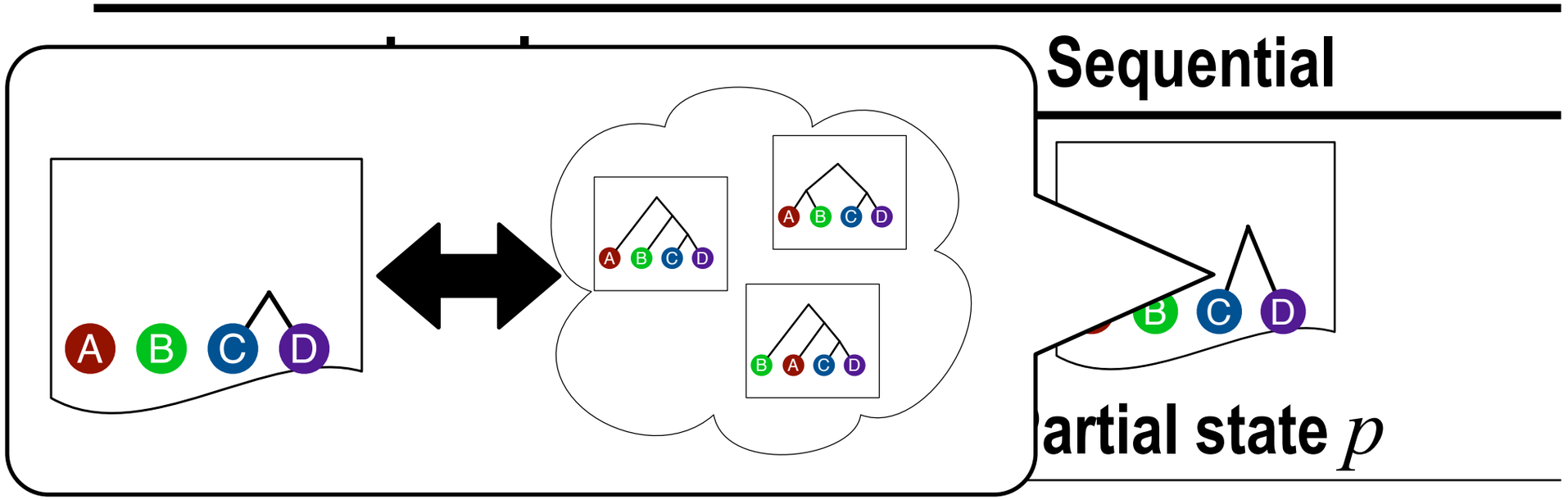
Key difference: representation

Local	Sequential
	
State t	**Partial state p**
Trees over the observed species	Forests over the observed species

Maximization over phylogenies

Two strategies: *Local* and *sequential* search

Key difference: representation



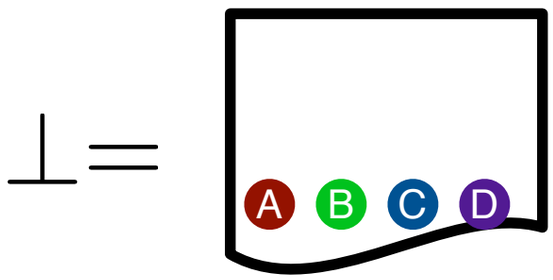
Trees over the observed species

Forests over the observed species

Maximization: *sequential* strategy

Meta-algorithm:

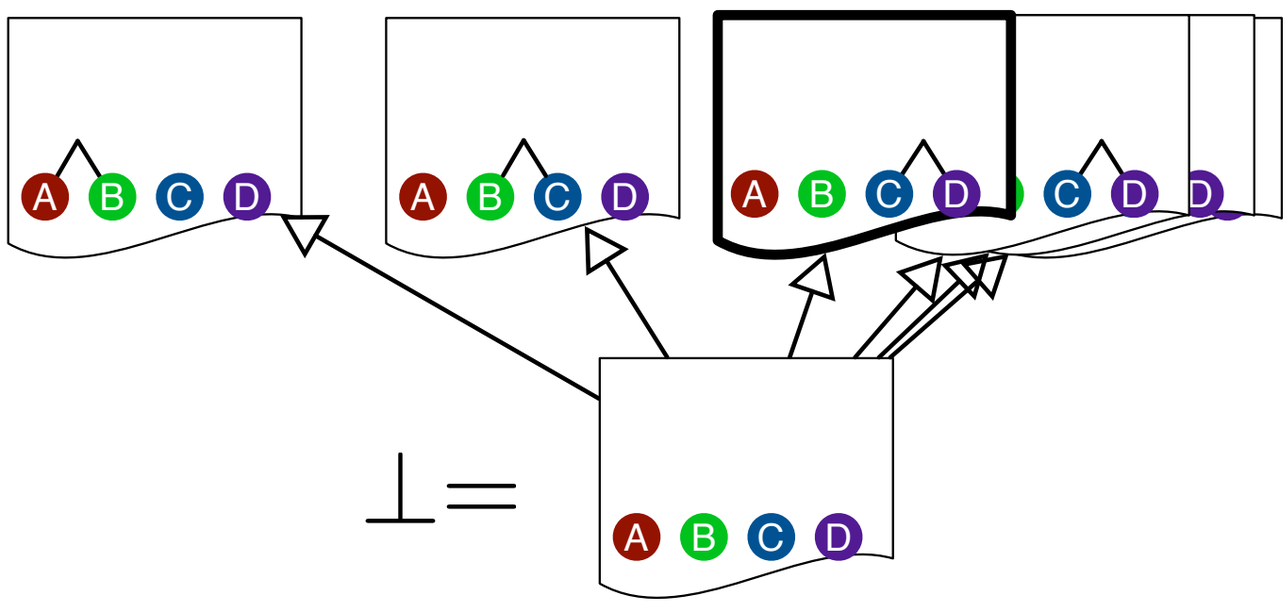
1. Start at the initial, unconstrained *partial state*



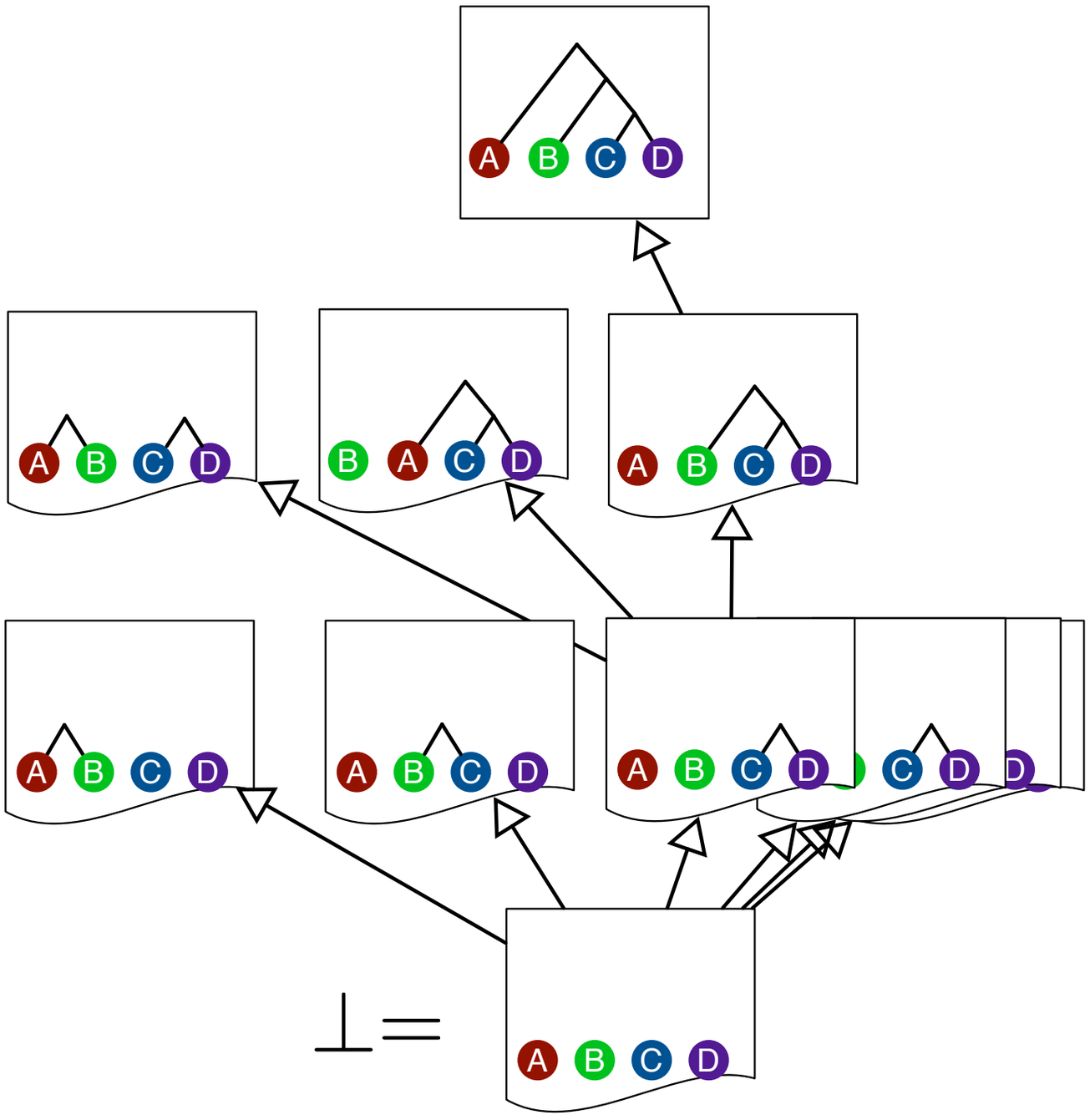
Maximization: *sequential* strategy

Meta-algorithm:

1. Start at the initial, unconstrained *partial state*
2. Iterate:
   i. Extend partial state
   ii. Estimate best successor



Maximization: *sequential* strategy



Meta-algorithm:

1. Start at the initial, unconstrained *partial state*
2. Iterate:
   i. Extend partial state
   ii. Estimate best successor
3. Return best final state

Example: neighbor joining

Parallel

Classification of phylogenetic algorithms

	Local strategy	Sequential strategy
Maximization	Stochastic annealing, ...	Neighbor-joining, ...
Integration		

Parallel

Classification of phylogenetic algorithms

	Local strategy	Sequential strategy
Maximization	Stochastic annealing, ...	Neighbor-joining, ...
Integration	MCMC algorithms	???

Parallel

Classification of phylogenetic algorithms

	Local strategy	Sequential strategy
Maximization	Stochastic annealing, ...	Neighbor-joining, ...
Integration	MCMC algorithms	???

Sequential Monte Carlo
(SMC)

Outline

Background: Importance sampling and Sequential Monte Carlo

SMC for phylogenetic inference

Framework for designing proposals

Experiments: comparisons with MCMC

Preview: Comparative advantages

SMC

+ Trivial to parallelize
+ Easier to get data likelihood estimate
+ No burn-in

MCMC

+ Easier to resample hyper-parameters
+ Easier to design proposal distribution

Preview: Comparative advantages

SMC

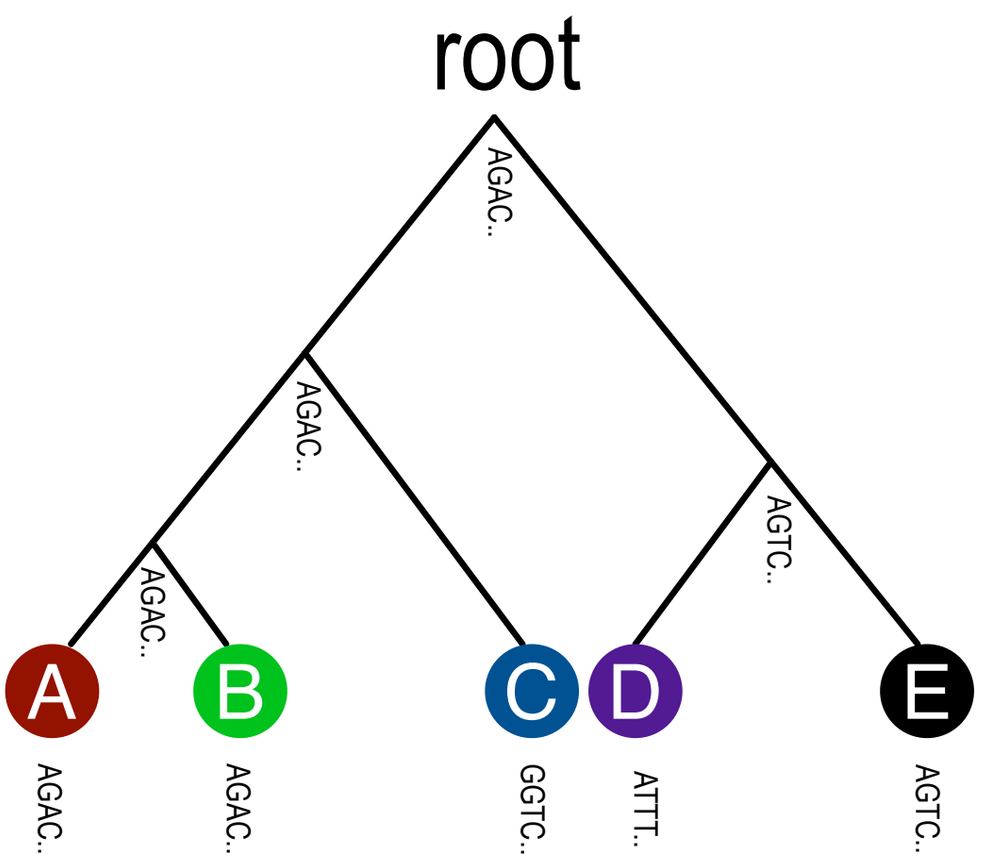
+ Trivial to parallelize
+ Easier to get data likelihood estimate
+ No burn-in

MCMC

+ Easier to resample hyper-parameters
+ Easier to design proposal distribution

Not exclusive: the two approaches can be combined

Phylogenetic setup: ultrametric trees

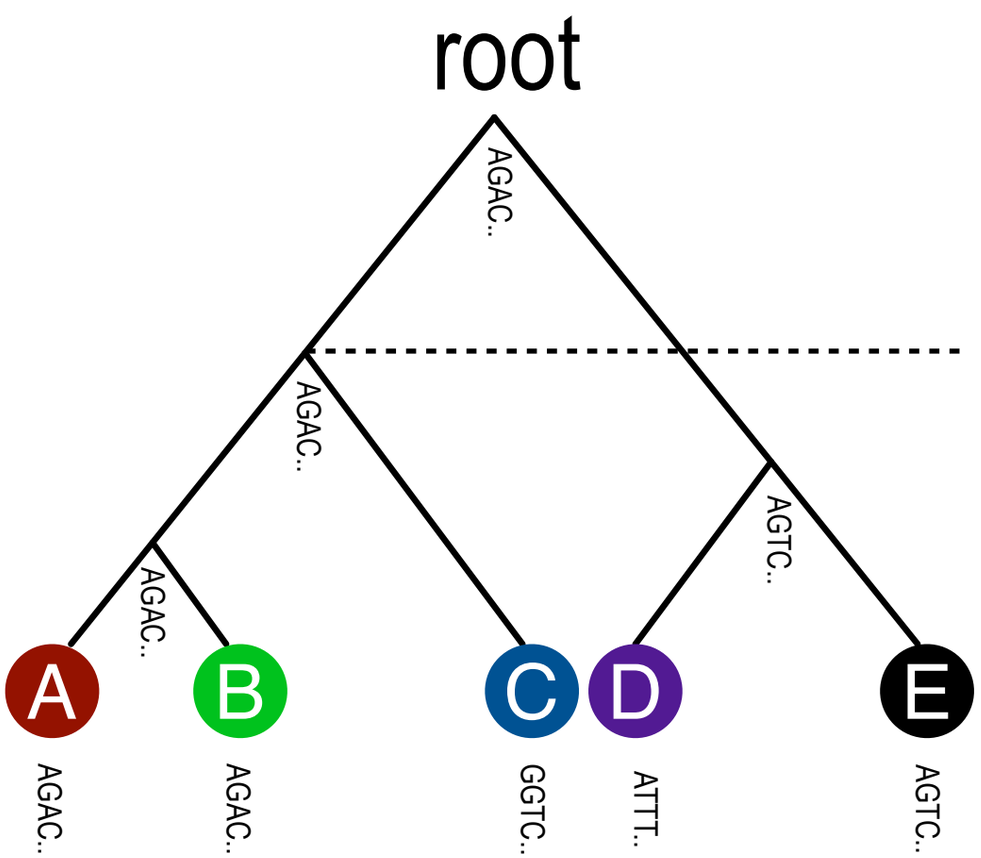


Tree T

Hidden sequences X

Observations $Y = y$

Phylogenetic setup: ultrametric trees



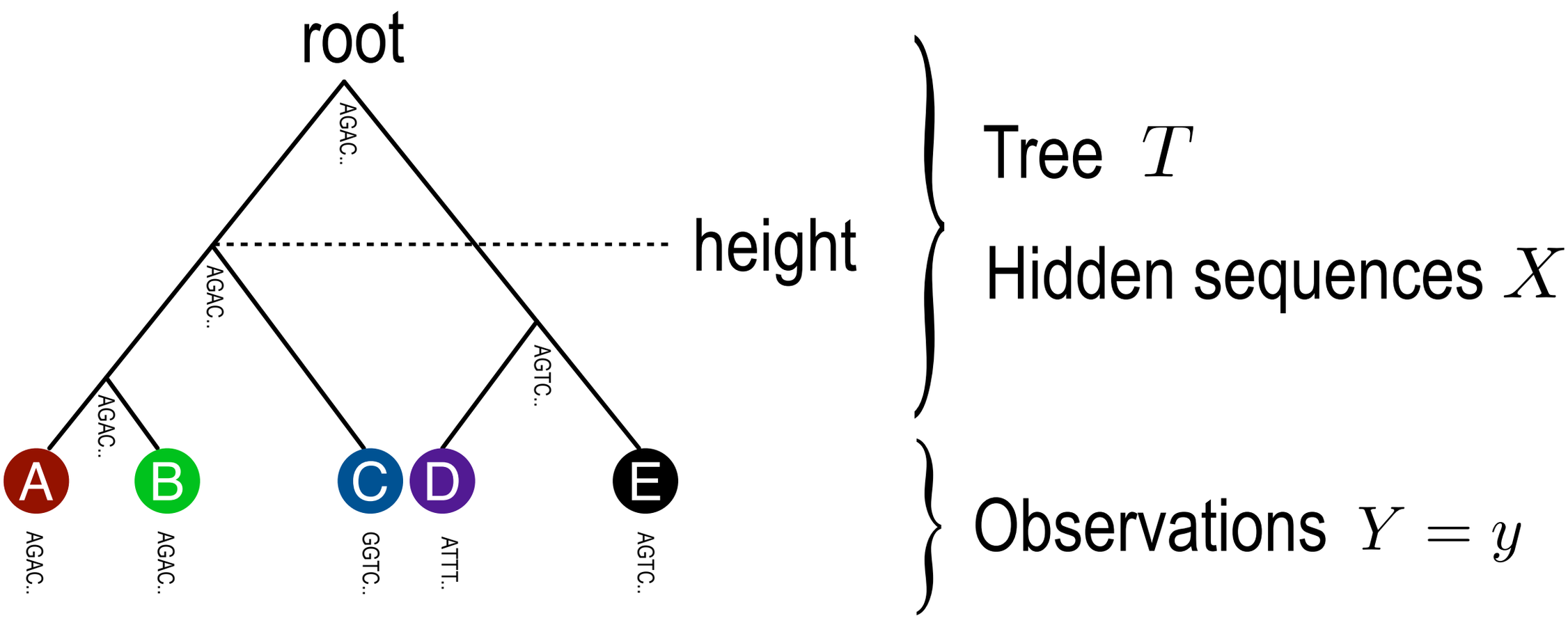
height

Tree T

Hidden sequences X

Observations $Y = y$

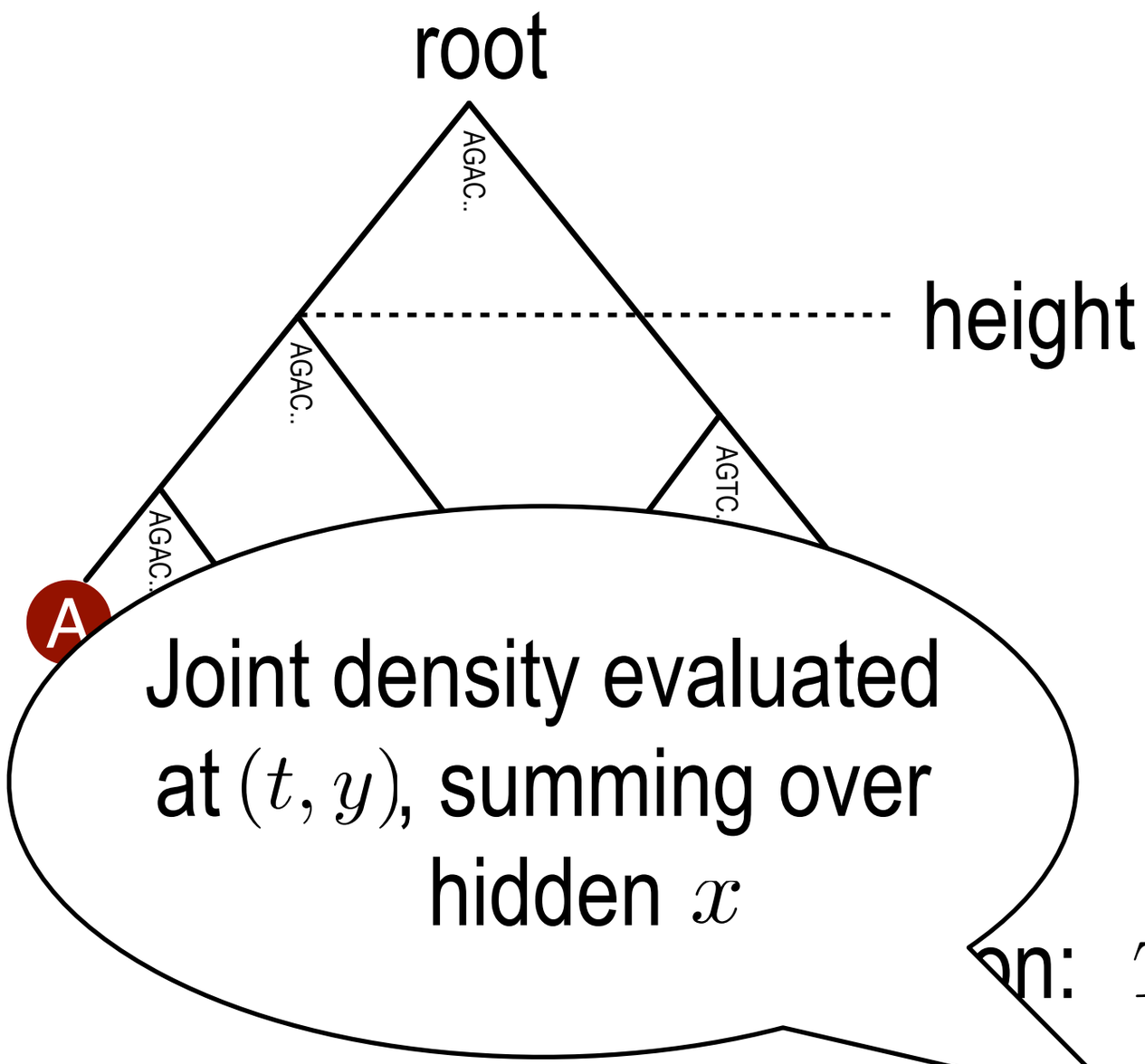
Phylogenetic setup: ultrametric trees



Target distribution: $T|Y \stackrel{d}{=} \pi$

with density: $\frac{\gamma(t)}{Z}$

Phylogenetic setup: ultrametric trees



Tree T

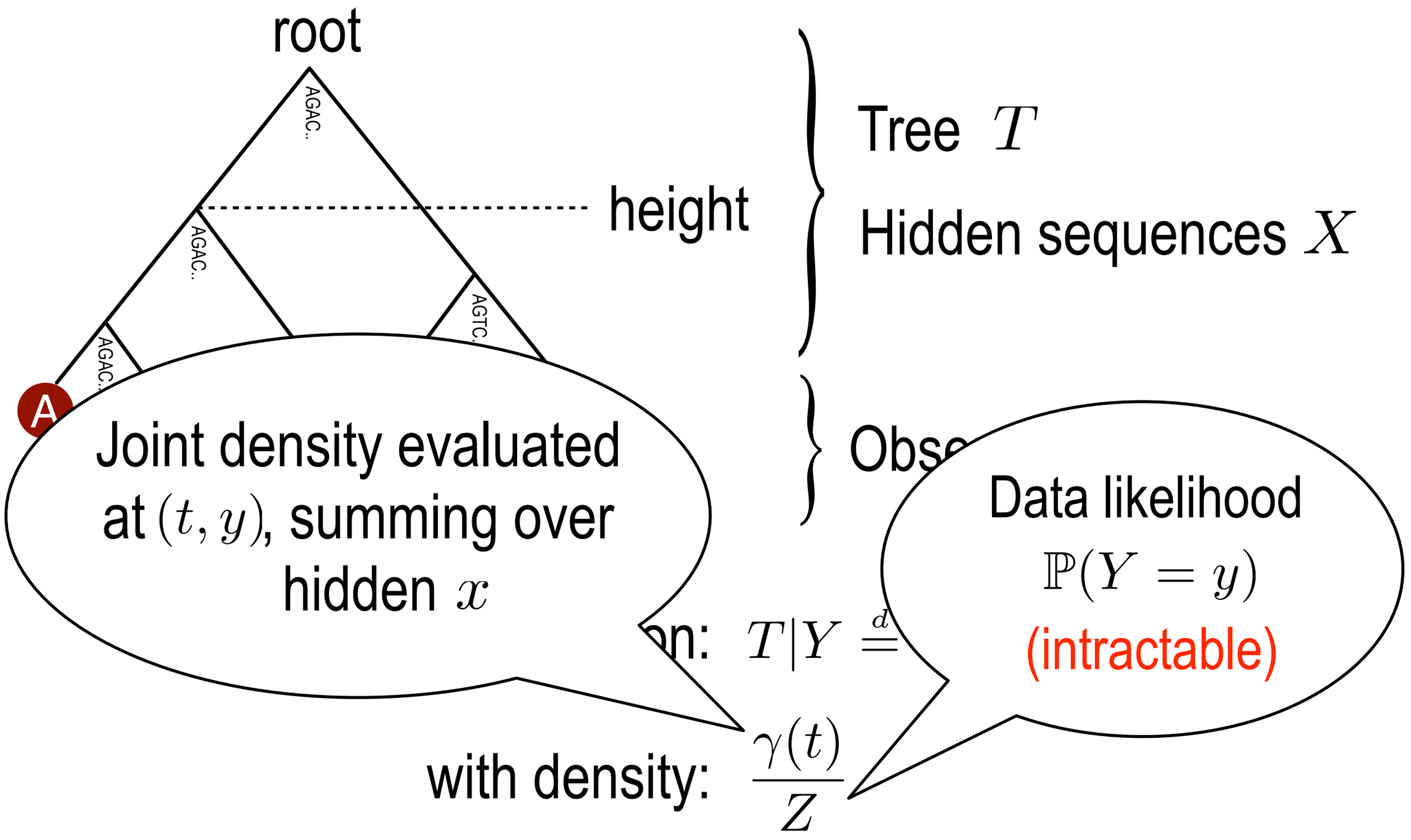
Hidden sequences X

Observations $Y = y$

on: $T|Y \stackrel{d}{=} \pi$

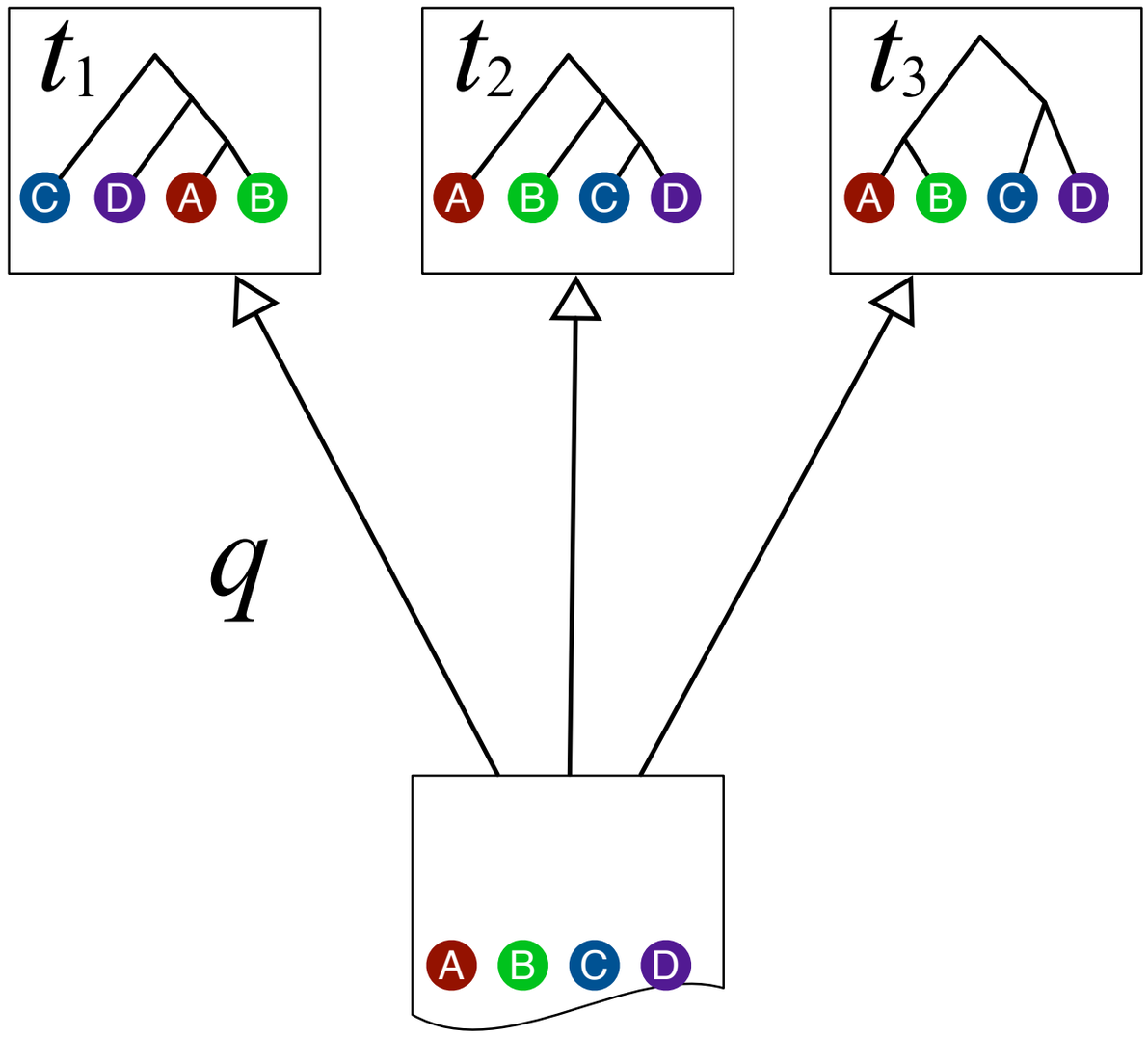
with density: $\frac{\gamma(t)}{Z}$

Phylogenetic setup: ultrametric trees



Sequential Monte Carlo (SMC)

Background: Importance Sampling (IS)

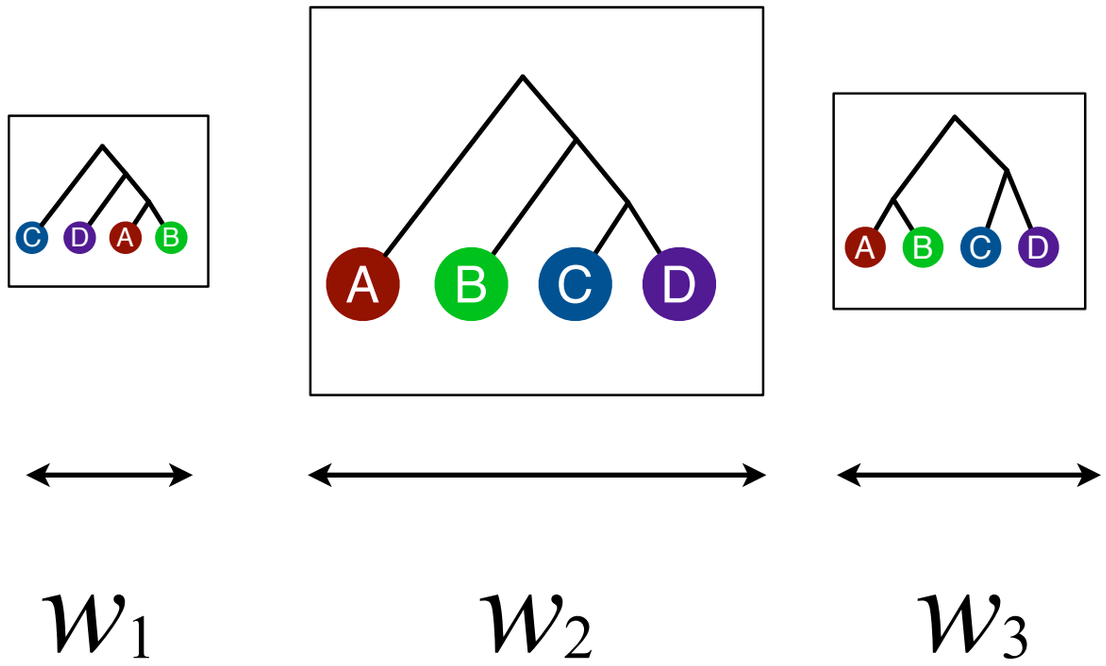


IS : Approximation for π

1. Sample trees from a proposal q: $t_i \sim q$

Sequential Monte Carlo (SMC)

Background: Importance Sampling (IS)



IS : Approximation for π

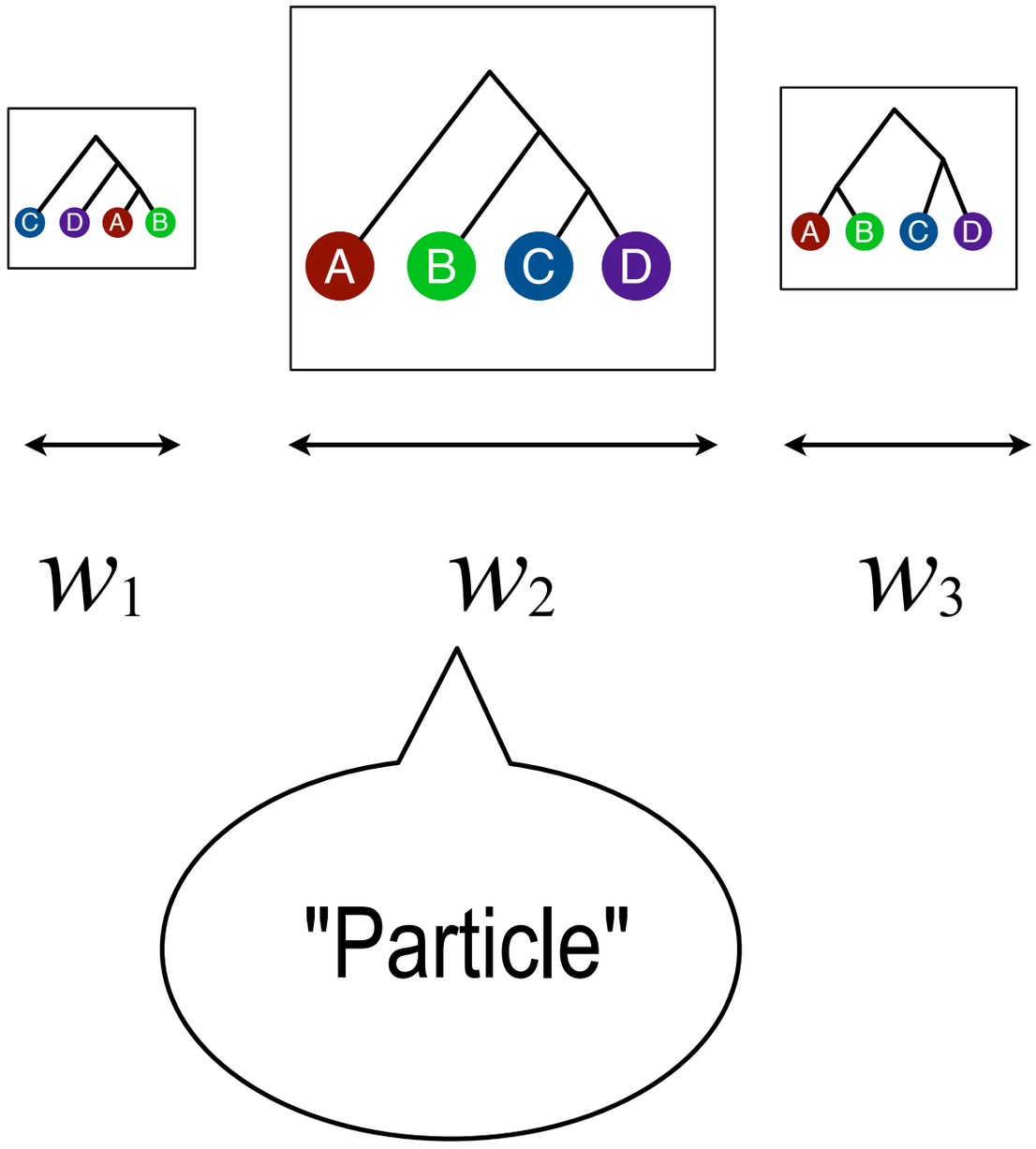
1. Sample trees from a proposal q: $t_i \sim q$
2. Compute weights

$$w_i = \gamma(t_i)/q(t_i)$$

3. Normalize weights

Sequential Monte Carlo (SMC)

Background: Importance Sampling (IS)



IS : Approximation for π

1. Sample trees from a proposal q: $t_i \sim q$
2. Compute weights

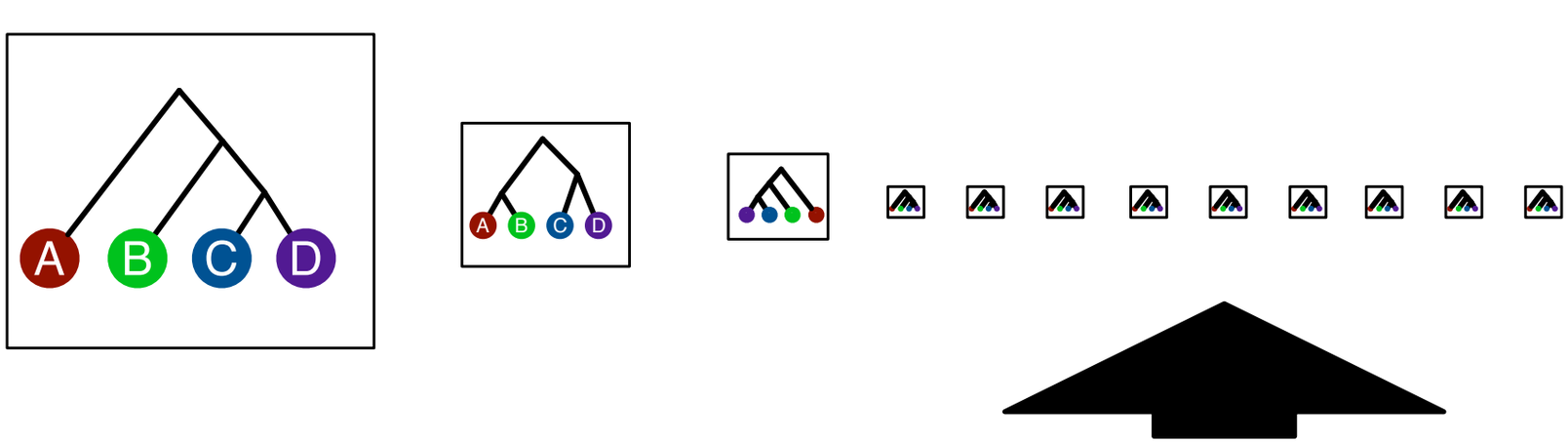
$$w_i = \gamma(t_i)/q(t_i)$$

3. Normalize weights

Sequential Monte Carlo (SMC)

Background:

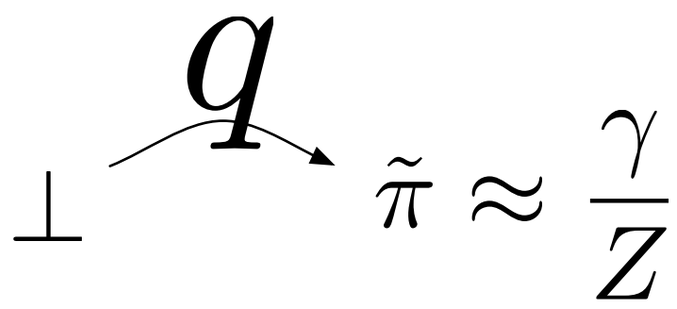
Problem with importance sampling: π is high-dimensional



Most particles will have tiny weights

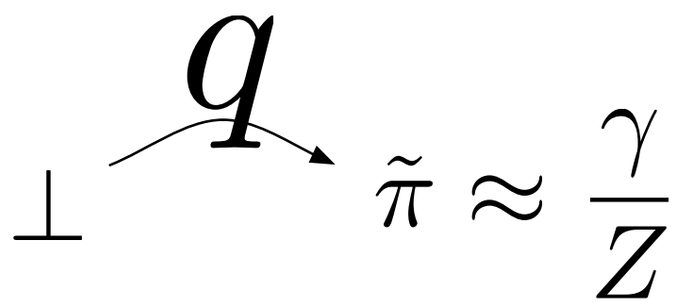
Sequential Monte Carlo (SMC)

Background: Importance Sampling

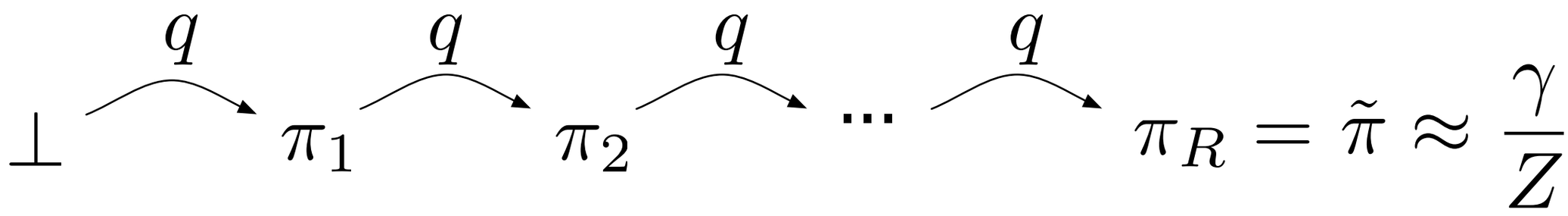


Sequential Monte Carlo (SMC)

Background: Importance Sampling

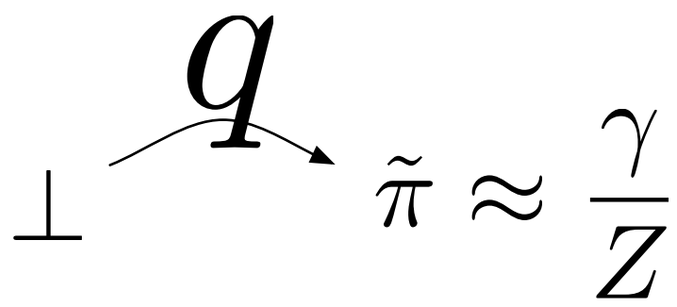


SMC: a sequence of proposals

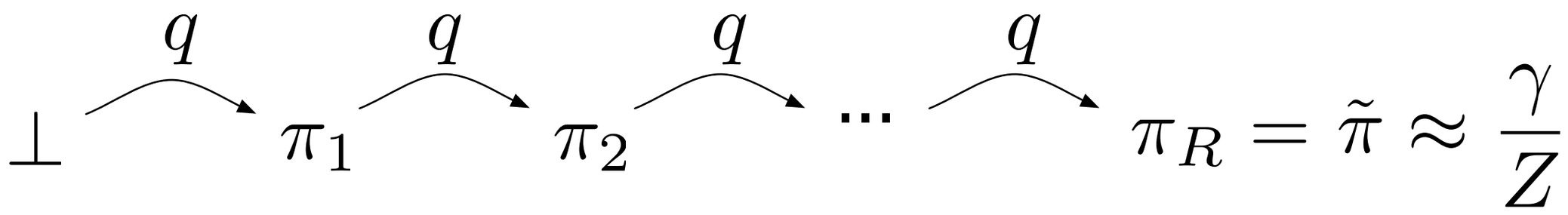


Sequential Monte Carlo (SMC)

Background: Importance Sampling

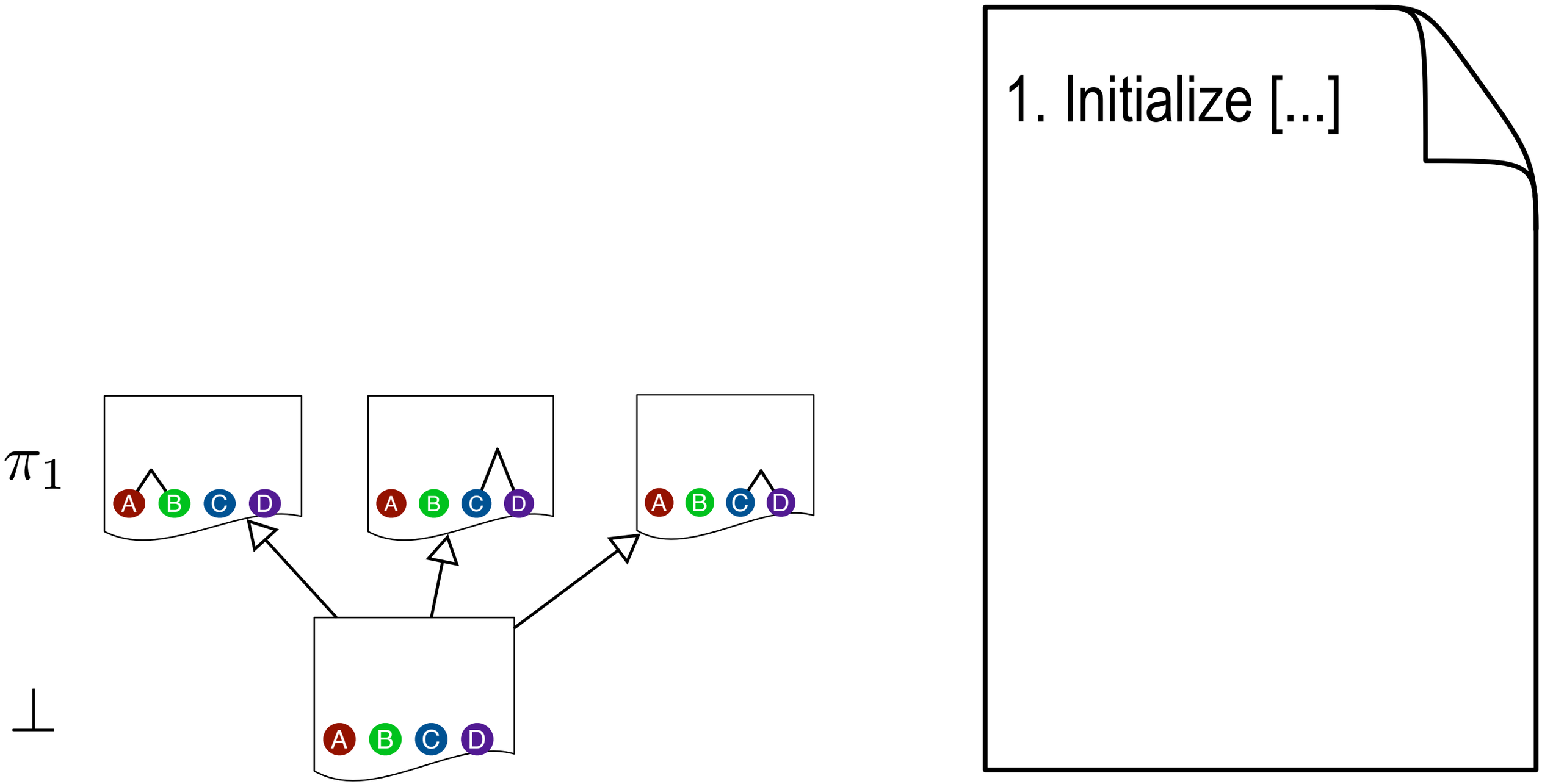


SMC: a sequence of proposals



SMC for phylogenies: π_r are distributions over partial states (forest)

Sequential Monte Carlo (SMC)



Sequential Monte Carlo (SMC)

SMC : Approximation for π

1. Initialize [...]

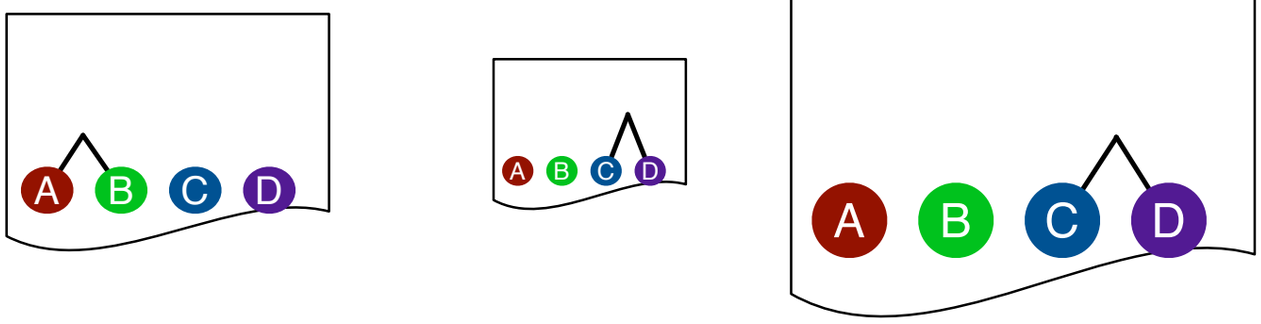
2. Iterate :

   i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

π_2

π_1



Sequential Monte Carlo (SMC)

SMC : Approximation for π

1. Initialize [...]

2. Iterate :

i. Sample partial states

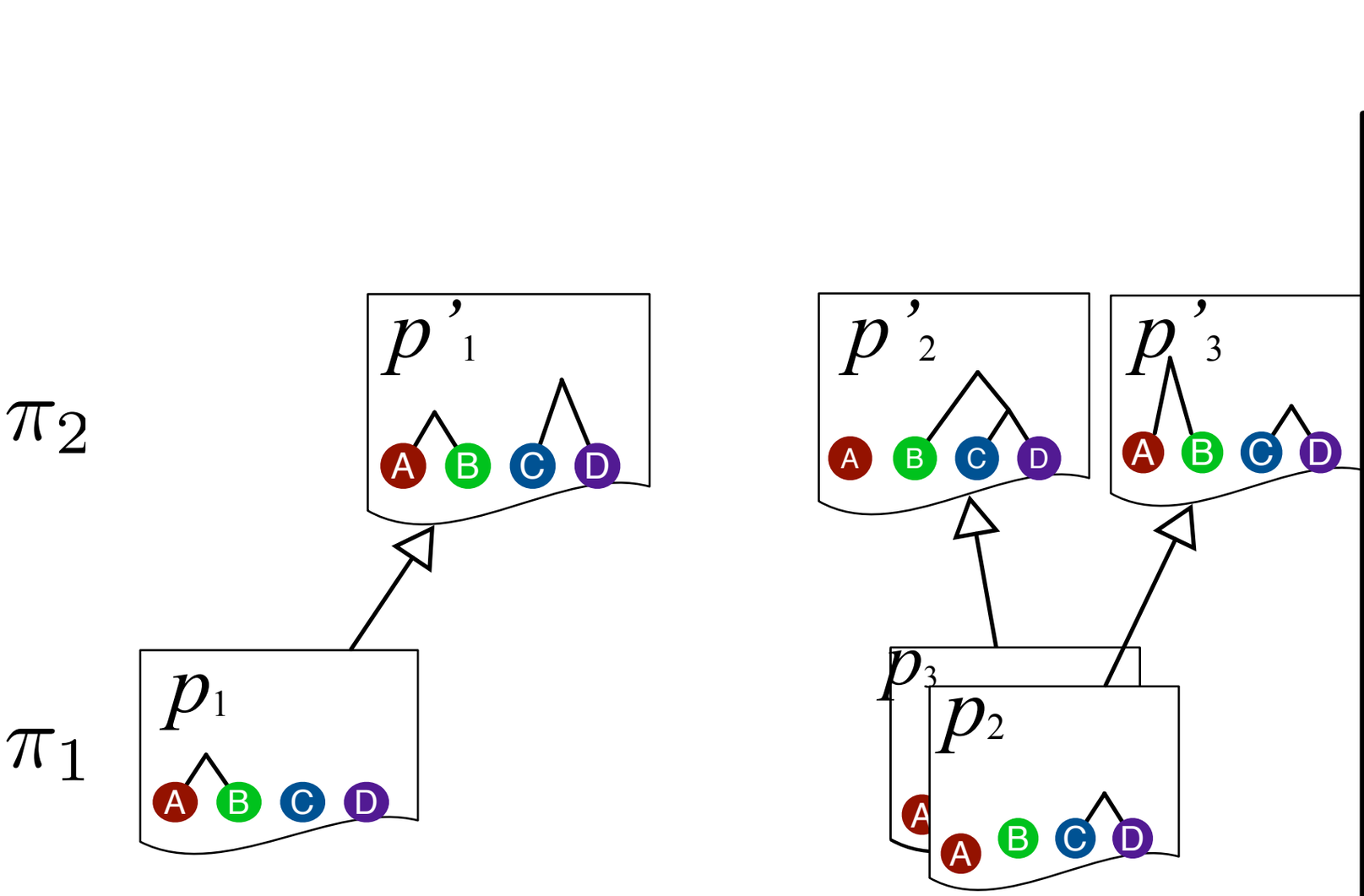
$$p'_i \sim \boxed{\pi_1} \times q$$

π_2

π_1

Sequential Monte Carlo (SMC)

SMC : Approximation for π



1. Initialize [...]

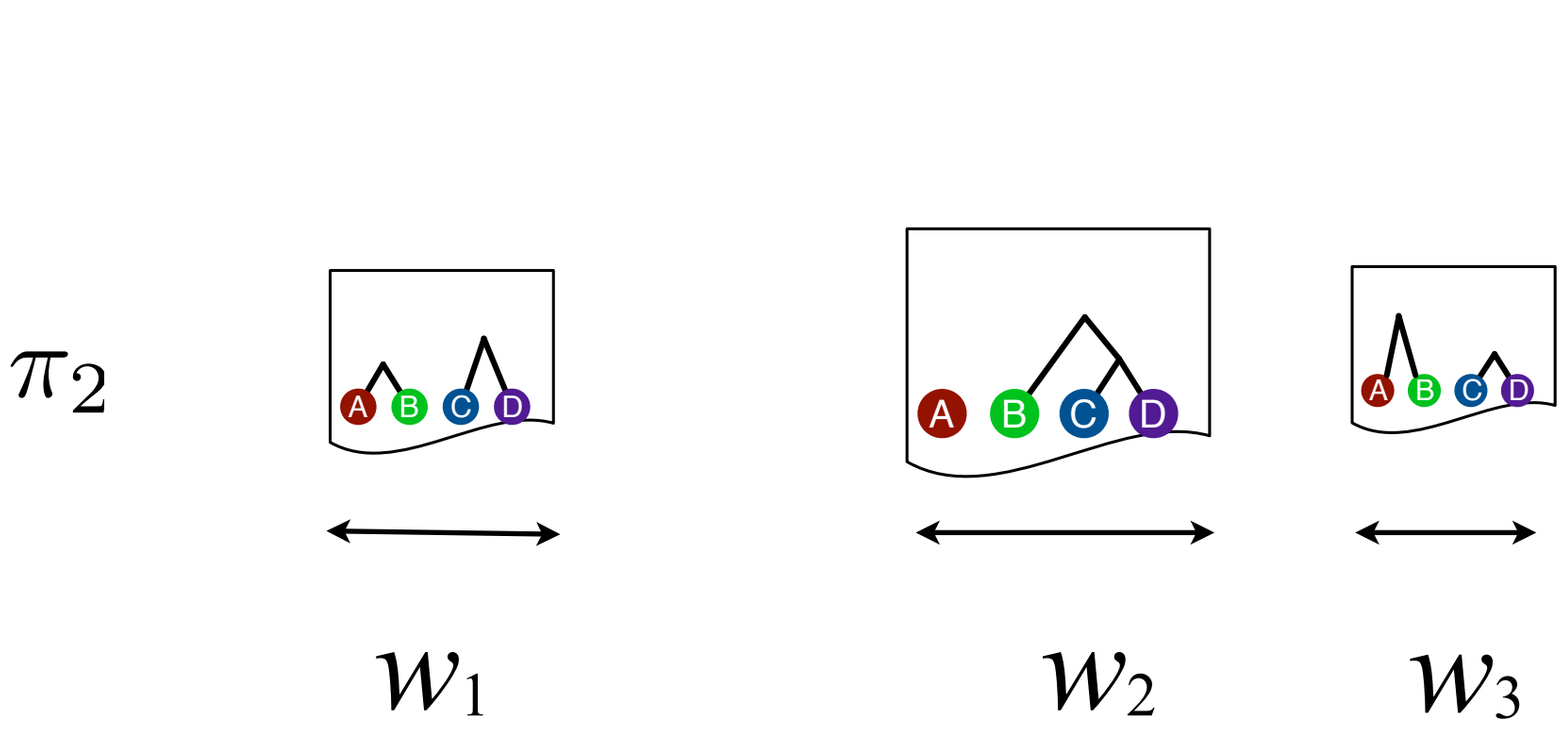
2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times \boxed{q}$$

Sequential Monte Carlo (SMC)

SMC : Approximation for π



1. Initialize [...]

2. Iterate :

i. Sample partial states

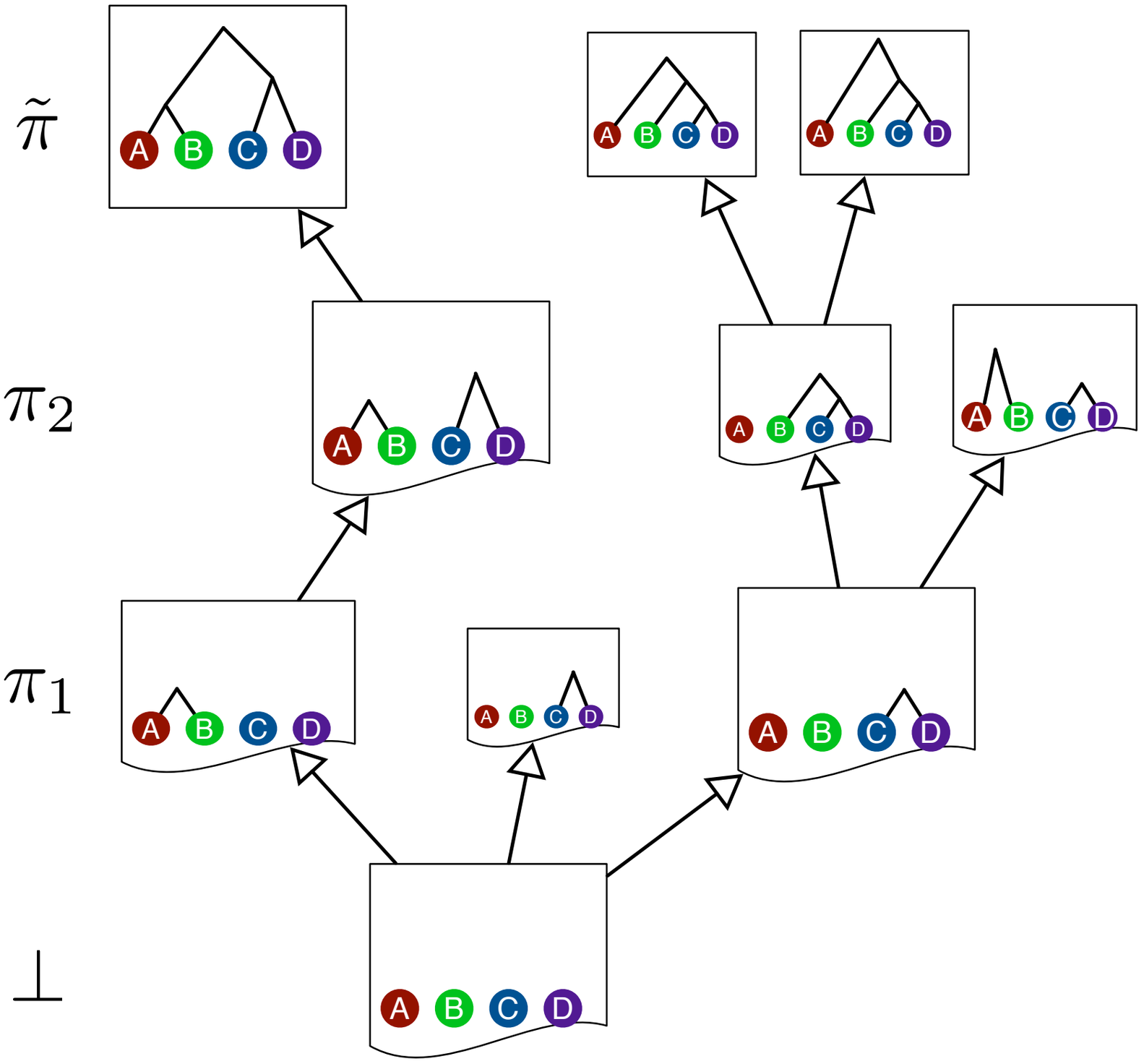
$$p'_i \sim \pi_1 \times q$$

ii. Compute weights

$$w_i = \frac{\gamma(p'_i)}{\gamma(p_i)} \frac{1}{q(p_i \rightarrow p'_i)}$$

iii. Normalize weights

Sequential Monte Carlo (SMC)



SMC : Approximation for π

1. Initialize [...]

2. Iterate :

i. Sample partial states

$$p'_i \sim \pi_1 \times q$$

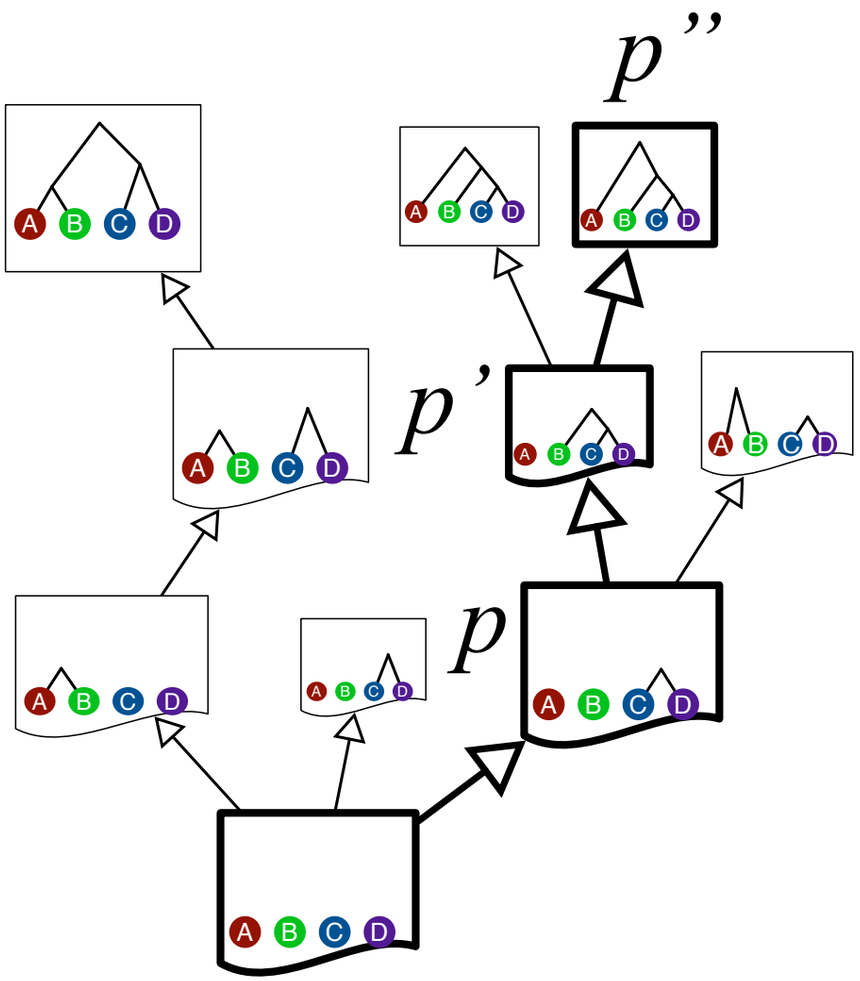
ii. Compute weights

$$w_i = \frac{\gamma(p'_i)}{\gamma(p_i)} \frac{1}{q(p_i \to p'_i)}$$

iii. Normalize weights

Intuition: why it works

Basic result: SMC is asymptotically consistent



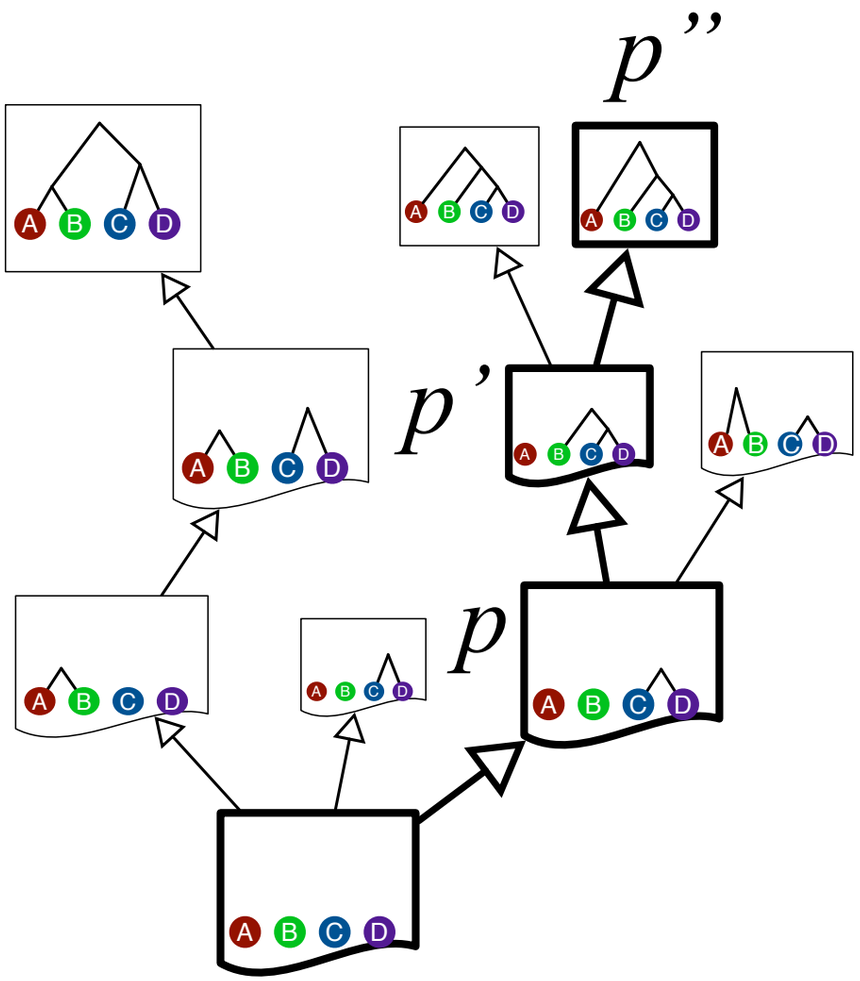
$$w'' = \frac{\gamma(p'')}{\gamma(p')} \frac{1}{q(p' \rightarrow p'')}$$

$$w' = \frac{\gamma(p')}{\gamma(p)} \frac{1}{q(p \rightarrow p')}$$

$$w = \frac{\gamma(p)}{1} \frac{1}{q(\perp \rightarrow p)}$$

Intuition: why it works

Basic result: SMC is asymptotically consistent



$$w'' = \boxed{\frac{\gamma(p'')}{\gamma(p')}} \frac{1}{q(p' \rightarrow p'')}$$

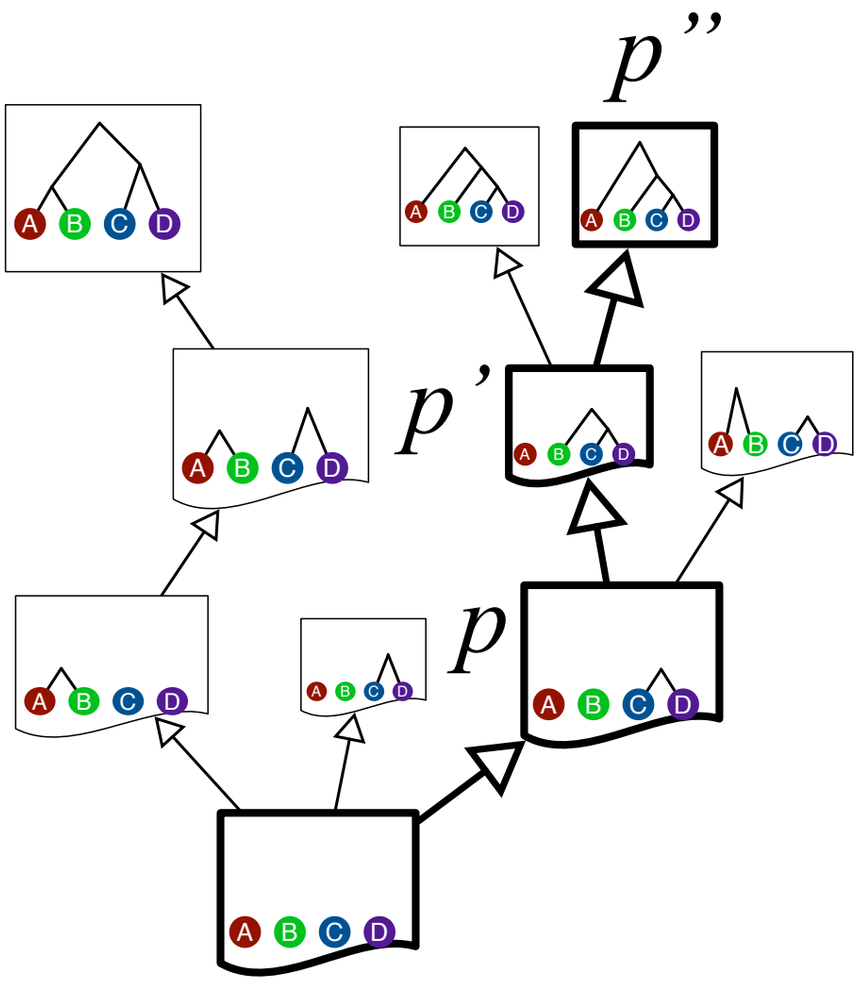
$$w' = \boxed{\frac{\gamma(p')}{\gamma(p)}} \frac{1}{q(p \rightarrow p')}$$

$$w = \boxed{\frac{\gamma(p)}{1}} \frac{1}{q(\perp \rightarrow p)}$$

$$w \cdot w' \cdot w'' = \frac{\boxed{\gamma(p'')}}{q(\perp \rightarrow p'')}$$

Intuition: why it works

Basic result: SMC is asymptotically consistent



$$w'' = \frac{\gamma(p'')}{\gamma(p')} \boxed{\frac{1}{q(p' \to p'')}}$$

$$w' = \frac{\gamma(p')}{\gamma(p)} \boxed{\frac{1}{q(p \to p')}}$$

$$w = \frac{\gamma(p)}{1} \boxed{\frac{1}{q(\bot \to p)}}$$

$$w \cdot w' \cdot w'' = \frac{\gamma(p'')}{\boxed{q(\bot \to p'')}}$$

Intuition: why it works

Compare:

Weights along a SMC path

$$w \cdot w' \cdot w'' = \frac{\gamma(p'')}{q(\perp \rightarrow p'')}$$

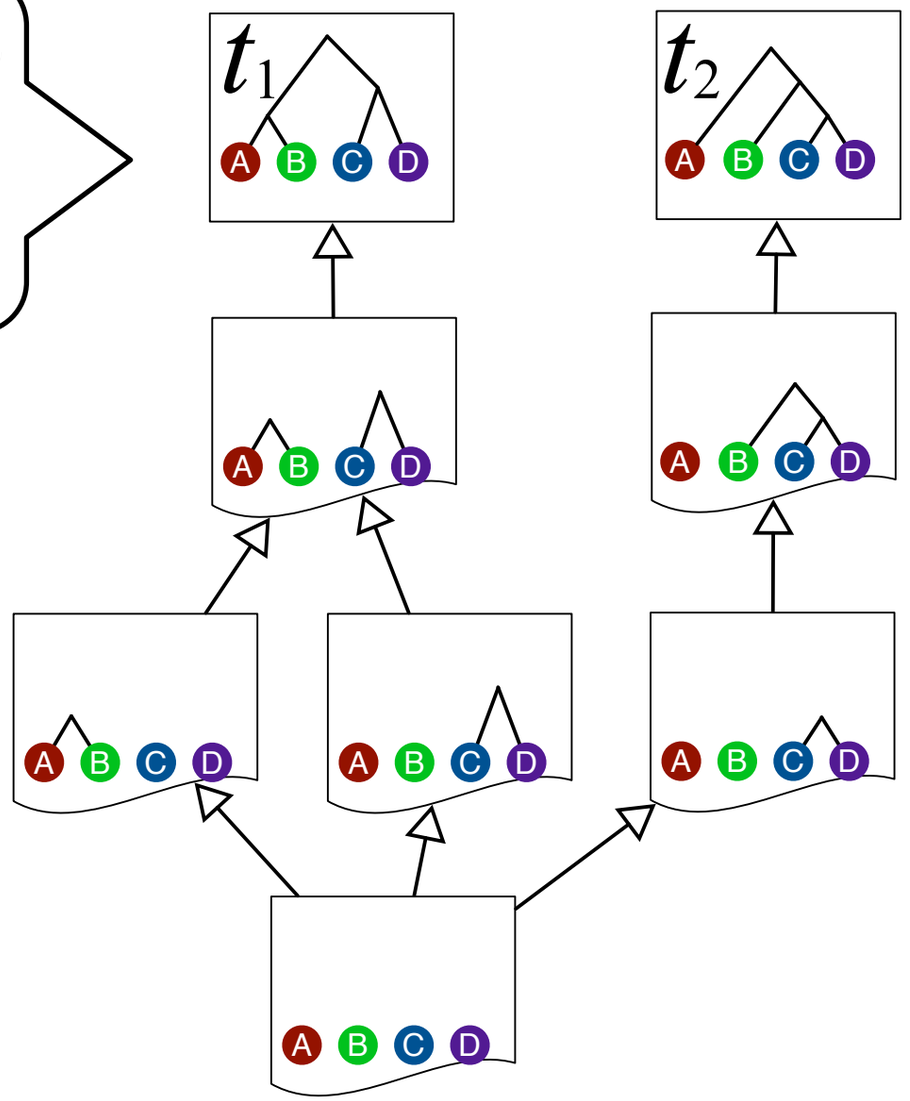
Importance sampling

$$w = \frac{\gamma(t)}{q(t)}$$

Designing a proposal q

Issue: Over-counting

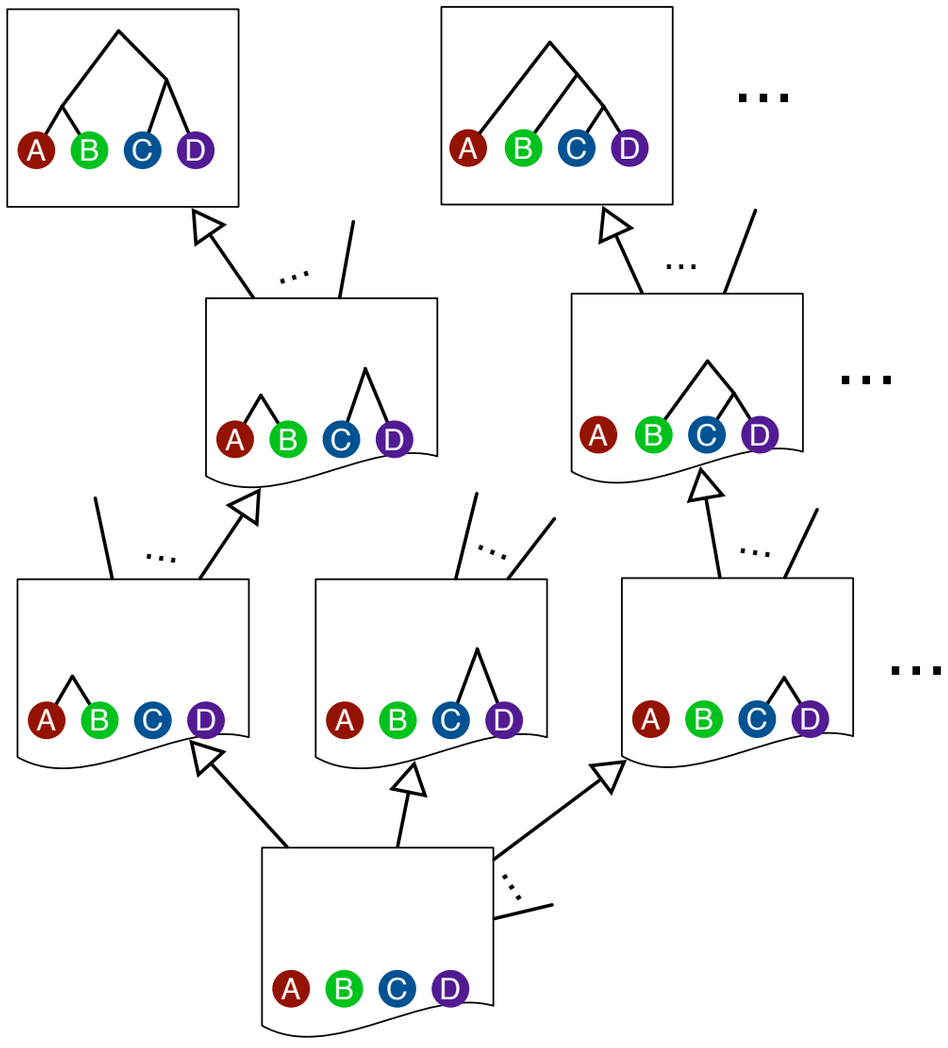
Two ways of building t_1



One way of building t_2

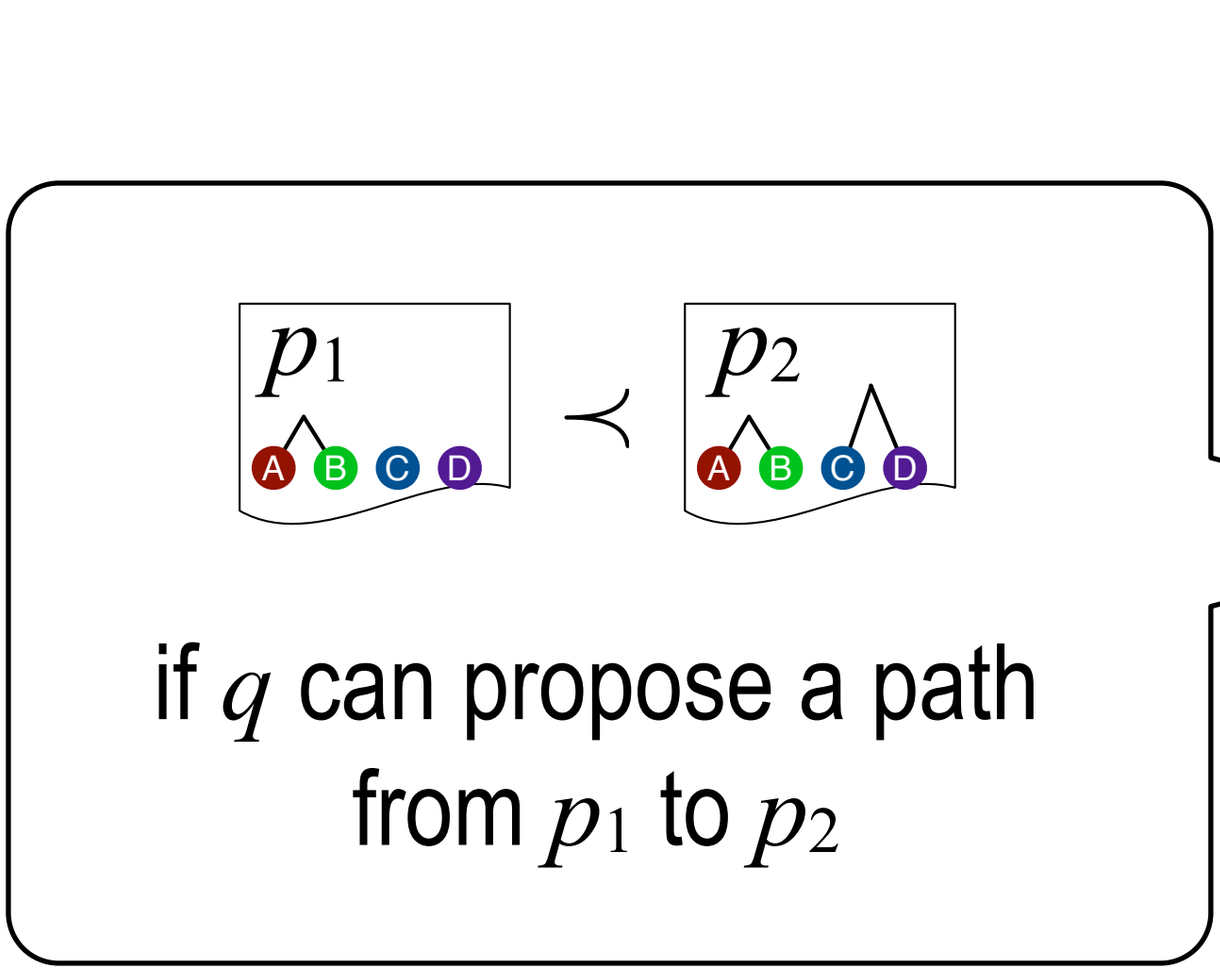
Designing a proposal q

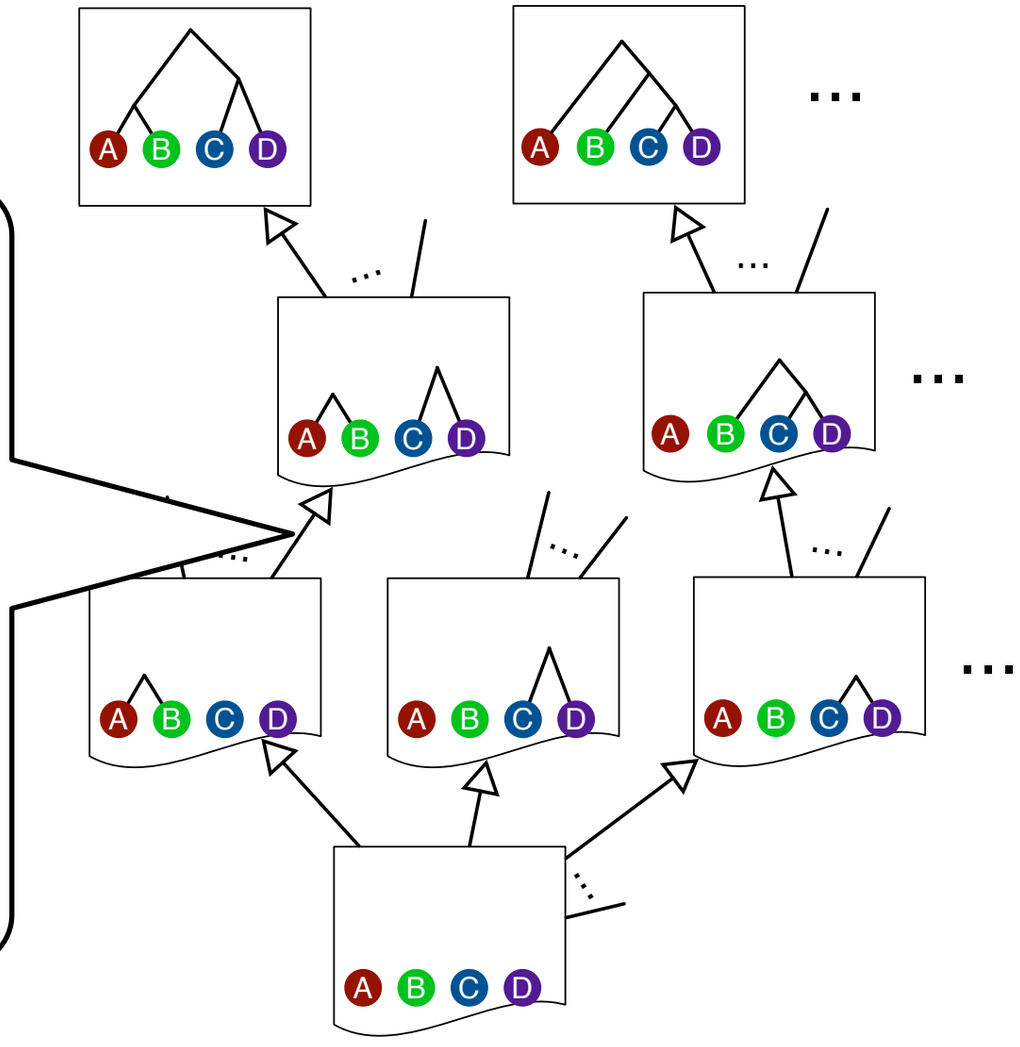
Useful abstraction: q induce a partial order (poset) P



Designing a proposal q

Useful abstraction: q induce a partial order (poset) P

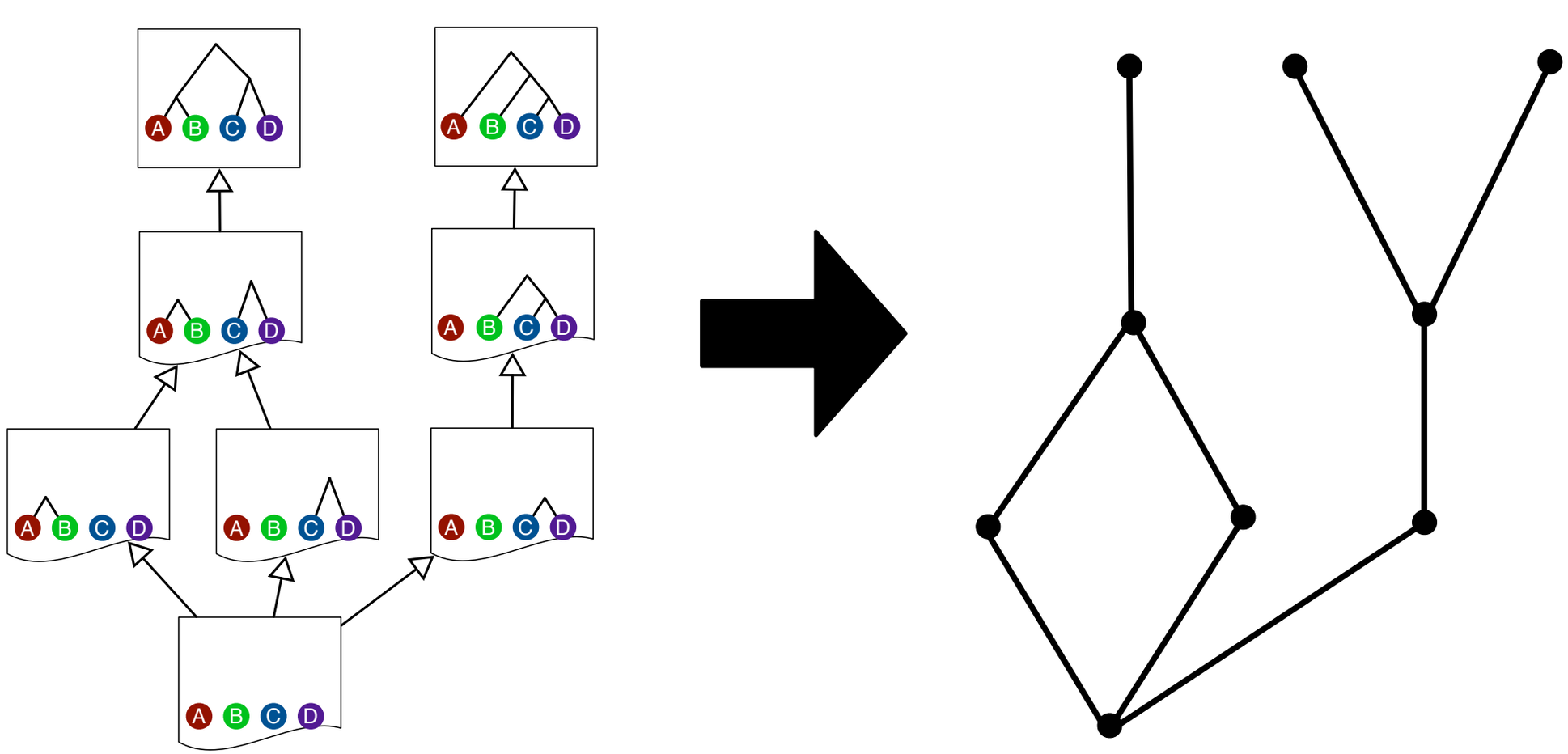




Designing a proposal q

Useful abstraction: q induce a partial order (poset) P

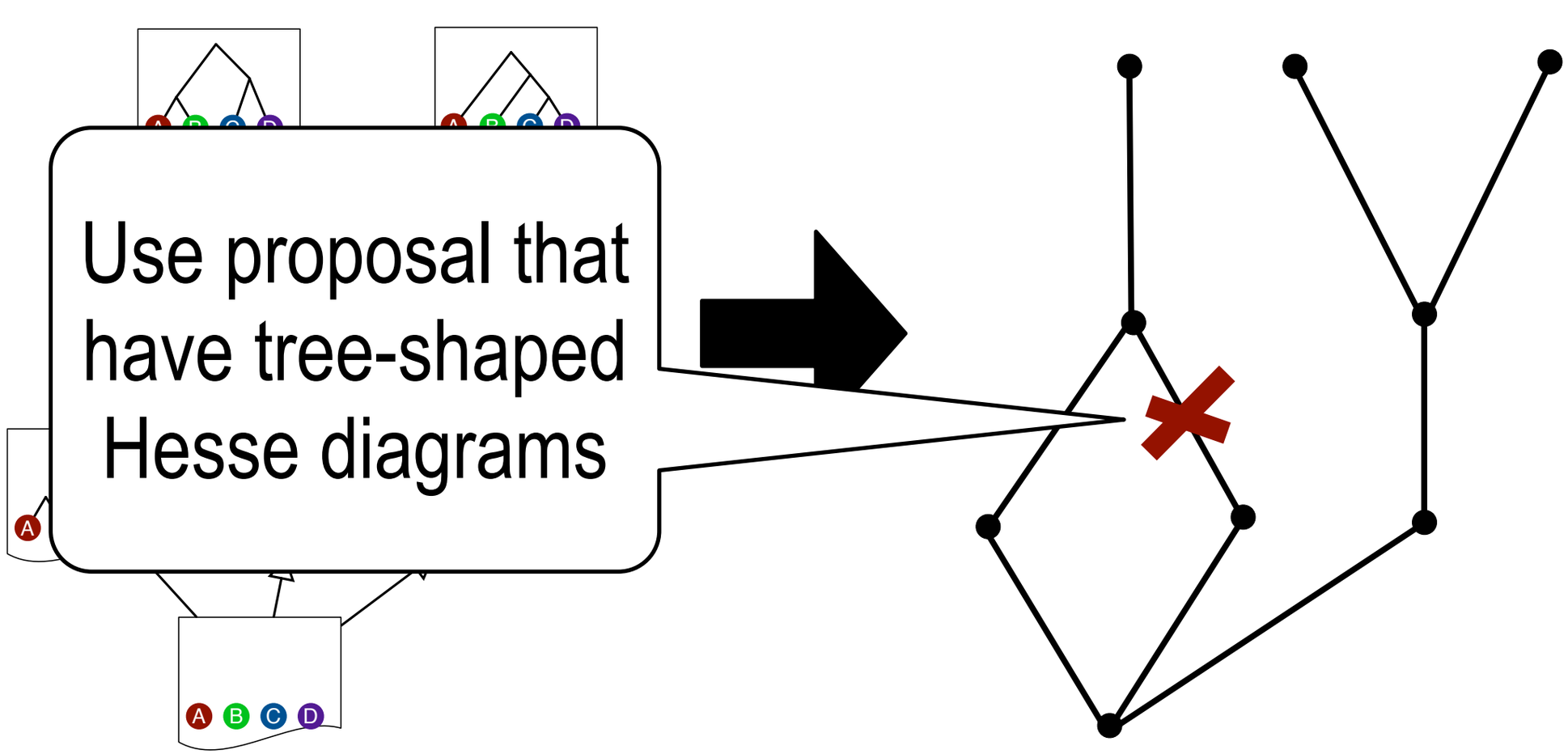
Poset's Hesse diagram:



Designing a proposal q

Useful abstraction: q induce a partial order (poset) P

Poset's Hesse diagram:



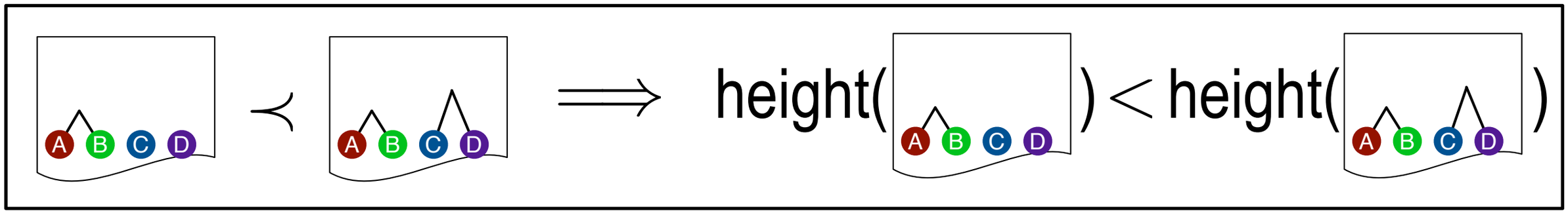
Designing a proposal q

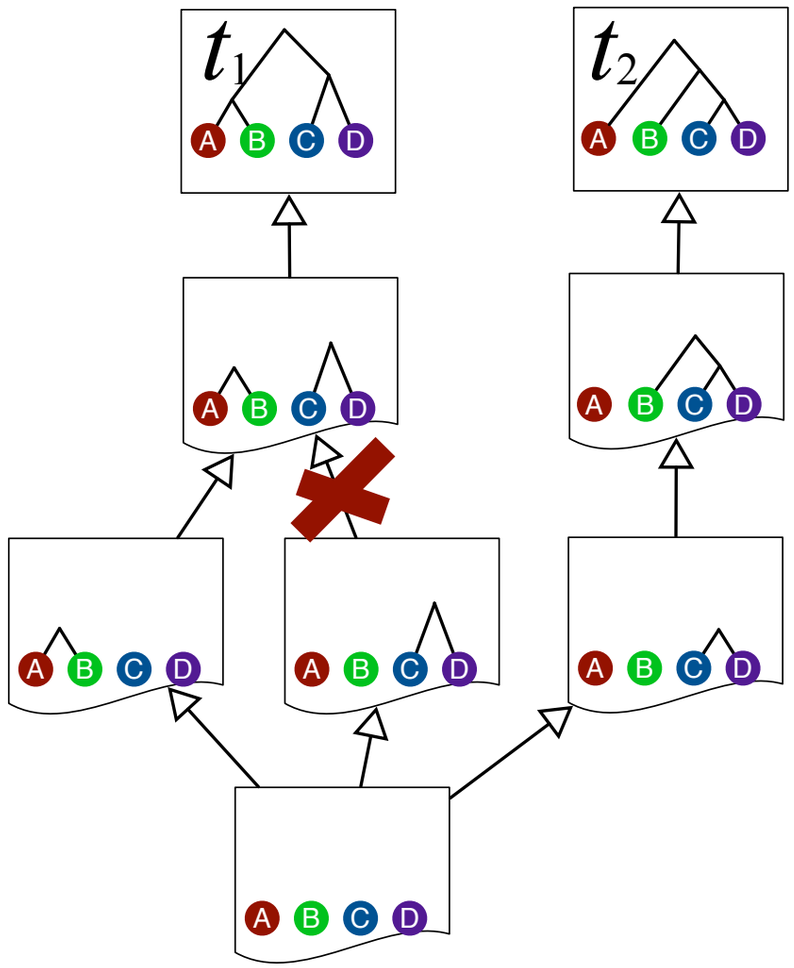
Example: a proposal that has a tree-shaped Hesse diagram.

1. Pick a pair of trees to merge uniformly at random

2. Pick a height for the new tree such that

Designing a proposal q





Experiments: setup

	Synthetic-small	Synthetic-med	Real data
Source	Generated from the model		Subset of HGDP
Likelihood model	Brownian motion on frequencies		
Number of sites	100		11,511
Number of nodes	25	51	25
Number of leaves	13	26	13

Synthetic experiments

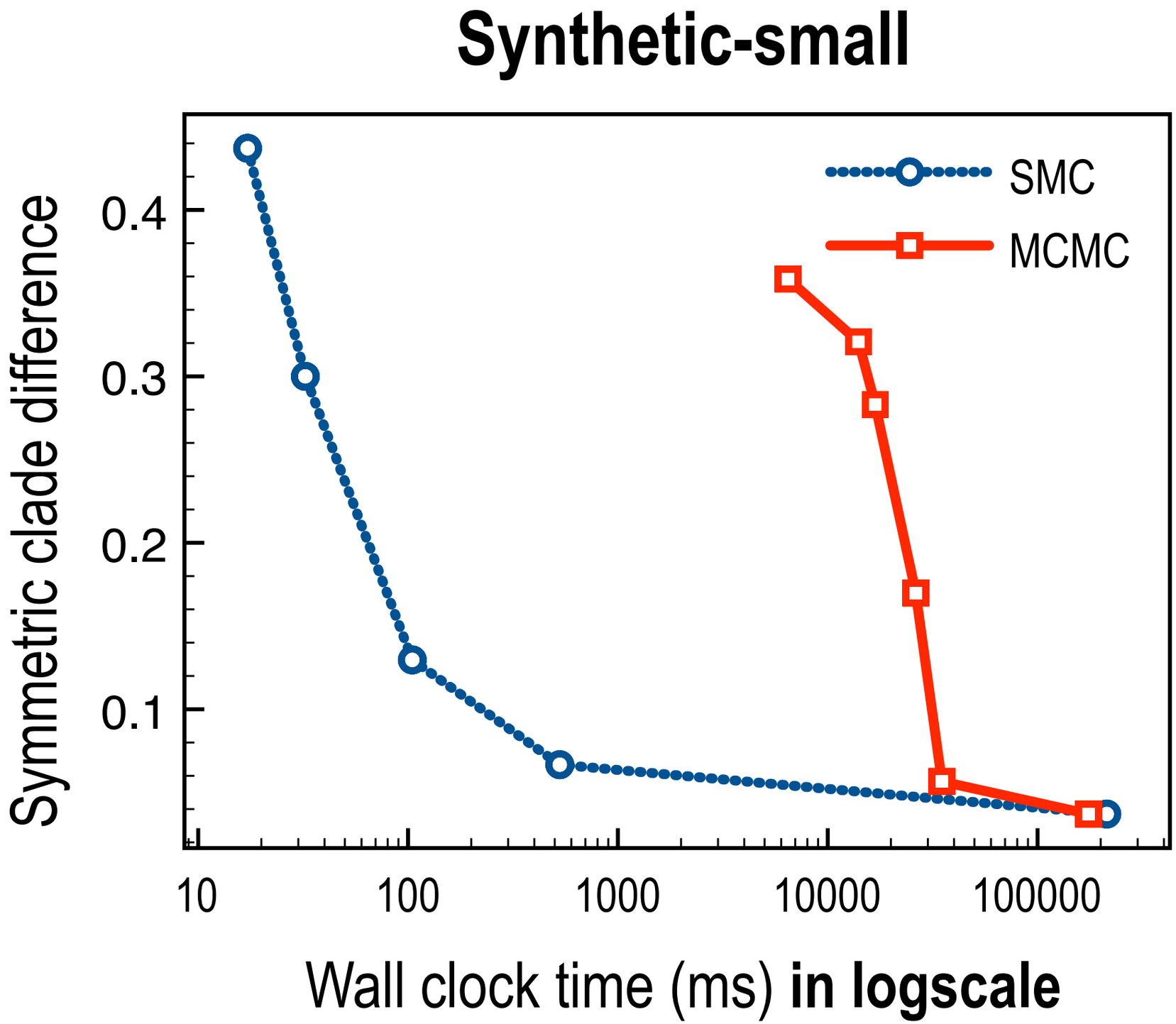
Goal: comparison against MCMC

Competitor: standard MCMC sampler, 4 tempering chains, shared sum-product implementation

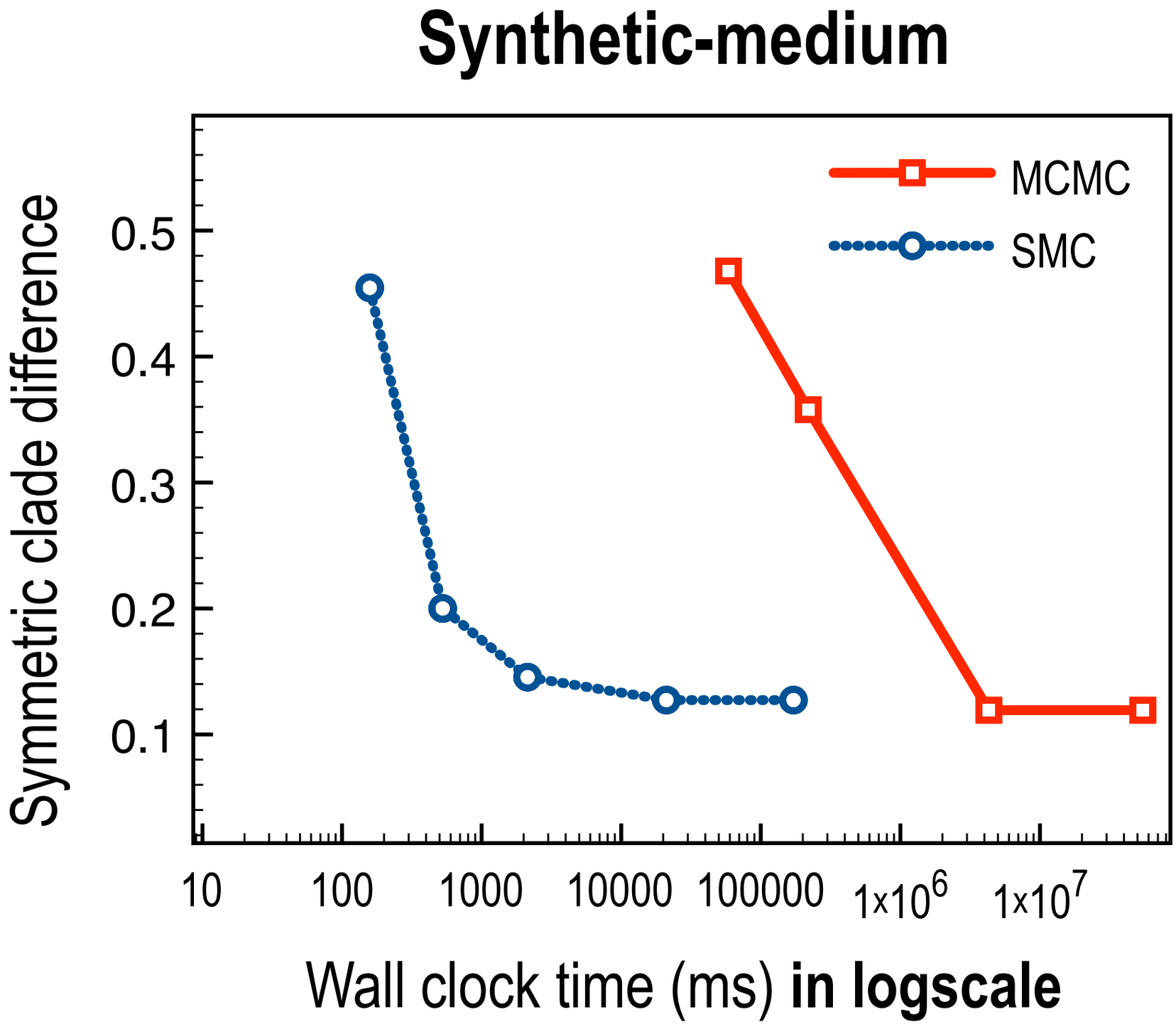
Metric: symmetric clade difference of the Minimum Bayes Risk reconstructed tree to the generating tree

Datapoints computed by increasing the number of particles (for SMC) and the number of sampling steps (for MCMC)

Comparison with MCMC



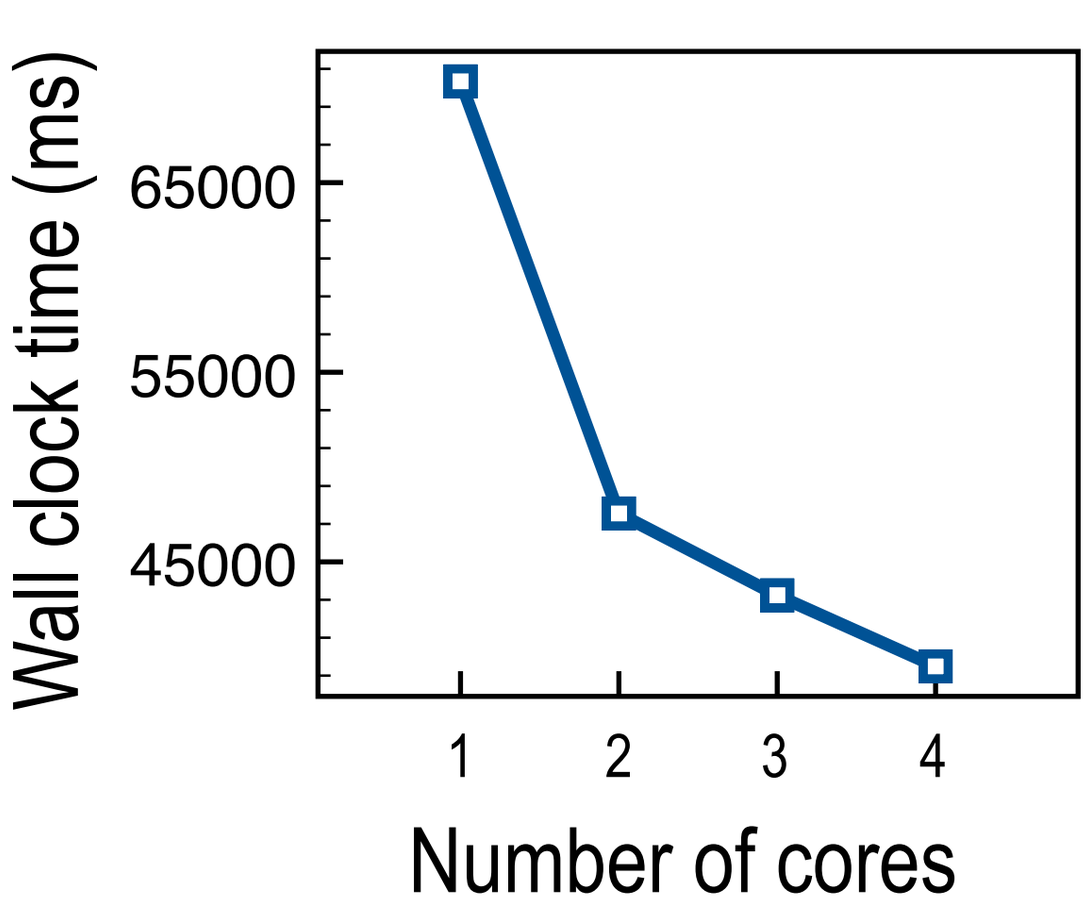
Comparison with MCMC



Experiments on real data

Goal: show that the method scales to large number of sites

Number of particle (10,000) determined using synthetic experiments, timing experiments with different numbers of cores:



Conclusion

SMC can be applied to a wide range of phylogenetic models; previous work limited to Coalescent priors [Teh et al. 07]

Order theoretic framework for designing proposals

Experiments: There are regimes where SMC outperforms MCMC

Promising applications of SMC in phylogenetic inference:

1. Quickly analyze large datasets
2. Initialization and large step proposal for MCMC chains