Variational Bayesian Phylogenetic Inference

Cheng Zhang

joint work with Frederick Matsen
Fred Hutchinson Cancer Research Center, Seattle WA

Jun 10, 2019
Introduction
The goal of **phylogenetic inference** is to reconstruct the evolution history (e.g., *phylogenetic trees*) from *molecular sequence data* (e.g., DNA, RNA or protein sequences)
Likelihood and Posterior

$\begin{align*}
\text{Evolution model:} & \quad p(ch | pa, q) \\
\text{Likelihood} & \quad p(Y | \tau, q) = \\
& \quad \prod_{i=1}^{\infty} \sum_{a_i} \eta(a_i \rho) \prod_{(u,v) \in E(\tau)} P_{au} a_{iv} (q_{uv}) \\
\text{Given a proper prior distribution} & \quad p(\tau, q), \text{ the posterior is} \\
& \quad p(\tau, q | Y) \propto p(Y | \tau, q) p(\tau, q) .
\end{align*}$
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

$q_e$: amount of evolution on $e$.

Likelihood

\[
p(Y|\tau, q) = \eta(a_\rho^i) \prod_{(u,v) \in E(\tau)} P_{a_u^i a_v^i}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y | \tau, q) = \sum a^i \eta(a^i) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(\text{ch}|\text{pa}, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_{\rho}) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_{\rho}) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]
Likelihood and Posterior

Evolution model:
\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood
\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_\rho) \prod_{(u,v)\in E(\tau)} P_{a_u^i a_v^i}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_{\rho}) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_\rho) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[ p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a_{\rho}^i) \prod_{(u,v) \in E(\tau)} P_{a_u^i, a_v^i}(q_{uv}) \]
Likelihood and Posterior

Evolution model:

\[ p(ch|pa, q_e) \]

\( q_e \): amount of evolution on \( e \).

Likelihood

\[
p(Y|\tau, q) = \prod_{i=1}^{M} \sum_{a^i} \eta(a^i_\rho) \prod_{(u,v) \in E(\tau)} P_{a^i_u a^i_v}(q_{uv})
\]

Given a proper prior distribution \( p(\tau, q) \), the posterior is

\[
p(\tau, q|Y) \propto p(Y|\tau, q)p(\tau, q).
\]
Random-walk MCMC: use simple random perturbation (e.g., Nearest Neighborhood Interchange) to generate new state

Challenges

- Large search space (combinatorially exploding)

\[ \text{# unrooted trees (n taxa)} = (2n - 5)!! \]

- Intertwined parameter space, low acceptance rate, hard to scale to data sets with many sequences.
Variational Bayesian Phylogenetic Inference
Encode tree structures via Bayesian networks!
Encode tree structures via **Bayesian networks**!
Encode tree structures via Bayesian networks!

Subsplit Bayesian Networks
Subsplit Bayesian Networks

Encode tree structures via Bayesian networks!
Subsplit Bayesian Networks

Encode tree structures via **Bayesian networks**!

![Diagram of Bayesian networks and tree structures]

- $S_1 ightarrow S_2 ightarrow S_4$
- $S_1 ightarrow S_3 ightarrow S_6 ightarrow S_7$
- $ABC \rightarrow \frac{ABC}{D}$
- $A \rightarrow \frac{BC}{A}$
- $B \rightarrow \frac{C}{B}$
- $\frac{D}{A}$
- $\frac{D}{B}$
- $\frac{D}{C}$
- $\frac{D}{D}$

对应树结构：

- $A 
  \rightarrow B$
- $C 
  \rightarrow D$
Probability Estimation Over Tree Topologies

Rooted Trees

\[ p_{\text{sbn}}(T = \tau) = p(S_1 = s_1) \prod_{i>1} p(S_i = s_i | S_{\pi_i} = s_{\pi_i}) \]
Unrooted Trees:

\[ p_{\text{bsn}}(T^u = \tau) = \sum_{s_1 \sim \tau} p(S_1 = s_1) \prod_{i > 1} p(S_i = s_i | S_{\pi_i} = s_{\pi_i}). \]
Use SBNs to refine posterior probability estimation given MCMC samples, useful when the posterior is diffuse.

<table>
<thead>
<tr>
<th>Data set</th>
<th>(#Taxa, #Sites)</th>
<th>Tree space size</th>
<th>Sampled trees</th>
<th>KL divergence to ground truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SRF</td>
</tr>
<tr>
<td>DS1</td>
<td>(27, 1949)</td>
<td>$5.84 \times 10^{32}$</td>
<td>1228</td>
<td>0.0155</td>
</tr>
<tr>
<td>DS2</td>
<td>(29, 2520)</td>
<td>$1.58 \times 10^{35}$</td>
<td>7</td>
<td><strong>0.0122</strong></td>
</tr>
<tr>
<td>DS3</td>
<td>(36, 1812)</td>
<td>$4.89 \times 10^{47}$</td>
<td>43</td>
<td>0.3539</td>
</tr>
<tr>
<td>DS4</td>
<td>(41, 1137)</td>
<td>$1.01 \times 10^{57}$</td>
<td>828</td>
<td>0.5322</td>
</tr>
<tr>
<td>DS5</td>
<td>(50, 378)</td>
<td>$2.84 \times 10^{74}$</td>
<td>33752</td>
<td>11.5746</td>
</tr>
<tr>
<td>DS6</td>
<td>(50, 1133)</td>
<td>$2.84 \times 10^{74}$</td>
<td>35407</td>
<td>10.0159</td>
</tr>
<tr>
<td>DS7</td>
<td>(59, 1824)</td>
<td>$4.36 \times 10^{92}$</td>
<td>1125</td>
<td>1.2765</td>
</tr>
<tr>
<td>DS8</td>
<td>(64, 1008)</td>
<td>$1.04 \times 10^{103}$</td>
<td>3067</td>
<td>2.1653</td>
</tr>
</tbody>
</table>

[Zhang and Matsen, NeurIPS 2018]

**Remark:** Unlike previous methods, SBNs are flexible enough to provide accurate approximations to real data posteriors!
An optimization approach

\[ q^* = \arg \min_{q \in Q} D_{KL} (q(\theta) || p(\theta|D)) \]

\( Q \): approximating distributions.
Variational Inference

An optimization approach

\[ q^* = \arg \min_{q \in Q} D_{KL} (q(\theta) \| p(\theta | D)) \]

\( Q \): approximating distributions.

equivalent to maximizing the evidence lower bound (ELBO)

\[ L(q, D) = \mathbb{E}_{q(\theta)} \log \left( \frac{p(\theta, D)}{q(\theta)} \right) \leq \log p(D) \]

- tends to be faster than MCMC
- easy to scale to large data sets (via stochastic gradient ascent)
Approximating Distribution:

\[ Q_{\phi}(\tau) \]

tree topology

Multi-sample Lower Bound:

\[ L_K(\phi, \psi) = \mathbb{E}_{Q_{\phi, \psi}(\tau_1:K, q_1:K)} \log \left( \frac{1}{K} \sum_{i=1}^{K} p(Y|\tau_i, q_i)p(\tau_i, q_i)Q_{\phi}(\tau_i)Q_{\psi}(q_i|\tau_i) \right) \]
Approximating Distribution:

\[ Q_{\phi,\psi}(\tau, q) \triangleq Q_{\phi}(\tau) \cdot Q_{\psi}(q|\tau) \]
Variational Bayesian Phylogenetic Inference

- **Approximating Distribution:**

\[ Q_{\phi, \psi}(\tau, q) \triangleq Q_\phi(\tau) \cdot Q_\psi(q | \tau) \]

- **Multi-sample Lower Bound:**

\[
L^K(\phi, \psi) = \mathbb{E}_{Q_{\phi, \psi}(\tau^{1:K}, q^{1:K})} \log \left( \frac{1}{K} \sum_{i=1}^{K} \frac{p(Y | \tau^i, q^i)p(\tau^i, q^i)}{Q_\phi(\tau^i)Q_\psi(q^i | \tau^i)} \right)
\]
Approximating Distribution:

\[ Q_{\phi,\psi}(\tau, q) \triangleq Q_{\phi}(\tau) \cdot Q_{\psi}(q|\tau) \]

Multi-sample Lower Bound:

\[ L^K(\phi, \psi) = \mathbb{E}_{Q_{\phi,\psi}(\tau^{1:K}, q^{1:K})} \log \left( \frac{1}{K} \sum_{i=1}^{K} \frac{p(Y|\tau^i, q^i)p(\tau^i, q^i)}{Q_{\phi}(\tau^i)Q_{\psi}(q^i|\tau^i)} \right) \]

Variational Bayesian Phylogenetic Inference:

\[ \hat{\phi}, \hat{\psi} = \arg \max_{\phi, \psi} L^K(\phi, \psi) \]

parameters trained via stochastic gradient ascent (SGA).
$Q_\phi(\tau)$
Variational Bayesian Phylogenetic Inference

\[ Q_\phi(\tau) \]

\( \text{sample} \)

e.g., ancestral sampling for SBNs

\[ L_K(\phi, \psi) \]

\[ \text{multi-sample lower bound} \]

SGA update
Variational Bayesian Phylogenetic Inference

\[ Q_\phi(\tau) \]

\[ Q_\psi(q|\tau) \]

e.g., \textit{ancestral sampling} for SBNs
Variational Bayesian Phylogenetic Inference

\[ Q_\phi(\tau) \] sample

\begin{array}{c}
\tau^1 \\
A & B \\
C & D
\end{array}

\[ Q_\psi(q|\tau) \] sample

\begin{array}{c}
\tau^2 \\
A & C \\
B & D
\end{array}

\[ \ldots \]

\begin{array}{c}
\tau^K \\
A & B \\
D & C
\end{array}

\begin{array}{c}
A & B \\
C & D
\end{array}

\begin{array}{c}
(\tau^1, q^1) \\
(\tau^2, q^2) \\
\ldots \\
(\tau^K, q^K)
\end{array}

e.g., ancestral sampling for SBNs

e.g., Lognormal for branch lengths
Variational Bayesian Phylogenetic Inference

\[ Q_{\phi}(\tau) \]

\[ Q_{\psi}(q|\tau) \]

\[ L^K(\phi, \psi) \]

e.g., ancestral sampling for SBNs

e.g., Lognormal for branch lengths

multi-sample lower bound
Variational Bayesian Phylogenetic Inference

$Q_\phi(\tau)$

Sample

$Q_\psi(q|\tau)$

Sample

$L^K(\phi, \psi)$

Multi-sample lower bound

SGA update

e.g., ancestral sampling for SBNs

e.g., Lognormal for branch lengths

$\varepsilon$ sample
Structured Parameterization

**SBNs Parameters**

\[
p(S_1 = s_1) = \frac{\exp(\phi_{s_1})}{\sum_{s_r \in S_r} \exp(\phi_{s_r})}, \quad p(S_i = s | S_{\pi_i} = t) = \frac{\exp(\phi_{s|t})}{\sum_{s \in S_{\pi_i}} \exp(\phi_{s|t})}
\]

**Branch Length Parameters**

\[
Q_\psi(q|\tau) = \prod_{e \in E(\tau)} p^{\text{Lognormal}}(q_e | \mu(e, \tau), \sigma(e, \tau))
\]

▶ **Simple Split**

\[
\mu_s(e, \tau) = \psi_{e/\tau}^\mu, \quad \sigma_s(e, \tau) = \psi_{e/\tau}^\sigma.
\]
Structured Parameterization

**SBNs Parameters**

\[
p(S_1 = s_1) = \frac{\exp(\phi_{s_1})}{\sum_{s_r \in S_r} \exp(\phi_{s_r})}, \quad p(S_i = s_i|S_{\pi_i} = t) = \frac{\exp(\phi_{s_t})}{\sum_{s \in S_{\pi}} \exp(\phi_{s_i})}
\]

**Branch Length Parameters**

\[
Q_{\psi}(q|\tau) = \prod_{e \in E(\tau)} p^{\text{Lognormal}}(q_e | \mu(e, \tau), \sigma(e, \tau))
\]

- **Simple Split**
  \[
  \mu_s(e, \tau) = \psi_{e/\tau}^\mu, \quad \sigma_s(e, \tau) = \psi_{e/\tau}^\sigma.
  \]

- **Primary Subsplit Pair (PSP)**
  \[
  \mu_{\text{psp}}(e, \tau) = \psi_{e/\tau}^\mu + \sum_{s \in e/\tau} \psi_{s}^\mu,
  \]
  \[
  \sigma_{\text{psp}}(e, \tau) = \psi_{e/\tau}^\sigma + \sum_{s \in e/\tau} \psi_{s}^\sigma.
  \]
SBNs Parameters $\phi$. With $\tau^j, q^j \overset{iid}{\sim} Q_{\phi,\psi}(\tau, q)$

- **VIMCO.** [Minh and Rezende, ICML 2016]
  \[
  \nabla_\phi L^K(\phi, \psi) \simeq \sum_{j=1}^K \left( \hat{L}_j^K(\phi, \psi) - \tilde{w}^j \right) \nabla_\phi \log Q_{\phi}(\tau^j).
  \]

- **RWS.** [Bornschein and Bengio, ICLR 2015]
  \[
  \nabla_\phi L^K(\phi, \psi) \simeq \sum_{j=1}^K \tilde{w}^j \nabla_\phi \log Q_{\phi}(\tau^j).
  \]

Branch Length Parameters $\psi$. $g_\psi(\epsilon|\tau) = \exp(\mu_{\psi,\tau} + \sigma_{\psi,\tau} \odot \epsilon)$.

- **Reparameterization Trick.** Let $f_{\phi,\psi}(\tau, q) = \frac{p(Y|\tau, q)p(\tau, q)}{Q_{\phi}(\tau)Q_{\psi}(q|\tau)}$.
  \[
  \nabla_\psi L^K(\phi, \psi) \simeq \sum_{j=1}^K \tilde{w}^j \nabla_\psi \log f_{\phi,\psi}(\tau^j, g_\psi(\epsilon^j|\tau^j))
  \]

where $\tau^j \overset{iid}{\sim} Q_{\phi}(\tau), \epsilon^j \overset{iid}{\sim} \mathcal{N}(0, I)$. 
Performance on Real Data

[Figure showing performance metrics such as KL divergence over iterations for different methods, including VIMCO, RWS, and MCMC, with and without PSP.]

[Text: More samples ⇒ better exploration ⇒ better approximation. More flexible branch length distributions across tree topologies (PSP) ease training and improve approximation. Outperform MCMC via much more efficient tree space exploration and branch length updates.]

[Zhang and Matsen, ICLR 2019]
Performance on Real Data

More samples ⇒ better exploration ⇒ better approximation

[Zhang and Matsen, ICLR 2019]
Performance on Real Data

[Zhang and Matsen, ICLR 2019]

More samples ⇒ better exploration ⇒ better approximation

More flexible branch length distributions across tree topologies (PSP) ease training and improve approximation
Performance on Real Data

More samples ⇒ better exploration ⇒ better approximation

More flexible branch length distributions across tree topologies (PSP) ease training and improve approximation

Outperform MCMC via much more efficient tree space exploration and branch length updates

[Zhang and Matsen, ICLR 2019]
## Performance on Real Data

<table>
<thead>
<tr>
<th>Data set</th>
<th>Marginal Likelihood (NATs)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VIMCO(10)</td>
<td>VIMCO(20)</td>
</tr>
<tr>
<td>DS1</td>
<td>-7108.43(0.26)</td>
<td>-7108.35(0.21)</td>
</tr>
<tr>
<td>DS2</td>
<td>-26367.70(0.12)</td>
<td>-26367.71(0.09)</td>
</tr>
<tr>
<td>DS3</td>
<td>-33735.08(0.11)</td>
<td>-33735.11(0.11)</td>
</tr>
<tr>
<td>DS4</td>
<td>-13329.90(0.31)</td>
<td>-13329.98(0.20)</td>
</tr>
<tr>
<td>DS5</td>
<td>-8214.36(0.67)</td>
<td>-8214.74(0.38)</td>
</tr>
<tr>
<td>DS6</td>
<td>-6723.75(0.68)</td>
<td>-6723.71(0.65)</td>
</tr>
<tr>
<td>DS7</td>
<td>-37332.03(0.43)</td>
<td>-37331.90(0.49)</td>
</tr>
<tr>
<td>DS8</td>
<td>-8653.34(0.55)</td>
<td>-8651.54(0.80)</td>
</tr>
</tbody>
</table>

[Zhang and Matsen, ICLR 2019]
## Performance on Real Data

### Data set Marginal Likelihood (NATs)

<table>
<thead>
<tr>
<th>Data set</th>
<th>VIMCO(10)</th>
<th>VIMCO(20)</th>
<th>VIMCO(10)+PSP</th>
<th>VIMCO(20)+PSP</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1</td>
<td>-7108.43(0.26)</td>
<td>-7108.35(0.21)</td>
<td>-7108.41(0.16)</td>
<td><strong>-7108.42(0.10)</strong></td>
<td>-7108.42(0.18)</td>
</tr>
<tr>
<td>DS2</td>
<td>-26367.70(0.12)</td>
<td>-26367.71(0.09)</td>
<td><strong>-26367.72(0.08)</strong></td>
<td>-26367.70(0.10)</td>
<td>-26367.57(0.48)</td>
</tr>
<tr>
<td>DS3</td>
<td>-33735.08(0.11)</td>
<td>-33735.11(0.11)</td>
<td><strong>-33735.10(0.09)</strong></td>
<td>-33735.07(0.11)</td>
<td>-33735.44(0.50)</td>
</tr>
<tr>
<td>DS4</td>
<td>-13329.90(0.31)</td>
<td>-13329.98(0.20)</td>
<td><strong>-13329.94(0.18)</strong></td>
<td>-13329.93(0.22)</td>
<td>-13330.06(0.54)</td>
</tr>
<tr>
<td>DS5</td>
<td>-8214.36(0.67)</td>
<td>-8214.74(0.38)</td>
<td>-8214.61(0.38)</td>
<td>-8214.55(0.43)</td>
<td><strong>-8214.51(0.28)</strong></td>
</tr>
<tr>
<td>DS6</td>
<td>-6723.75(0.68)</td>
<td>-6723.71(0.65)</td>
<td>-6724.09(0.55)</td>
<td><strong>-6724.34(0.45)</strong></td>
<td>-6724.07(0.86)</td>
</tr>
<tr>
<td>DS7</td>
<td>-37332.03(0.43)</td>
<td>-37331.90(0.49)</td>
<td>-37331.90(0.32)</td>
<td><strong>-37332.03(0.23)</strong></td>
<td>-37332.76(2.42)</td>
</tr>
<tr>
<td>DS8</td>
<td>-8653.34(0.55)</td>
<td>-8651.54(0.80)</td>
<td><strong>-8650.63(0.42)</strong></td>
<td>-8650.55(0.46)</td>
<td>-8649.88(1.75)</td>
</tr>
</tbody>
</table>

[Zhang and Matsen, ICLR 2019]

Competitive to state-of-the-art (stepping-stone), dramatically reducing cost at test time: **VBPI(1000)** vs **SS(100,000)**
Performance on Real Data

<table>
<thead>
<tr>
<th>Data set</th>
<th>Marginal Likelihood (NATs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VIMCO(10)</td>
</tr>
<tr>
<td>DS1</td>
<td>-7108.43(0.26)</td>
</tr>
<tr>
<td>DS2</td>
<td>-26367.70(0.12)</td>
</tr>
<tr>
<td>DS3</td>
<td>-33735.08(0.11)</td>
</tr>
<tr>
<td>DS4</td>
<td>-13329.90(0.31)</td>
</tr>
<tr>
<td>DS5</td>
<td>-8214.36(0.67)</td>
</tr>
<tr>
<td>DS6</td>
<td>-6723.75(0.68)</td>
</tr>
<tr>
<td>DS7</td>
<td>-37332.03(0.43)</td>
</tr>
<tr>
<td>DS8</td>
<td>-8653.34(0.55)</td>
</tr>
</tbody>
</table>

[Zhang and Matsen, ICLR 2019]

Competitive to state-of-the-art (stepping-stone), dramatically reducing cost at test time: **VBPI(1000) vs SS(100,000)**

**PSP** alleviates the demand for large samples, reducing computation while maintaining approximation accuracy
We introduced **VBPI**, a general variational framework for Bayesian phylogenetic inference.

**VBPI** allows efficient learning on both tree topology and branch lengths, providing competitive performance to MCMC while requiring much less computation.

Can be used for further statistical analysis (e.g., marginal likelihood estimation) via importance sampling.


Thank you!