

Adapting cluster graphs for inference of continuous trait evolution on phylogenetic networks

Benjamin Teo and Cécile Ané

Abstract—Dynamic programming approaches have long been applied to fit models of univariate and multivariate trait evolution on phylogenetic trees for discrete and continuous traits, and more recently adapted to phylogenetic networks with reticulation. We previously showed that various trait evolution models on a network can be readily cast as probabilistic graphical models, so that likelihood-based estimation can proceed efficiently via belief propagation on an associated clique tree. Even so, exact likelihood inference can grow computationally prohibitive for large complex networks. Loopy belief propagation can similarly be applied to these settings, using non-tree cluster graphs to optimize a factored energy approximation to the log-likelihood, and may provide a more practical trade-off between estimation accuracy and runtime. However, the influence of cluster graph structure on this trade-off is not precisely understood. We conduct a simulation study using the Julia package `PhyloGaussianBeliefProp` to investigate how varying maximum cluster size affects this trade-off for Gaussian trait evolution models on networks. We discuss recommended choices for maximum cluster size, and prove the equivalence of likelihood-based and factored-energy-based parameter estimates for the homogeneous Brownian motion model.

Index Terms—continuous trait, linear Gaussian, admixture graph, belief propagation, cluster graph, approximate inference

I. MOTIVATION

LOOPY belief propagation (LBP) provides a means of parameter estimation by approximate inference [1, Sec. 20.5.1]. LBP extends belief propagation (BP) in an appealing way that retains BP’s simple formulation and intuitive message-passing rules [1, Sec. 11.3.2]. Although LBP faces non-convergence issues and approximation error, especially when the “simplest” cluster graphs are used [2], [3], its performance can seemingly be improved by constructing alternative cluster graphs [4], [5]. However, such alternatives have received less attention, in part because of limited automatic construction procedures [6] and because the trade-offs between different cluster graphs are difficult to characterize precisely.

We take up the challenge of systematically investigating how cluster graph structure affects the performance of LBP for fitting Gaussian trait models on phylogenetic networks [5], [7]. For the scope of this study, we focus on maximum likelihood estimation for the Brownian motion model [8, Ch. 3.5] on three networks of varying topological complexity, taken from the admixture graph and ancestral recombination graph literature [9], [10]. We use the join-graph structuring algorithm to

generate cluster graphs of varying maximum cluster size k [11], and address the problem of choosing k to strike a good balance between computational cost and estimation accuracy. We run LBP with a fixed strategy for scheduling messages and initializing cluster graph beliefs (Sec. III-B), though other choices could be explored to improve convergence.

Our objective is to assess the potential of LBP as a tool for more scalable, approximate inference of Gaussian trait models on evolutionary networks, which can be valuable when parameter estimation requires iterative numerical optimization [12], [13]. We hope that our findings initiate or encourage the design of new cluster graphs, message schedules, and belief initializations that better suit the phylogenetic context.

II. BACKGROUND

This section provides a brief overview of the key concepts involved for using LBP to fit Gaussian trait models on phylogenetic networks. For a more detailed exposition, see [5].

A. Trait evolution on phylogenetic networks

A phylogeny is a graph that describes the shared genealogy for a set of entities, typically but not necessarily biological (e.g., species [14] or languages [15]). Nodes represent these entities and their common ancestors, while edges represent lineages (Fig. 1a). Rooting the phylogeny by specifying an oldest node fixes the direction of time and evolution along each edge. As time cannot go backwards, a rooted phylogeny cannot have any directed cycle.

Definition 2.1 (Rooted phylogenetic network [5, Ex. 2]). A *rooted phylogenetic network* is a connected, directed acyclic graph (DAG) $N = (V, E)$ with a single *root* (a node with no parents) and taxon-labeled leaves. Nodes with at most one parent are called *tree nodes*, and their in-edges are called *tree edges*. Nodes with multiple parents represent populations with mixed ancestry. They are called *hybrid nodes*, and their in-edges are called *hybrid edges*. Each hybrid edge $e = (u, h)$ is assigned an *inheritance weight* $\gamma(e) > 0$ that represents the proportion of the genome in hybrid h that was inherited from its parent u . The inheritance weights for each hybrid node must sum to 1.

B. Teo is with the Department of Statistics, University of Wisconsin-Madison, Madison, WI, USA

C. Ané is with the Departments of Statistics and Botany, University of Wisconsin-Madison, Madison, WI, USA.

Trait evolution is widely modeled over a rooted phylogeny as a forwards-in-time stochastic process that progresses along the graph edges, culminating in the observed traits at the leaves (Fig. 1b). Evolutionary changes along separate edges are typically assumed independent given their start states [16]. If not, such as to model interacting populations or the variability of gene histories [17], [18], trait evolution can be modeled over a supergraph of the phylogeny [5]. Either way, trait evolution models can be described by a graphical model with conditional distributions local to each edge, which are induced by the stochastic process [5], [19]–[21].

B. Gaussian trait models

Gaussian trait models remain a workhorse in the study of continuous trait evolution [22], and are often characterized by linear Gaussian models along each edge [7].

Definition 2.2 (Linear Gaussian model [5, Sec. 3(a)]). A *linear Gaussian model* on a directed tree edge (u, v) assumes that the p -dimensional trait X_v at node v has the following conditional distribution, given the trait X_u of its single parent u :

$$X_v \mid X_u \sim \mathcal{N}(\mathbf{q}_v X_u + \omega_v, \mathbf{V}_v) \quad (1)$$

where the $p \times p$ actualization matrix \mathbf{q}_v , length- p trend vector ω_v , and $p \times p$ covariance matrix \mathbf{V}_v do not depend on X_u .

If v has multiple parents $\text{pa}(v) = \{u_1, \dots, u_m\}$, whose stacked traits $\text{vec}([X_{u_1} \dots X_{u_m}])$ we denote by $X_{\text{pa}(v)}$, then the *node family* $\{v\} \cup \text{pa}(v)$ has a linear Gaussian model if

$$X_v \mid X_{\text{pa}(v)} \sim \mathcal{N}(\mathbf{q}_v X_{\text{pa}(v)} + \omega_v, \mathbf{V}_v) \quad (2)$$

with \mathbf{q}_v , ω_v , and \mathbf{V}_v independent of $X_{\text{pa}(v)}$, where \mathbf{q}_v is now of size $p \times (mp)$. A DAG for which every node family has a linear Gaussian model is known as a *linear Gaussian network*.

If v has multiple parent edges $e_k = (u_k, v)$, it is reasonable to assume that X_v is a function of $(X_{e_k})_{k=1, \dots, m}$, where X_{e_k} denotes the trait at the end of edge e_k . For example, a weighted average (Fig. 1b) is a reasonable approximation for continuous traits that are influenced by many genes [23].

Definition 2.3 (Weighted-average merging rule [5, Eq. 3.3]). For a hybrid node v with parent edges e_1, \dots, e_m and inheritance weights $\gamma(e_1), \dots, \gamma(e_m)$, the *weighted-average model* assumes that

$$X_v = \sum_{e_k \text{ parent of } v} \gamma(e_k) X_{e_k}. \quad (3)$$

If each e_k has a linear Gaussian model (1), then the weighted-average model (3) implies a linear Gaussian model (2) for $\{v\} \cup \text{pa}(v)$.

C. Belief propagation on cluster graphs

Belief propagation is a technique for efficiently computing the marginals of a probability density expressed as a graphical model. It can be leveraged to compute the likelihood for parameter optimization. We focus on directed graphical models.

Definition 2.4 (Directed graphical model [5, Sec. 4(a)]). Let p_θ be the joint probability density for a set of random vectors

$\{X_v, v \in V\}$, with parameters $\theta \in \Theta$. A *directed graphical model* for p_θ consists of a DAG $G = (V, E)$ and a set $\Phi := \{\phi_v, v \in V\}$ of non-negative functions such that:

- 1) $p_\theta = \prod_{\phi_v \in \Phi} \phi_v$
- 2) Each ϕ_v is proportional to the conditional density for v 's node family in G : $\phi_v \propto p_\theta(X_v \mid X_u, u \in \text{pa}(v))$.

We say that p_θ *factorizes* over G and refer to the ϕ_v as *factors*. The set of X_w that ϕ_v is defined over is referred to as its *scope*, denoted $\text{scope}(\phi_v)$. Finally, $\text{scope}(\Phi) := \cup_{\phi_v \in \Phi} \text{scope}(\phi_v)$.

Gaussian trait models on a phylogenetic network can typically be cast as a graphical model with $G = N$ and Φ consisting of linear Gaussian factors. Belief propagation then proceeds by passing messages on a cluster graph constructed from the graphical model.

Definition 2.5 (Cluster graph [5, Def. 1]). A *cluster graph* for a directed graphical model (G, Φ) is an undirected graph $\mathcal{U} = (\mathcal{V}, \mathcal{E})$, whose vertices $\mathcal{C}_i \in \mathcal{V}$, $i \in [|\mathcal{V}|]$ are subsets of $\text{scope}(\Phi)$ and are called *clusters*. Additionally, \mathcal{U} must satisfy:

- 1) Each factor ϕ_v can be assigned to a cluster $\mathcal{C}_{\alpha(\phi_v)}$ that contains its scope, i.e., $\text{scope}(\phi_v) \subseteq \mathcal{C}_{\alpha(\phi_v)}$; α denotes a suitable map from factors to cluster indices.
- 2) Each edge $\{\mathcal{C}_i, \mathcal{C}_j\} \in \mathcal{E}$ is labeled with a non-empty *separating set* (or *sepset*) $\mathcal{S}_{i,j} \subseteq \mathcal{C}_i \cap \mathcal{C}_j$. Thus, only clusters with common elements can be joined.
- 3) For each $X_v \in \text{scope}(\Phi)$, $(\mathcal{V}_{X_v}, \mathcal{E}_{X_v})$ is a subtree of \mathcal{U} , where $\mathcal{V}_{X_v} \subseteq \mathcal{V}$ and $\mathcal{E}_{X_v} \subseteq \mathcal{E}$ are the sets of clusters and edges whose labels contain X_v .

If \mathcal{U} is a tree, then we call \mathcal{U} a *clique tree* and refer to its clusters as *cliques* (Fig. 1c). For a clique tree, properties 2 and 3 imply that $\mathcal{S}_{i,j} = \mathcal{C}_i \cap \mathcal{C}_j$. If \mathcal{U} contains one or more cycles, then we call \mathcal{U} a *loopy cluster graph* (Fig. 1d).

\mathcal{U} acts as a data structure for ordering the sequence of operations and holding intermediate computations used to build up various marginals of p_θ . When an edge is traversed, intermediate computations aggregated at the source cluster are marginalized, and the result is passed to the target cluster for downstream computations. The intermediate computations tracked at each cluster and edge are scalar functions that are defined over their variables and called *beliefs*, denoted $\beta_i(\mathcal{C}_i)$ or $\mu_{i,j}(\mathcal{S}_{i,j})$. Cluster beliefs are marginalized to produce *messages*, which update neighbor cluster beliefs by multiplication, while edge beliefs record the most recent messages passed. Sending a message across an edge is called a *belief update* (Alg. 1), while circulating messages throughout \mathcal{U} is known as *belief propagation*, or *loopy belief propagation* to emphasize the use of a loopy cluster graph. Beliefs are initialized as the product of factors assigned to the corresponding cluster (Def. 2.5) and are instantiated at the observed data. That is, if $X_v = x_v$ is observed, then this value is fixed in all beliefs defined over X_v , and X_v is removed from their scopes. Variables determined by model parameters (e.g., the root state X_ρ in a Brownian motion trait model) are similarly removed from beliefs' scopes.

If Φ consists of linear Gaussian factors, then beliefs can be compactly parametrized and belief updates can be expressed as updates to these parameters [5, Sec. 4(c)]. This is an instance

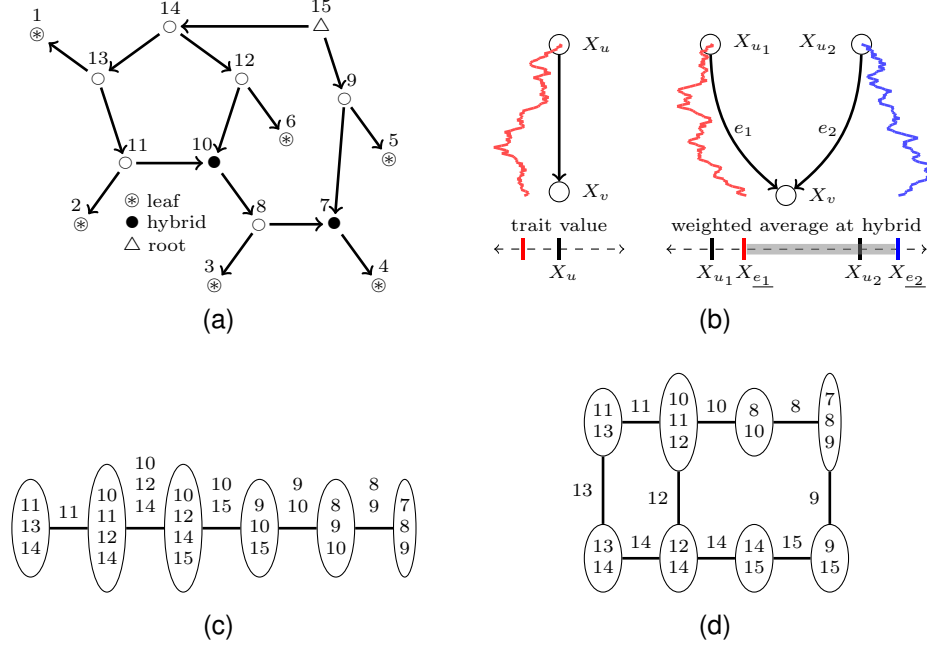


Fig. 1. (a) Rooted phylogenetic network. (b) Example realization of a univariate trait evolution, along one edge uv (left) or both parent edges of a hybrid node v (right). The gray region shows the range of possible trait values at the hybrid node under a weighted-average model. (c) Clique tree for any graphical model on the network in (a), with numbers referring to nodes in (a). Its maximum cluster size is $k^* = 4$. (d) Loopy cluster graph for the network in (a), with maximum cluster size $k = 3$.

Algorithm 1 Belief update from \mathcal{C}_i to \mathcal{C}_j [5, Alg. 1]

- | | |
|--|--|
| 1: Compute message $(\tilde{\mu}_{i \rightarrow j})$
from source belief (β_i) : | $\tilde{\mu}_{i \rightarrow j} \leftarrow \int \beta_i d(\mathcal{C}_i \setminus \mathcal{S}_{i,j})$ |
| 2: Update target (β_j) and
edge $(\mu_{i,j})$ beliefs: | $\beta_j^{\text{new}} \leftarrow (\tilde{\mu}_{i \rightarrow j} / \mu_{i,j}) \beta_j$
$\mu_{i,j}^{\text{new}} \leftarrow \tilde{\mu}_{i \rightarrow j}$ |
-

of *Gaussian belief propagation*, where the graphical model (not necessarily directed) represents a Gaussian distribution.

D. Calibration and the factored energy

\mathcal{U} is said to be *calibrated* if all its beliefs are *marginally consistent*, or equivalently, if for any pair of neighbors \mathcal{C}_i and \mathcal{C}_j : $\int \beta_i d(\mathcal{C}_i \setminus \mathcal{S}_{i,j}) \propto \mu_{i,j} \propto \int \beta_j d(\mathcal{C}_j \setminus \mathcal{S}_{i,j})$. If so, belief updates do not change the beliefs of a calibrated cluster graph. For a clique tree, calibration is guaranteed in two traversals: a postorder traversal towards any designated root clique, followed by a preorder traversal outwards from the root clique. For a loopy cluster graph, calibration requires multiple traversals, and may not be guaranteed.

Since neighbor clusters must exchange messages for their beliefs to attain marginal consistency, a sound *message schedule* (i.e., a sequence of messages to be passed) should traverse each edge, in both directions. Since multiple exchanges between neighbor clusters may be needed, a *valid schedule* [24] should additionally repeat each message infinitely often. For example, cycling through a sound finite schedule satisfies this requirement.

Beliefs can be viewed as unnormalized estimates of the conditional distribution of cluster or edge variables given the

observed data Y : $\beta_i \propto \hat{p}_\theta(\mathcal{C}_i | Y)$ and $\mu_{i,j} \propto \hat{p}_\theta(\mathcal{S}_{i,j} | Y)$. Beliefs are exact ($\beta_i \propto p_\theta(\mathcal{C}_i | Y)$ and $\mu_{i,j} \propto p_\theta(\mathcal{S}_{i,j} | Y)$) on calibrated clique trees, but approximate on calibrated loopy cluster graphs. Upon calibration, the factored energy can be computed to approximate the log-likelihood.

Definition 2.6 (Factored energy [5, Eq. 5.1]). The *factored energy* of a cluster graph $\mathcal{U} = (\mathcal{V}, \mathcal{E})$ with calibrated beliefs $q^* = \{\beta_i^*, \mu_{i,j}^* | \mathcal{C}_i \in \mathcal{V}, \{\mathcal{C}_i, \mathcal{C}_j\} \in \mathcal{E}\}$ is defined as

$$\tilde{F}_{\mathcal{U}}(q^*) := \sum_{\mathcal{C}_i \in \mathcal{V}} E_{\beta_i^*}(\log \psi_i | Y) + \sum_{\mathcal{C}_i \in \mathcal{V}} H(\beta_i^*) - \sum_{\{\mathcal{C}_i, \mathcal{C}_j\} \in \mathcal{E}} H(\mu_{i,j}^*) \quad (4)$$

where $\psi_i = \prod_{\phi_v \in \Phi, \alpha(\phi_v)=i} \phi_v$ combines the factors assigned to cluster i , $(\cdot) | Y$ denotes evaluation at the observed data, $E_{\beta_i^*}(\cdot)$ is the expectation with respect to β_i^* , and $H(\cdot)$ is the entropy.

$\tilde{F}_{\mathcal{U}}$ resembles the evidence lower bound (ELBO) to the log-likelihood $LL = \log(\int p_\theta | Y dX)$, where X denotes the unobserved variables in $\text{scope}(\Phi)$ [1], [25]. If \mathcal{U} is a clique tree, then $\tilde{F}_{\mathcal{U}} = LL$ exactly¹.

In the factored energy (4), each term is an integral of lower dimension and cheaper to compute than the likelihood, $\int p_\theta | Y dX$. The computational cost of each such integral is parametrized by the size of the associated cluster $|\mathcal{C}|$ or sepset $|\mathcal{S}|$. Thus, the cost of belief propagation and of evaluating the factored energy are parametrized by the maximum cluster size $k = \max_{\mathcal{C} \in \mathcal{V}} |\mathcal{C}|$. A larger k is associated with higher cost. For

¹For a calibrated clique tree, the likelihood can also be obtained by fully integrating any of the beliefs, e.g., $\int \beta_i^* d\mathcal{C}_i = \int \mu_{i,j}^* d\mathcal{S}_{i,j} = \int p_\theta | Y dX$.

a given graphical model, k^* denotes the minimum k among all possible clique trees for this model. While the exact LL and calibrated beliefs can be computed with a clique tree (with $k \geq k^*$ necessarily), a loopy cluster graph permits $k < k^*$ with smaller clusters and hence cheaper messages.

III. METHODS

This section describes simulations, using our Julia package `PhyloGaussianBeliefProp`, that investigate the use of LBP on cluster graphs to fit a Brownian motion (BM) trait model on a network. We first compare the runtime and accuracy of using the factored energy to approximate the log-likelihood, for different maximum cluster sizes k . We then assess the accuracy of parameter estimation by maximizing the factored energy, using k chosen from the previous step.

A. Maximum factored energy estimation

The error for using the factored energy (FE) of a calibrated cluster graph to approximate the log-likelihood (LL) intuitively improves with larger clusters and sepsets, and a more treelike topology, but is otherwise not well understood as a function of cluster graph structure. Whether or not maximum factored energy (MFE) estimation is a reasonable proxy for maximum likelihood (ML) estimation depends on the size of the error across parameter space and more importantly, the extent to which the shapes of the FE and LL surfaces are similar. Further, whether or not MFE estimation is practical depends on the extent to which cheaper messages reduce the computational effort needed to obtain FE values. Though individual messages can be cheaper on a loopy cluster graph than on a clique tree, calibration requires more traversals.

We use an illustrative case to better understand if and how MFE estimation can be applied. We compared MFE and ML estimation for fitting a BM model of p -dimensional trait evolution with variance rate $\Sigma \in \mathbb{R}^{p \times p}$ on a phylogenetic network $N = (V, E)$ with n tips and fixed root state $X_\rho = \mu \in \mathbb{R}^p$. The parameters to be estimated are Σ and μ . We focus on this model because it has closed-form ML estimates (5).

Let $Y_i \in \mathbb{R}^p$ be the unobserved trait vector at leaf i , $\mathbf{Y} = [Y_1 \cdots Y_n] \in \mathbb{R}^{p \times n}$, and $Y = \text{vec}(\mathbf{Y}) \in \mathbb{R}^{np}$. $Y \sim \mathcal{N}(1_n \otimes \mu, \mathbf{P}_y \otimes \Sigma)$, where $\mathbf{P}_y \in \mathbb{R}^{n \times n}$ is the shared-path matrix for the leaves [23, Eq. 1], and 1_n is the n -vector of 1s. Alternatively, $\mathbf{Y} \sim \mathcal{MN}_{p,n}(\mu 1_n^\top, \Sigma, \mathbf{P}_y)$ in the matrix-normal formulation. The closed-form expressions for the ML estimates and LL of this model (given observed data $\mathbf{Y} = \mathbf{Y}$) [26], [27] obviate the use of BP and loopy BP, to the extent that \mathbf{P}_y can be accurately inverted:

$$\begin{aligned} \hat{\mu}_{\text{ML}} &= \mathbf{Y} \mathbf{P}_y^{-1} 1_n / \|1_n\|_{\mathbf{P}_y^{-1}}^2 \\ \hat{\Sigma}_{\text{ML}} &= \|\mathbf{Y}^\top - 1_n \hat{\mu}_{\text{ML}}^\top\|_{\mathbf{P}_y^{-1}}^2 / n \\ \text{LL}(\hat{\mu}_{\text{ML}}, \hat{\Sigma}_{\text{ML}}) &= -\frac{np}{2} (1 + \log 2\pi) - \frac{p}{2} \log |\mathbf{P}_y| \\ &\quad - \frac{n}{2} \log |\hat{\Sigma}_{\text{ML}}| \end{aligned} \quad (5)$$

where $\|\cdot\|_{\mathbf{P}_y^{-1}}^2$ denotes the function $(\cdot)^\top \mathbf{P}_y^{-1} (\cdot)$ when applied to an input matrix of conformable dimensions. These equations allow us to conveniently assess the accuracy of the MFE and the corresponding parameter estimates $\hat{\mu}_{\text{MFE}}$, $\hat{\Sigma}_{\text{MFE}}$.

B. Spanning trees schedule and regularizing beliefs

For LBP, we based our message schedule on a small, though not necessarily minimal, set of spanning trees of the cluster graph, that together cover all its edges and hence all possible messages [5, Sec. 5(a)]. This set is chosen by iteratively applying Kruskal's algorithm for finding a minimum-weight spanning tree [28], each time incrementing the weights of edges that are selected for the current spanning tree, and stopping when all edges of the cluster graph have been used. The schedule cycles through the spanning trees, and at each step traverses the current spanning tree in postorder, then pre-order. We refer to each pass through this set of spanning trees as an *iteration* of LBP. This is a specific implementation of the tree-based reparametrization framework proposed in [24], except that each "tree update" does not necessarily calibrate the current spanning tree (e.g., because some messages are defined over a smaller scope than the intersection of the source and target clusters, i.e., $\mathcal{S}_{i,j} \subset \mathcal{C}_i \cap \mathcal{C}_j$, or because some messages are infinite).

A known issue in LBP is that infinite messages may arise regardless of the chosen schedule. In step 1 of Algorithm 1, message $\tilde{\mu}_{i \rightarrow j}$ is infinite if integrating out the required variables from the source belief evaluates to infinity, thus preventing update and calibration of the target belief [5, Sec. 7(a)]. This issue is specific to beliefs over continuous variables, which can be unnormalizable², and is more prominent for directed graphical models since the factors are conditional densities. Cluster beliefs initialized by multiplying conditional densities and without any instantiation at the observed data may be unnormalizable. Infinite messages can sometimes be avoided by *regularizing* the initial beliefs while ensuring that the cluster graph still represents the same probability density $p_\theta|Y$ [5, Sec. SM-E]. For LBP, we applied the regularization strategy from [5, Alg. R4] so that all cluster beliefs were normalizable before applying our spanning trees schedule.

C. Simulations

1) *Network preprocessing*: We used three networks of varying topological complexity, in terms of network level ℓ [29] and moralized treewidth $k^* - 1$ (Sec. II-D)³. We refer to these as the Sikora [30] ($n = 13$, $\ell = 6$, $k^* = 5$), Lipson [31] ($n = 12$, $\ell = 12$, $k^* = 7$) and Müller [32] ($n = 40$, $\ell = 358$, $k^* = 54$) networks. For the Sikora and Lipson networks, we used the admixture weights as inheritance weights, and the drift weights as edge lengths. Any degree-2 nodes were suppressed, and any length-0 edges were reassigned the minimum positive edge length. We avoided length-0 edges because they introduce deterministic factors that require more sophisticated belief updates [5, Sec. SM-F]. Such cases will be handled in a future version of `PhyloGaussianBeliefProp`. For the Müller network, we used the edge lengths provided (which represent calendar time) and estimated the inheritance weights for each hybrid edge from the recombination breakpoints

²The normalizability of a cluster's belief is sufficient but not necessary for messages computed from that cluster to be finite.

³ k^* was not computed exactly, but approximated by the maximum cluster size of a clique tree constructed using the min-fill heuristic [1, Sec. 9.4.3].

provided (see `muller2022_nexus2newick.jl` in https://github.com/bstjkj/graphicalmodels_for_phylogenetics_code).

2) *Simulating trait data*: For each network, we used `PhyloNetworks` v0.16.4 [33] to simulate 100 datasets under a univariate ($p = 1$) and multivariate ($p = 4$) Brownian motion trait model with root mean $\mu_0 = 0$, assuming a weighted-average model at the hybrid nodes (Def. 2.3). The following variance rates were used: $\sigma_0^2 = 1$ for univariate data, and for multivariate data

$$\Sigma_0 = \begin{bmatrix} 0.8 & -0.71 & -0.8 & 0.49 \\ \cdot & 0.8 & 0.81 & -0.41 \\ \cdot & \cdot & 1.1 & -0.4 \\ \cdot & \cdot & \cdot & 0.5 \end{bmatrix}.$$

The choice of Σ_0 was motivated by real data from [34] on four leaf traits: photosynthetic rate, leaf lifespan, leaf mass per area, and leaf nitrogen content (Appendix A). The covariance matrix implied by the path diagram in [34, Fig. 2 (top)], estimated by phylogenetic structural equation models, is $\Sigma_0 \cdot 10^{-3}$. The corresponding correlation matrix between the 4 traits is approximately

$$\rho_0 = \begin{bmatrix} 1 & -0.887 & -0.853 & 0.775 \\ \cdot & 1 & 0.863 & -0.648 \\ \cdot & \cdot & 1 & -0.539 \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}.$$

3) Accuracy and runtime versus maximum cluster size k :

For each network and dataset, we ran LBP on cluster graphs generated using join-graph structuring for different values of k , ranging from 3 to k^* . The smallest value was $k = 3$ because all networks here are *bicombining*, where each hybrid node has two parents. For the Sikora and Lipson networks, we varied k from 3 to 5 and 3 to 7 by increments of 1. For the Müller network, we varied k from 3 to 20 by increments of 1, from 25 to 50 by increments of 5, then from 51 to 54 by increments of 1. For each task (e.g., LBP for a given network, dataset, and k) we evaluated the FE at the true parameter values μ_0, Σ_0 after calibration was detected or after 50 iterations had passed, whichever came first, then recorded the relative deviation $|\frac{FE-LL}{LL}|$ and the combined runtime for passing messages and computing the FE. Each task was run on a separate physical core, using a single thread, of an Intel Xeon E5-2680 v3 processor (30M Cache, 2.50GHz).

The computed FE values only approximate the true FE values since calibration was detected within a specified numerical tolerance, or may not even have been detected; but (4) can be well-defined before calibration. We excluded the runtimes for building the cluster graph, initializing and regularizing beliefs, and generating a schedule, though these were negligible in comparison. Since our software is written in Julia [35], a Just-In-Time compiled language, compilation latency affects the initial run, but not subsequent runs on similar inputs. Thus, each task was repeated to record the second runtime.

We plotted relative deviation and runtime against k to choose a k that strikes a good balance between accuracy and runtime (Figs. 2–3). This choice was made separately for each combination of network and trait dimension.

4) *Parameter estimation with a chosen k* : Using k chosen in the previous step, we compared the accuracy of MFE and ML estimation for each network and trait dimension. The ML estimates were computed exactly using (5), while the MFE estimates were obtained by numerically optimizing an approximate FE (see above) using the L-BFGS algorithm (inverse Hessian approximated from past 10 steps) [36] with Hager-Zhang line search [37] and with the FE gradient approximated by finite differences. We started the optimization at $\hat{\mu}_{\text{init}} = \frac{1}{n} \sum_{i=1}^n Y_i$ and $\hat{\Sigma}_{\text{init}} = \frac{1}{nh} \sum_{i=1}^n (Y_i - \hat{\mu}_{\text{init}})(Y_i - \hat{\mu}_{\text{init}})^\top$, where $h = \text{median}_{1 \leq i \leq n} (\mathbf{P}_y)_{ii}$ is a measure of the network’s height and can be obtained in linear time in the number of nodes [23, Prop. 2]. These are the ML estimates under a BM model for an ultrametric star tree, for which the leaf states have equal variances but no phylogenetic correlation. As above, the FE was computed after calibration was detected or after 50 iterations had passed. The maximum number of optimization steps was set to 50, with early termination triggered if, after any step, the FE increased by less than 0.01%, or if all coordinates of the approximated gradient had magnitude less than 10^{-8} .

We compared the root-mean-square (RMS) of the difference between the MFE and ML estimates of μ_0, σ_0^2 (or Σ_0). Each task (i.e., FE optimization for a given network and dataset) was run on a separate physical core, using a single thread, of an Intel Xeon E5-2687 v2 processor (30M Cache, 2.70GHz).

IV. RESULTS

A. Simulations

1) Accuracy and runtime versus maximum cluster size k :

For the Sikora and Lipson networks, which are moderately complex ($k^* = 5, 7$ respectively), accuracy and runtime generally improved as k increased (Fig. 2). The relative deviation $|\frac{FE-LL}{LL}|$ dropped sharply from slightly under 10^{-3} for $k < k^*$ to under 10^{-12} at $k = k^*$. Thus, we picked $k = k^*$ (using a clique tree) to minimize error and runtime.

For the Müller network, which is highly complex ($k^* = 54$), accuracy and runtime generally improved with k in the univariate case, but varied less trivially with k in the multivariate case (Fig. 3). Relative deviation varied similarly in the univariate ($p = 1$) and multivariate ($p = 4$) cases, greatly exceeding 1 (100%) for $k < 10$, dropping below 0.1 (10%) for $k \geq 11$, and below 0.01 (1%) when $k \geq 25$ for $p = 1$, or when $k \geq 20$ for $p = 4$. It decreased rapidly from $k = 3$ to 10, and then more gradually from $k = 11$ to 53, with a sharp drop (on the log-scale) at $k = k^* = 54$. In contrast, runtime differed markedly between $p = 1$ and $p = 4$. For $p = 1$, runtime increased slightly from $k = 3$ to 35 then decreased after $k \geq 35$, therefore we picked $k = k^*$ as the best value. For $p = 4$, runtime increased substantially between $k = 16$ and 18, so we picked $k = 11 < k^*$ for $p = 4$ to achieve both a shorter runtime and good accuracy.

Calibration was consistently detected within 50 iterations across all datasets for the Sikora and Lipson networks for all k , and for the Müller network only for $k \geq 35$, with the number of iterations required generally decreasing in k (Table I). We expect runtime to decrease with k in these

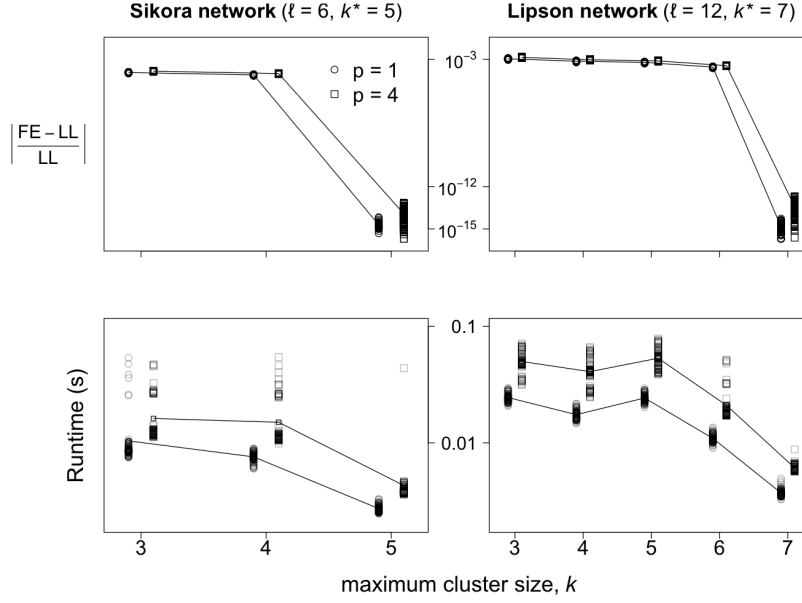


Fig. 2. Relative deviation and runtime (on the log-scale, vertical axis) versus maximum cluster size k for the Sikora and Lipson networks, in the univariate and multivariate cases (across 100 datasets each). Trajectories of the mean values as k increases are traced. Top: relative deviation $|(FE - LL)/LL|$. Bottom: post-compilation runtimes for passing messages and evaluating the FE (upon detection of calibration or after the first 50 iterations). For the Lipson network in the multivariate case, the mean number of iterations before calibration was detected (± 1 standard deviation) is 7.87 ± 0.32 iterations when $k = 4$ versus 11.22 ± 0.44 iterations when $k = 5$. This explains the rise in runtime from $k = 4$ to $k = 5$.

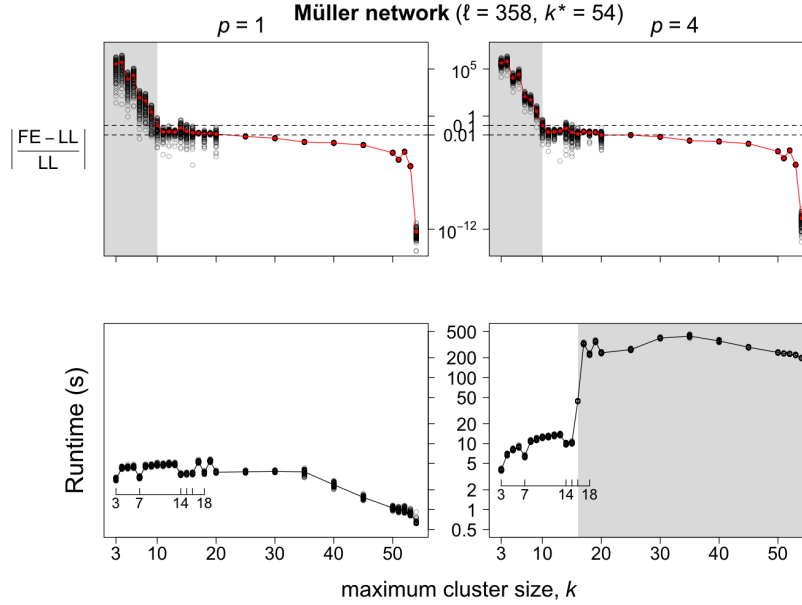


Fig. 3. Relative deviation and runtime (on the log-scale, vertical axis) versus maximum cluster size k for the Müller network, across 100 datasets. Trajectories of the mean values are traced. Top: relative deviation $|(FE - LL)/LL|$. The horizontal lines mark relative deviations of 10% and 1%. The shaded regions indicate values of k with highly inaccurate FE (average error $\geq 10\%$). Bottom: post-compilation runtimes for passing messages and evaluating the FE (upon detection of calibration or after the first 50 iterations). The shaded regions indicate values of k where runtime is notably worse than for other values of k . In the univariate case (left), calibration was consistently detected within 50 iterations after 35. In the multivariate case (right), there is a large increase in runtime around $k = 16$. For $k = 3, 7, 14, 15, 16, 18$ (marked on inset axis), the message schedule used two trees to span the cluster graph, while for other values of k , the schedule used three trees. This explains the observed dips in runtime at $k = 3, 7, 14, 15, 16, 18$.

cases, as the decrease in the number of passed messages may outweigh the increased cost per message. When calibration was not detected within 50 iterations, then the runtime was for 50 iterations exactly, and we expect it to increase with k due to more expensive messages. The observed runtimes generally conformed with our expectations, with two exceptions:

runtime was not monotone decreasing from $k = 3$ to 5 for the Lipson network, and not monotone increasing from $k = 3$ to 19 for the Müller network. The former can be explained by differences in the mean number of iterations before calibration was detected (7.9 ± 0.3 iterations when $k = 4$ versus 11.2 ± 0.4 iterations when $k = 5$), while the latter can be explained by

differences in the length of the message schedule (two versus three spanning trees).

TABLE I
MEAN NUMBER OF ITERATIONS (± 1 STANDARD DEVIATION) TO
CALIBRATION FOR THE MÜLLER NETWORK AT μ_0 AND σ_0^2 (Σ_0).

k	mean number of iterations to calibration	
	univariate ($p = 1$)	multivariate ($p = 4$)
35	43.18 \pm 3.40	46.38 \pm 1.65
40	24.69 \pm 1.65	26.99 \pm 1.40
45	13.34 \pm 0.84	13.99 \pm 0.17
50	7 \pm 0.67	7.66 \pm 0.48
51	5.85 \pm 0.36	6.02 \pm 0.14
52	5.62 \pm 0.51	5.98 \pm 0.14
53	4.18 \pm 0.39	4.62 \pm 0.49
54	2	2

2) *Parameter estimation with a chosen k* : Since $k < k^*$ (using a loopy cluster graph) was chosen only for the Müller network in the multivariate case, we only assessed the accuracy of LBP for parameter estimation in that setting. We noticed a bimodal distribution in the relative deviation $|\widehat{\text{MFE}} - \widehat{\text{ML}}|/\widehat{\text{ML}}$ between the optimized FE (MFE) and the true maximum LL at the true ML estimate ($\widehat{\text{ML}}$). The relative deviation was below 5% for a “good” group of 67 datasets, and exceeded 27% for the remaining “bad” 33 datasets (Fig. D.1).

For the mean parameter, the MFE and ML estimates were similarly far from μ_0 , though close to each other, across both groups (Fig. D.2). For the variance and correlation parameters, the MFE and ML estimates were similarly close to Σ_0 , ρ_0 in the “good” group, though the MFE estimates could be much worse than the corresponding ML estimates in the “bad” group. This “bad” group led to an increased root-mean-square error for the MFE estimator compared to the exact ML estimator (Fig. D.3).

V. DISCUSSION

We showed empirically that the factored energy can closely approximate the log-likelihood using cluster graphs of maximum cluster size k much smaller than needed for exact inference using a clique tree (k^*). Further, we demonstrated that such k , assessed for good FE approximation to the LL at the true parameter values only, can be effective more generally for parameter estimation by numerically optimizing the FE. The resulting MFE estimates were very close to the theoretical ML estimates a majority of the time. Theoretical analysis reveals that this good performance is unsurprising, since the LL surface and the FE surface assuming perfect calibration are parallel over the parameter space under our simulation model:

Theorem 1. Consider the Brownian motion model of evolution for a p -dimensional trait, with root state μ and homogeneous variance rate Σ , on a bicomining phylogenetic network with observed data Y at its leaves. Assume that all tree edges have positive length, and each hybrid node has at least one parent edge with positive length. Let \mathcal{U} be a valid cluster graph for this model, as in Def. 2.5. If \mathcal{U} can be calibrated, then its factored energy differs from the log-likelihood by an additive constant across all $\mu \in \mathbb{R}^p$ and $\Sigma \in \mathbb{R}^{p \times p}$ positive-definite.

Proof. See Appendix B. \square

The prospect of reliable MFE estimation is perhaps only relevant for networks of high complexity, i.e., large k^* , so that there is a potential trade-off between approximation error and runtime. For simpler networks and smaller trait dimensions, belief propagation on a clique tree appears to run faster than LBP.

A. Phase transitions for accuracy and runtime with maximum cluster size

Given that the FE can approximate the LL well using k much smaller than k^* , it is natural to ask (1) if the approximation error decreases with k , and (2) if there is a clear threshold for k beyond which the error becomes acceptably low. In our simulations, the approximation error appeared to decrease with k , (Figs. 2–3). Notably, on the highly complex Müller network, the FE approximation error showed two distinct regimes: a fast rate of decrease followed by a slow rate of decrease, with a regime transition at $k = 11$. This transition aligned with the threshold $k \geq 10$ to reach a reasonable approximation with at most 10% error. For $k < 10$, the error exceeded 100%. The abruptness of this transition suggests that $k \geq 10$ might correspond to the required cluster size for calibration to be possible, though calibration does not intrinsically guarantee low error. Indeed, for the Sikora and Lipson networks, for which the approximation error was always under 0.1%, calibration was detected fairly quickly (within 8 iterations). For the Müller network, calibration was detected consistently (within 50 iterations) only for $k \geq 35$. Increasing the maximum number of iterations beyond 50 showed that calibration could be consistently detected for $k \geq 10$. However, more iterations led the FE to diverge further for $k < 10$. This boost in convergence and divergence from more iterations provides an explanation for the apparent phase transition in the FE accuracy after the calibration threshold $k = 10$.

Runtime generally decreased with k for the Sikora and Lipson networks. In contrast, runtime generally increased until $k \leq 35$ and decreased after $k \geq 35$ for the Müller network. These trends can mostly be explained in terms of the relative importance, for different problem sizes (defined by network size and trait dimension), of having to compute more expensive messages versus having to pass fewer messages for calibration as k increases. However, there are aspects of runtime behavior that require deeper explanation. Notably, for the Müller network for $p = 4$, there was a large increase in runtime around $k = 16$, after which runtime was stable. We suspect that a separate phase transition for runtime is associated with memory latency, only observed when trait dimension and network complexity, both of which parametrize the space complexity of (L)BP, are higher.

Assuming that reasonable accuracy cannot be attained before some calibration threshold k , it makes sense to pick a cluster graph with this k or higher when attempting to find a good trade-off between accuracy and runtime. Additionally, knowledge of the problem size and memory resources available should suggest a practical upper bound for k .

B. Choosing the maximum cluster size k

In practice, we need a way to choose k without expending too much computational effort. For accuracy and runtime considerations, this choice should intuitively depend on network complexity and trait dimension. Our simulations suggest that on networks with complexity as high as $k^* = 54$ and for small trait dimensions ($p \leq 4$), good accuracy may be attained for $k \approx 10$. Our results also suggest that using a clique tree can still be optimal with both lowest runtime and best accuracy for a univariate trait ($p = 1$). Since the space and time complexities of representing a belief and sending messages are parametrized by powers of the dimension pk of the largest cluster's belief precision (Sec. V-C1), a blunt rule could be to choose $k = k^*$ if $pk^* \lesssim 60$ and $p \leq 2$, and $k = \min(cp, k^*)$ otherwise, where $c \approx 3$. This rule interpolates our simulation results. It favors exact BP when runtime is expected to be feasible, and otherwise chooses LBP with large-enough clusters to attempt reliable calibration. Naturally, its applicability is machine-dependent and should be further evaluated. However, it considers the effects of problem size on runtime, and has the advantage of being computationally cheap to apply: k^* is estimated once per network (e.g., using the min-fill heuristic, which has polynomial complexity in the number of nodes), independently of the observed data.

Choosing a good k is necessary but not sufficient for accurate parameter estimation. Another choice to be made is the maximum number of iterations n_{iter} of LBP, at each fixed parameter θ , before $\text{FE}(\theta)$ is computed. Ideally, the FE should be close to its limiting value after n_{iter} iterations. Fortunately, this may occur prior to calibration. In examples from [5, Fig. 6], the FE converged faster and more smoothly towards its limit than did the cluster graph beliefs. In our simulations, we used $n_{\text{iter}} = 50$ with reasonable effectiveness, regardless of k . Alternatively, we could have sought to jointly optimize k and n_{iter} for good runtime and accuracy. For each k and for some fixed reasonable θ , $n_{\text{iter}}(k)$ can be chosen by tracking the convergence of the FE (a computationally expensive task) across iterations of LBP. The overall performance of different $(k, n_{\text{iter}}(k))$ can then be compared over multiple datasets.

C. Making Gaussian LBP useful for phylogenetics

Gaussian LBP opens the door to many applications in phylogenetics, for analyses of large datasets under complex and heterogeneous evolutionary models along complex phylogenetic networks, for which no methods currently exist. The utility of Gaussian LBP for phylogenetics will require reliable parameter estimation and scalability.

1) *Reliability*: MFE estimation appeared to be a reliable proxy for ML estimation, for k and n_{iter} large enough, in a majority of our simulations. However, there were practical challenges in computing the FE. In Gaussian LBP, beliefs and messages are parametrized as log-quadratic forms with a precision matrix \mathbf{K} (positive semidefinite) and a potential

vector h [5, Sec. 4(c)]:

$$\begin{aligned} \beta_i \left(\begin{bmatrix} x_S \\ x_I \end{bmatrix} \right) &\propto \exp \left(-\frac{1}{2} \begin{bmatrix} x_S \\ x_I \end{bmatrix}^\top \overbrace{\begin{bmatrix} \mathbf{K}_S & \mathbf{K}_{S,I} \\ \mathbf{K}_{S,I}^\top & \mathbf{K}_I \end{bmatrix}}^{\mathbf{K}} \begin{bmatrix} x_S \\ x_I \end{bmatrix} \right. \\ &\quad \left. + \begin{bmatrix} h_S \\ h_I \end{bmatrix}^\top \begin{bmatrix} x_S \\ x_I \end{bmatrix} \right) \\ \tilde{\mu}_{i \rightarrow j}(x_S) &\propto \exp \left(-\frac{1}{2} \|x_S\|_{\mathbf{K}/\mathbf{K}_I}^2 \right. \\ &\quad \left. + (h_S - \mathbf{K}_{S,I} \mathbf{K}_I^{-1} h_I)^\top x_S \right) \end{aligned} \quad (6)$$

where x_S are the variables in $\mathcal{S}_{i,j}$ (the Scope of the message), x_I are the remaining variables in $\mathcal{C}_i \setminus \mathcal{S}_{i,j}$ (to be Integrated out) and $\mathbf{K}/\mathbf{K}_I := \mathbf{K}_S - \mathbf{K}_{S,I} \mathbf{K}_I^{-1} \mathbf{K}_{S,I}^\top$. All beliefs must have a positive-definite precision for the FE to be well-defined, and presumably be close enough to calibration for the LL to be well approximated. Thus, the FE is typically computed upon multiple traversals of the cluster graph, which makes the resulting beliefs susceptible to numerical error (e.g., truncation) and hence limits the numerical precision of the FE. To send a message, a principal submatrix of the belief precision is inverted (\mathbf{K}_I above) but numerical error can render this submatrix near-singular and prevent the message from being computed. This hinders calibration since beliefs are unable to “communicate” and keeps the FE ill-defined. This is a crucial issue to deal with when many FE evaluations are required during numerical optimization, such as to approximate the gradient or to adapt the step size using line search.

In practice, we attributed failure to attain positive-definite belief precisions during LBP, despite prior regularization, to numerical errors driven by a poor candidate parameter θ far from the optimal value. For example, belief precisions are scaled by the variance rate Σ in the homogeneous BM model (Appendix B.2) so that large values of Σ can effectively overwhelm adjustments made by regularization, which may be much smaller. To deal with this practical challenge, our implementation returned a very large value for the FE during optimization (e.g., Inf or 10^{10}) whenever an infinite message is encountered or the FE was ill-defined. This had the effect of tuning down the magnitude of the step size of the optimization routine to propose the next candidate value θ , and was effective at steering candidate values towards the right order of magnitude in our simulations.

In general, it is difficult to anticipate how implementation and input (e.g., regularization method, trait model, candidate parameter values, network edge lengths) jointly contribute to numerical errors. Efforts to improve numerical robustness are further complicated by the difficulty of checking if a message fails for numerical or theoretical reasons. Theoretical work is needed to tease these cases apart, e.g., by proving or disproving that regularization can theoretically guarantee well-defined messages throughout LBP.

2) *Scalability*: We suggested that LBP may scale better with network complexity than BP, or the naive approach of computing the inverse covariance for the leaf states to obtain the LL. Here, we compare the cost of these approaches

theoretically as the data and cluster graphs get larger and more complex.

The cost of computing the LL or FE using (L)BP mainly consists of (i) the cumulative cost for messages and (ii) the cost of either fully integrating the root belief to obtain the LL for BP, or evaluating the FE at the final beliefs for LBP (Appendix C). The cost of a message is $\mathcal{O}(d^3)$, where d is the dimension of its precision matrix and is at most kp . Hence, cost (i) is $\mathcal{O}(s(kp)^3)$, where s is the scheduled number of messages to be passed. Cost (ii) is $\mathcal{O}((kp)^3)$ for BP and $\mathcal{O}((v_{cg} + e_{cg})(kp)^3)$ for LBP, where v_{cg} and e_{cg} are the numbers of clusters and edges in the cluster graph. Since the cluster graph is connected and at least $2e_{cg}$ messages need to be passed (one in each edge direction) before computing the FE, we have $v_{cg} + e_{cg} \leq 2e_{cg} \leq s$. Thus, the combined cost of (i) and (ii) is $\mathcal{O}(s(kp)^3)$ for (L)BP. Now using s^* and k^* for BP on a clique tree and keeping s and k for LBP, the above suggests that $sk^3 \ll s^*k^{*3}$ for LBP to be competitive with BP.

The cost of the naive approach depends on how the inverse covariance $\mathbf{V}_{\text{leaf}}^{-1}$ for the leaves is computed. Let n and \tilde{n} be the number of leaves and internal nodes in the network. For linear Gaussian networks, $\mathbf{V}_{\text{leaf}}^{-1}$ can be obtained in $\mathcal{O}((np)^3)$ as follows. In a preorder traversal of the network, we can compute the marginal (co)variances for node v : $\text{var}(X_v) = \mathbf{V}_v + \mathbf{q}_v \text{var}(X_{\text{pa}(v)}) \mathbf{q}_v^\top$, and $\text{cov}(X_v, X_u) = \mathbf{q}_v \text{cov}(X_{\text{pa}(v)}, X_u)$ for u listed before v , using notations from Def. 2.2. Assuming that each hybrid node has a bounded number of parents (typically 2), these operations construct the covariance \mathbf{V}_{all} over all $n + \tilde{n}$ nodes in $\mathcal{O}((n + \tilde{n})^2 p^3)$. For the BM model with the weighted-average rule, \mathbf{q}_v has form $r_v \otimes \mathbf{I}$ for some row vector r_v , so the cost to construct \mathbf{V}_{all} drops to $\mathcal{O}((n + \tilde{n})^2 p^2)$. Finally, we get \mathbf{V}_{leaf} as a submatrix of \mathbf{V}_{all} , and invert it in $\mathcal{O}((np)^3)$. The costs presented for (L)BP and the naive approach are driven by matrix inversion, and have similar multiplicative constants (Appendix C). For (L)BP to be competitive with the naive approach, the above suggests that the scheduled number of messages should be $s < (n/k)^3$.

A clique tree constructed from a chordal graph over the same node set as the original phylogenetic network has at most $n + \tilde{n}$ clusters [38, Prop. 4.16], and join-graph structuring constructs clique trees with exactly $n + \tilde{n}$ clusters [11]. In either case, the number of messages s^* is at most $2(n + \tilde{n})$. Combining with the earlier criterion, $(n + \tilde{n}) \ll (n/k^*)^3$ is favorable for BP to be competitive with the naive approach. In a rooted binary network $\tilde{n} = n - 1 + 2h$ [39, Lem. 2.1], so this criterion simplifies to $(n + h) \ll (n/k^*)^3$.

We illustrate these bounds using the Müller network, whose number of leaves $n = 40$ is moderately small, but with $h = 361$ hybrids. Our clique tree had $k^* = 54$, $e_{cg}^* = 800$ edges, and $s^* = 1600$ to pass a message in each edge direction. Our cluster graph with $k = 11$ had $e_{cg} = 1160$ and required $s \approx (980 \times 2) \times 3 \times 50$ from traversing 3 spanning trees repeatedly $n_{\text{iter}} = 50$ times. As $sk^3 > s^*k^{*3}$, LBP is expected to be slower than BP, which aligns with our runtime results in the univariate case (Fig. 3, bottom-left). However, both are expected to be slower than the naive approach for the BM model, because they require to pass many more messages than

$(n/k^*)^3 \approx 0.4$ (for BP) or $(n/k)^3 \approx 48$ (for LBP).

The optimal number of scheduled messages s^* is known for BP on given a clique tree: $s^* = e_{cg}^*$ (from one traversal). However, for LBP it is generally unclear how to devise schedules that minimize s , while getting close to calibration. The reliance on suboptimal message schedules limits the utility of LBP. Adaptive schedules have been proposed to improve the per-message efficiency of attaining calibration [40]–[43], yet these use more memory and introduce a computational overhead for selecting messages. Although convergence of the FE is more relevant than calibration for deciding when to stop passing messages, it is less used since repeated evaluations of the FE are computationally impractical.

3) *Possible use cases*: The theoretical bounds above, which do not account for implementation details such as cost of memory access, shed some light on possible use cases for LBP to fit Gaussian trait models on phylogenetic networks. For example, it may not be easy for (L)BP to decrease runtime compared to the naive approach since the covariance matrix for the leaves can be efficiently constructed. This applies to standard models such as the BM or Ornstein-Uhlenbeck (OU) process, possibly combined with edge length transformations [8, Ch. 6] or heterogeneity in the evolutionary parameters (e.g., the variance rate and optimum for the OU model) across different sets of edges in the network. For LBP to be beneficial, both $(n/k)^3$ and $s^*(k^*/k)^3$ have to be large enough to accommodate multiple traversals of the cluster graph. These conditions may be less favorable for smaller networks, which generally have smaller k^* [5, Fig. 5].

In conclusion, our analysis points to two main settings where LBP could be useful: large phylogenetic studies on thousands of leaves [44]–[47] when their history involves reticulate evolution, and phylodynamic studies, which increasingly involve large highly-reticulated networks [32], [48], [49]. In particular, the framework it provides is amenable to the use of flexible multi-regime Gaussian trait models [19], [50] that we expect to see wider usage of in both settings.

D. Future work

We conclude by listing several theoretical and engineering challenges that could be tackled.

- 1) Do our regularization strategy and spanning trees message schedule, in combination, ensure that all messages are theoretically well-defined? If not, can alternative belief initializations and schedules provide this guarantee, or a stronger one that ensures that belief precisions remain positive-definite after every belief update?
- 2) How can we best incorporate regularization during message passing to improve robustness to numerical error? For example, if we know theoretically that a message should be finite, but requires inverting a submatrix that is singular due to numerical error, we can regularize the sender and sepset beliefs for this submatrix to be invertible. This does not approximate the original message but allows LBP to proceed, and can be applied regardless of whether the message is theoretically well-defined. Additional heuristics or theory could improve regularization techniques during message passing.

- 3) We relied exclusively on join-graph structuring to construct cluster graphs of a desired maximum cluster size k . Can modifications or alternatives to join-graph structuring be devised to produce cluster graphs with better structural features to achieve calibration more often and faster, given a desired k ? For example, larger sepsets tend to be favorable for calibration. However, join-graph structuring typically includes multiple size-1 sepsets by construction.
- 4) Where to initialize the optimization? We used parameter estimates that assume no phylogenetic correlation, which adapt to the data yet are fast to calculate. A reasonable improvement could be to use the ML estimates assuming the *major tree* — the tree obtained by keeping only the parent edge with the largest inheritance weight, for each node in the network⁴. These estimates can be efficiently computed for the BM model [51]. For other models such as the OU process, even naive estimates that assume no phylogenetic correlation require numerical optimization for some parameters.

This non-exhaustive list stresses the equal importance of theory and implementation for making this tool more useful to applied researchers.

VI. ACKNOWLEDGEMENTS

We thank Paul Bastide for helpful feedback on an earlier draft. This work was supported in part by the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation; and by the National Science Foundation through grants DMS-2023239 to CA and DMS-1929284 while BT and CA were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the “Theory, Methods, and Applications of Quantitative Phylogenomics” program.

SOFTWARE AND DATA AVAILABILITY

Data and code for all simulations and analyses are available at https://github.com/bstjk/adaptingclustergraphs_code. In particular, the Julia package `PhyloGaussianBeliefProp` was used at commit `f801d07`.

REFERENCES

- [1] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.
- [2] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, “Walk-sums and belief propagation in gaussian graphical models,” *The Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.
- [3] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [4] F. Kamper, S. J. Steel, and J. A. du Preez, “On the convergence of gaussian belief propagation with nodes of arbitrary size,” *Journal of Machine Learning Research*, vol. 20, no. 165, pp. 1–37, 2019. [Online]. Available: <http://jmlr.org/papers/v20/18-040.html>
- [5] B. Teo, P. Bastide, and C. Ané, “Leveraging graphical model techniques to study evolution on phylogenetic networks,” *Phil. Trans. R. Soc. B*, vol. 380, no. 20230310, 2025.
- [6] S. Streicher and J. du Preez, “Graph coloring: Comparing cluster graphs to factor graphs,” in *Proceedings of the ACM Multimedia 2017 Workshop on South African Academic Participation*, 2017, pp. 35–42.
- [7] V. Mitov, K. Bartoszek, G. Asimomitis, and T. Stadler, “Fast likelihood calculation for multivariate Gaussian phylogenetic models with shifts,” *Theoretical Population Biology*, vol. 131, pp. 66–78, 2020.
- [8] L. J. Harmon, *Phylogenetic comparative methods*. EcoEvoRxiv, 2019.
- [9] M. Lipson, “Applying f4-statistics and admixture graphs: Theory and examples,” *Molecular Ecology Resources*, vol. 20, no. 6, pp. 1658–1667, 2020.
- [10] A. L. Lewanski, M. C. Grundler, and G. S. Bradburd, “The era of the arg: An introduction to ancestral recombination graphs and their significance in empirical evolutionary genomics,” *PLoS Genetics*, vol. 20, no. 1, p. e1011110, 2024.
- [11] R. Mateescu, K. Kask, V. Gogate, and R. Dechter, “Join-graph propagation algorithms,” *Journal of Artificial Intelligence Research*, vol. 37, pp. 279–328, 2010.
- [12] O. G. Pybus, M. A. Suchard, P. Lemey, F. J. Bernardin, A. Rambaut, F. W. Crawford, R. R. Gray, N. Arinaminpathy, S. L. Stramer, M. P. Busch, and E. L. Delwart, “Unifying the spatial epidemiology and molecular evolution of emerging epidemics,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 37, pp. 15 066–15 071, Sep. 2012.
- [13] K. Bartoszek, J. F. Gonzalez, V. Mitov, J. Pienaar, M. Piwczyński, R. Puchałka, K. Spalik, and K. L. Vojte, “Model selection performance in phylogenetic comparative methods under multivariate Ornstein-Uhlenbeck models of trait evolution,” *Systematic Biology*, vol. 72, no. 2, pp. 275–293, 2023.
- [14] J. P. Rose, C. A. P. Toledo, E. M. Lemmon, A. R. Lemmon, and K. J. Sytsma, “Out of sight, out of mind: widespread nuclear and plastid-nuclear discordance in the flowering plant genus *Polemonium* (Polemoniaceae) suggests widespread historical gene flow despite limited nuclear signal,” *Systematic Biology*, vol. 70, no. 1, pp. 162–180, 2021.
- [15] L. Sagart, G. Jacques, Y. Lai, R. J. Ryder, V. Thouzeau, S. J. Greenhill, and J.-M. List, “Dated language phylogenies shed light on the ancestry of sino-tibetan,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 21, pp. 10 317–10 322, 2019.
- [16] M. Steel, *Phylogeny: Discrete and Random Processes in Evolution*. SIAM-Society for Industrial and Applied Mathematics, 2016.
- [17] M. Manceau, A. Lambert, and H. Morlon, “A unifying comparative phylogenetic framework including traits coevolving across interacting lineages,” *Systematic Biology*, vol. 66, no. 4, pp. 551–568, 2017.
- [18] C.-E. Rabier, V. Berry, M. Stoltz, J. D. Santos, W. Wang, J.-C. Glaszmann, F. Pardi, and C. Scornavacca, “On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo,” *PLOS Computational Biology*, vol. 17, no. 9, pp. 1–39, 2021.
- [19] V. Mitov, K. Bartoszek, and T. Stadler, “Automatic generation of evolutionary hypotheses using mixed Gaussian phylogenetic models,” *Proceedings of the National Academy of Sciences*, p. 201813823, Aug. 2019.
- [20] P. Bastide, L. S. T. Ho, G. Baele, P. Lemey, and M. A. Suchard, “Efficient Bayesian inference of general Gaussian models on large phylogenetic trees,” *The Annals of Applied Statistics*, vol. 15, no. 2, pp. 971–997, 2021.
- [21] P. Bastide and G. Didier, “The cauchy process on phylogenies: a tractable model for pulsed evolution,” *Systematic Biology*, vol. 72, no. 6, pp. 1296–1315, 2023.
- [22] L. S. T. Ho and C. Ané, “A linear-time algorithm for gaussian and non-gaussian trait evolution models,” *Systematic Biology*, vol. 63, no. 3, pp. 397–408, 2014.
- [23] P. Bastide, C. Solís-Lemus, R. Kriebel, K. William Sparks, and C. Ané, “Phylogenetic comparative methods on phylogenetic networks with reticulations,” *Systematic Biology*, vol. 67, no. 5, pp. 800–820, 2018.
- [24] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, “Tree-based reparameterization framework for analysis of sum-product and related algorithms,” *IEEE Transactions on information theory*, vol. 49, no. 5, pp. 1120–1146, 2003.
- [25] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [26] L. J. Revell and L. J. Harmon, “Testing quantitative genetic hypotheses about the evolutionary rate matrix for continuous characters,” *Evolutionary Ecology Research*, vol. 10, pp. 311–331, 2008.

⁴Unlabelled leaves, with no data, are pruned from the major tree. These may arise if the network is not *tree-child*, that is, if some internal node has hybrid children only.

- [27] H. Glanz and L. Carvalho, "An expectation–maximization algorithm for the matrix normal distribution with an application in remote sensing," *Journal of Multivariate Analysis*, vol. 167, pp. 31–48, 2018.
- [28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. MIT Press, 2009.
- [29] P. Gambette, V. Berry, and C. Paul, "The structure of level-k phylogenetic networks," in *Annual Symposium on Combinatorial Pattern Matching*, 2009, pp. 289–300.
- [30] M. Sikora, V. V. Pitulko, V. C. Sousa, M. E. Allentoft, L. Vinner, S. Rasmussen, A. Margaryan, P. de Barros Damgaard, C. de la Fuente, G. Renaud *et al.*, "The population history of northeastern siberia since the pleistocene," *Nature*, vol. 570, no. 7760, pp. 182–188, 2019.
- [31] M. Lipson, I. Ribot, S. Mallick, N. Rohland, I. Olalde, N. Adamski, N. Broomandkhoshbacht, A. M. Lawson, S. López, J. Oppenheimer *et al.*, "Ancient west african foragers in the context of african population history," *Nature*, vol. 577, no. 7792, pp. 665–670, 2020.
- [32] N. F. Müller, K. E. Kistler, and T. Bedford, "A bayesian approach to infer recombination patterns in coronaviruses," *Nature communications*, vol. 13, no. 1, p. 4186, 2022.
- [33] C. Solís-Lemus, P. Bastide, and C. Ané, "PhyloNetworks: A package for phylogenetic networks," *Molecular Biology and Evolution*, vol. 34, no. 12, pp. 3292–3298, 2017.
- [34] J. T. Thorson and W. van der Bijl, "phylosem: A fast and simple R package for phylogenetic inference and trait imputation using phylogenetic structural equation models," *Journal of Evolutionary Biology*, vol. 36, no. 10, pp. 1357–1364, 2023.
- [35] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.
- [36] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering, 2006.
- [37] W. W. Hager and H. Zhang, "Algorithm 851: Cg_descent, a conjugate gradient method with guaranteed descent," *ACM Transactions on Mathematical Software (TOMS)*, vol. 32, no. 1, pp. 113–137, 2006.
- [38] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., 2004.
- [39] C. McDiarmid, C. Semple, and D. Welsh, "Counting phylogenetic networks," *Annals of Combinatorics*, vol. 19, no. 1, pp. 205–224, 2015.
- [40] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: informed scheduling for asynchronous message passing," in *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006, p. 165–173.
- [41] C. Sutton and A. McCallum, "Improved dynamic schedules for belief propagation," in *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 2007, p. 376–383.
- [42] C. Knoll, M. Rath, S. Tschitschek, and F. Pernkopf, "Message scheduling methods for belief propagation," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15*, 2015, pp. 295–310.
- [43] V. Aksenov, D. Alistarh, and J. H. Korhonen, "Scalable belief propagation via relaxed scheduling," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 361–22 372, 2020. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/fdb2c3bab9d0701c4a050a4d8d782c7f-Paper.pdf
- [44] J. Kattge, G. Bönsch, S. Díaz, S. Lavorel, I. C. Prentice, P. Leadley, S. Tautenhahn, G. D. Werner, T. Aakala, M. Abedi *et al.*, "TRY plant trait database–enhanced coverage and open access," *Global change biology*, vol. 26, no. 1, pp. 119–188, 2020.
- [45] J. A. Tobias, C. Sheard, A. L. Pigot, A. J. Devenish, J. Yang, F. Sayol, M. H. Neate-Clegg, N. Alioravainen, T. L. Weeks, R. A. Barber *et al.*, "Avonet: morphological, ecological and geographical data for all birds," *Ecology Letters*, vol. 25, no. 3, pp. 581–597, 2022.
- [46] D. S. Maynard, L. Bialic-Murphy, C. M. Zohner, C. Averill, J. van den Hoogen, H. Ma, L. Mo, G. R. Smith, A. T. Acosta, I. Aubin *et al.*, "Global relationships in tree functional traits," *Nature Communications*, vol. 13, no. 1, p. 3185, 2022.
- [47] M. Peeri and T. Tuller, "High-resolution modeling of the selection on local mrna folding strength in coding sequences across the tree of life," *Genome Biology*, vol. 21, pp. 1–20, 2020.
- [48] M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, and A. Rambaut, "Bayesian phylogenetic and phylodynamic data integration using beast 1.10," *Virus evolution*, vol. 4, no. 1, p. vey016, 2018.
- [49] N. F. Müller, U. Stolz, G. Dudas, T. Stadler, and T. G. Vaughan, "Bayesian inference of reassortment networks reveals fitness benefits of reassortment in human influenza viruses," *Proceedings of the National Academy of Sciences*, vol. 117, no. 29, pp. 17 104–17 111, 2020.
- [50] B. Brahmantio, K. Bartoszek, and E. Yapar, "Bayesian inference of mixed gaussian phylogenetic models," 2025. [Online]. Available: <https://arxiv.org/abs/2410.11548>
- [51] R. P. Freckleton, "Fast likelihood calculations for comparative analyses," *Methods in Ecology and Evolution*, vol. 3, no. 5, pp. 940–947, 2012.
- [52] B. Teo, P. Bastide, and C. Ané, "Supplementary material from: Leveraging graphical model techniques to study evolution on phylogenetic networks," 2025.
- [53] G. H. Golub and C. F. Van Loan, *Matrix Computations - 4th Edition*. Philadelphia, PA: Johns Hopkins University Press, 2013.

APPENDIX A

COVARIANCES FROM PATH DIAGRAM

We derive here the trait covariance (rescaled by 10^{-3}) implied by the path analysis in [34], which we used in our simulation study for the multivariate case with $p = 4$ traits.

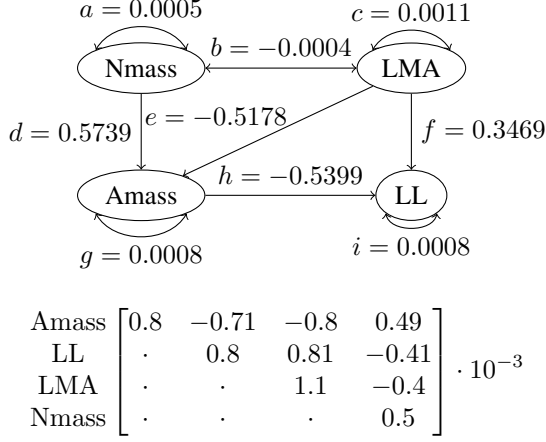


Fig. A.1. Left: Path diagram of four leaf traits: photosynthetic rate (Amass), leaf lifespan (LL), leaf mass per area (LMA), leaf nitrogen content (Nmass), reproduced from [34, Fig. 2, top]. Self-loops are labeled by the trait's variance, e.g., $\text{var}(\text{LMA}) = 0.0011$. Bidirectional edges between two traits are labeled by their covariance, e.g., $\text{cov}(\text{Nmass}, \text{LMA}) = -0.0004$. Unidirectional edges from one trait to another are labeled by linear dependence weights, e.g., $\text{Amass} = d \cdot \text{Nmass} + e \cdot \text{LMA} + \varepsilon$. Right: Covariance matrix implied by the path diagram. Each covariance is the sum of contributions from all valid paths between the two traits. The contribution of each path is determined by its edge weights. Calculations of the covariances from adding up distinct path contributions are given below. For example, there are three paths for $\text{cov}(\text{Amass}, \text{LL})$ and two paths for $\text{cov}(\text{LMA}, \text{Amass})$.

$$\begin{aligned}
 \text{cov}(\text{Amass}, \text{LL}) &= gh + ecf + dbf = -0.00070914 \\
 \text{cov}(\text{LMA}, \text{Amass}) &= ce + bd = -0.00079914 \\
 \text{cov}(\text{LMA}, \text{LL}) &= cf + ce h + bd h = 0.00081305 \\
 \text{cov}(\text{Nmass}, \text{Amass}) &= ad + be = 0.00049407 \\
 \text{cov}(\text{Nmass}, \text{LL}) &= adh + beh + bf = -0.00040551
 \end{aligned}$$

APPENDIX B

LIKELIHOOD AND FACTORED ENERGY SURFACES

We consider a bicompleting phylogenetic network $N = (V, E)$ with n leaves, where all tree edges have positive length, and each hybrid node has at least one parent edge with positive length. On this network, we assume the BM model for the evolution of a p -dimensional trait with state μ at the root of the network and homogeneous variance rate Σ .

To prove Theorem 1, we derive difference equations for how the FE changes when Σ is linearly transformed or when μ is shifted, and prove their equivalence to the difference equations for the LL. This will show that the LL and FE surfaces are parallel, as claimed in Theorem 1.

Using notations from section III-A, $Y = \text{vec}([Y_1 \dots Y_n]) \in \mathbb{R}^{np}$ is the random vector of the leaf states, distributed as $Y \sim \mathcal{N}(1_n \otimes \mu, \mathbf{P}_y \otimes \Sigma)$, and \mathbf{Y} is the corresponding vector of observed data.

1. Log-likelihood difference equations

Let $\text{LL}(\mu, \Sigma)$ denote the LL at the parameters μ and Σ , and define $\delta_1^{\text{LL}}(\mu_1, \mu_2, \Sigma) := \text{LL}(\mu_1, \Sigma) - \text{LL}(\mu_2, \Sigma)$ and $\delta_2^{\text{LL}}(\mu, \Sigma_1, \Sigma_2) := \text{LL}(\mu, \Sigma_1) - \text{LL}(\mu, \Sigma_2)$. The following equations are easily derived:

$$\delta_2^{\text{LL}}(\mu, \mathbf{A}\Sigma, \Sigma) = -\frac{1}{2} \|\mathbf{Y} - 1_n \otimes \mu\|_{\mathbf{U}}^2 - \frac{n}{2} \log |\mathbf{A}| \quad (7)$$

$$\begin{aligned}
 \delta_1^{\text{LL}}(\mu_2, \mu_1, \Sigma) &= -\frac{1}{2} \|1_n \otimes \Delta\mu\|_{\mathbf{V}_y}^2 \\
 &\quad + (1_n \otimes \Delta\mu)^\top \mathbf{V}_y^{-1} (\mathbf{Y} - 1_n \otimes \mu_1)
 \end{aligned} \quad (8)$$

where $\mathbf{V}_y = \mathbf{P}_y \otimes \Sigma$, $\mathbf{U} = \mathbf{V}_y^{-1}(\mathbf{I}_n \otimes (\mathbf{A}^{-1} - \mathbf{I}_p))$, $\mathbf{A} \in \mathbb{R}^{p \times p}$ is full-rank such that $\mathbf{A}\Sigma$ is symmetric positive-definite, and $\Delta\mu = \mu_2 - \mu_1$.

2. Factored energy difference equations

To derive the corresponding quantities for the FE (4), we consider the energy and entropy terms separately:

$$\tilde{F}_{\mathcal{U}}(q^*) = \underbrace{\sum_{C_i \in \mathcal{V}} E_{\beta_i^*}(\log \psi_i | \mathbf{Y})}_{\text{energy term}} + \underbrace{\sum_{C_i \in \mathcal{V}} H(\beta_i^*) - \sum_{\{C_i, C_j\} \in \mathcal{E}} H(\mu_{i,j}^*)}_{\text{entropy term}}$$

At calibration, $\sum_{C_i \in \mathcal{V}} E_{\beta_i^*}(\log \psi_i) = \sum_{v \in V \setminus \{\rho\}} E_{\beta_v^*}(\log \phi_v)$, where β_v^* is the belief of some cluster that contains node v and its parents, ϕ_v is the factor for node v (Def. 2.4), and ρ is the root node. We work with $\sum_{v \in V \setminus \{\rho\}} E_{\beta_v^*}(\log \phi_v)$, which is more convenient for our calculations.

1) *Varying Σ for a fixed μ :* Under the BM model, we have $X_v | X_{\text{pa}(v)} \sim \mathcal{N}((\gamma \otimes \mathbf{I}_p) X_{\text{pa}(v)}, \ell_v \Sigma)$. If v is a tree node, $\gamma = [1]$ and ℓ_v is the length of v 's parent edge. If v is a hybrid, $\gamma = [\gamma_1 \ \gamma_2]$ contains the inheritance weights of the parent edges of v and ℓ_v is a weighted-average of their lengths, weighted by their inheritance weights. Then

$$\begin{aligned}
 E_{\beta_v^*}(\log \phi_v) &= -\frac{1}{2} E_{\beta_v^*}(\|X_{v, \text{pa}(v)}\|_{\mathbf{B}_v \otimes (\ell_v \Sigma)}^2) \\
 &\quad + \log |2\pi \ell_v \Sigma| \\
 &= -\frac{1}{2} \left(\|\hat{\mu}_{v, \text{pa}(v)}\|_{\mathbf{B}_v \otimes (\ell_v \Sigma)}^2 + \log |2\pi \ell_v \Sigma| \right. \\
 &\quad \left. + \text{tr}((\mathbf{B}_v \otimes (\ell_v \Sigma)^{-1}) [\mathbf{K}_{\beta_v^*}^{-1}]_{v, \text{pa}(v)}) \right)
 \end{aligned} \quad (9)$$

where $\mathbf{B}_v = [1, -1]^\top [1, -1]$ if v is a tree node, $\mathbf{B}_v = [1, -\gamma_1, -\gamma_2]^\top [1, -\gamma_1, -\gamma_2]$ if v is a hybrid node, and $X_{v, \text{pa}(v)}$ stacks the states of node v on top of those of its parents. Upon calibration, $\hat{\mu}_{v, \text{pa}(v)} = E(X_{v, \text{pa}(v)} | \mathbf{Y})$ does not depend on \mathcal{U} , or on Σ [52, Sec. SM-4]. $[\mathbf{K}_{\beta_v^*}^{-1}]_{v, \text{pa}(v)}$ denotes the submatrix of $\mathbf{K}_{\beta_v^*}^{-1}$ corresponding to $X_{v, \text{pa}(v)}$, where $\mathbf{K}_{\beta_v^*} = \mathbf{J}_{\beta_v^*} \otimes \Sigma^{-1}$ is the precision of β_v^* and $\mathbf{J}_{\beta_v^*}$ does not depend on μ , Σ , or \mathbf{Y} [52, Lem. 3]. Note that $\text{tr}((\mathbf{B}_v \otimes (\ell_v \Sigma)^{-1}) [\mathbf{K}_{\beta_v^*}^{-1}]_{v, \text{pa}(v)})$ does not depend on Σ .

We are now ready to determine how the energy varies with the variance rate Σ . Letting $\beta^*(\cdot)$ denote β^* as a function of Σ :

$$\begin{aligned} & \sum_{\mathcal{C}_i \in \mathcal{V}} (\mathbb{E}_{\beta_i^*(\mathbf{A}\Sigma)}(\log \psi_i) - \mathbb{E}_{\beta_i^*(\Sigma)}(\log \psi_i)) \\ &= \sum_{v \in V \setminus \{\rho\}} (\mathbb{E}_{\beta_v^*(\mathbf{A}\Sigma)}(\log \phi_v) - \mathbb{E}_{\beta_v^*(\Sigma)}(\log \phi_v)) \\ &= \sum_{v \in V \setminus \{\rho\}} \left(-\frac{1}{2} \|\hat{\mu}_{v, \text{pa}(v)}\|_{\mathbf{U}_v}^2 - \frac{1}{2} \log |\mathbf{A}| \right) \end{aligned} \quad (10)$$

where $\mathbf{U}_v = (\ell_v^{-1} \mathbf{B}_v \otimes \Sigma^{-1})(\mathbf{I}_{n_v} \otimes (\mathbf{A}^{-1} - \mathbf{I}_p))$ does not depend on \mathcal{U} , and n_v is the number of rows or columns of \mathbf{B}_v ($n_v = 3$ if v is hybrid, 2 otherwise). When there is absorption of evidence in ϕ_v , e.g., if the value of X_v or $X_{\text{pa}(v)}$ is observed or fixed as a parameter, the effect on (9) is that the trace term changes but remains independent of Σ . Thus, (10) continues to hold in this case.

We now consider the entropy term, starting with the contribution of a single cluster:

$$\begin{aligned} \mathbb{H}(\beta_i^*(\Sigma)) &= (1 + \log 2\pi) \frac{m_i p}{2} + \frac{1}{2} \log |\mathbf{J}_{\beta_i^*}^{-1} \otimes \Sigma| \\ \mathbb{H}(\beta_i^*(\mathbf{A}\Sigma)) - \mathbb{H}(\beta_i^*(\Sigma)) &= \frac{m_i}{2} \log |\mathbf{A}| \end{aligned} \quad (11)$$

where $m_i = |\mathcal{C}_i|$. (11) holds for edge beliefs $\mu_{i,j}^*$ as well, but with $m_{i,j} = |\mathcal{S}_{i,j}|$ and $\mathbf{J}_{\mu_{i,j}^*}$ replacing m_i and $\mathbf{J}_{\beta_i^*}$. Summing over the contributions from all clusters and edges:

$$\begin{aligned} & \sum_{\mathcal{C}_i \in \mathcal{V}} (\mathbb{H}(\beta_i^*(\mathbf{A}\Sigma)) - \mathbb{H}(\beta_i^*(\Sigma))) \\ & - \sum_{\{\mathcal{C}_i, \mathcal{C}_j\} \in \mathcal{E}} (\mathbb{H}(\mu_{i,j}^*(\mathbf{A}\Sigma)) - \mathbb{H}(\mu_{i,j}^*(\Sigma))) \\ &= \frac{1}{2} \left(\sum_{\mathcal{C}_i \in \mathcal{V}} m_i - \sum_{\{\mathcal{C}_i, \mathcal{C}_j\} \in \mathcal{E}} m_{i,j} \right) \log |\mathbf{A}| \\ &= \frac{|V| - n - 1}{2} \log |\mathbf{A}| \end{aligned} \quad (12)$$

where the last equality is a consequence of the running-intersection property (Def. 2.5, condition 3). Indeed, for each non-root internal node in N , the clusters that contain that node induce a subtree of \mathcal{U} , so that the node appears in one more cluster than sepset.

Let $\tilde{F}_{\mathcal{U}}(\mu, \Sigma)$ denote the FE at the parameters μ and Σ , and define $\delta_2^{\text{FE}}(\mu, \Sigma_1, \Sigma_2) := \tilde{F}_{\mathcal{U}}(\mu, \Sigma_1) - \tilde{F}_{\mathcal{U}}(\mu, \Sigma_2)$. Then combining the energy and entropy terms gives us:

$$\delta_2^{\text{FE}}(\mu, \mathbf{A}\Sigma, \Sigma) = \sum_{v \in V \setminus \{\rho\}} \left(-\frac{1}{2} \|\hat{\mu}_{v, \text{pa}(v)}\|_{\mathbf{U}_v}^2 - \frac{n}{2} \log |\mathbf{A}| \right). \quad (13)$$

2) *Varying μ for a fixed Σ* : Define $\delta_1^{\text{FE}}(\mu_1, \mu_2, \Sigma) := \tilde{F}_{\mathcal{U}}(\mu_1, \Sigma) - \tilde{F}_{\mathcal{U}}(\mu_2, \Sigma)$ and let $\hat{\mu}_{v, \text{pa}(v)}(\cdot)$ denote $\hat{\mu}_{v, \text{pa}(v)}$ as a function of the root state μ . From (11), the entropy term of the FE does not depend on μ . From the energy term in (9), we get:

$$\begin{aligned} \delta_1^{\text{FE}}(\mu_2, \mu_1, \Sigma) &= -\frac{1}{2} \sum_{v \in V \setminus \{\rho\}} \left(\|\hat{\mu}_{v, \text{pa}(v)}(\mu_2)\|_{\mathbf{B}_v \otimes (\ell_v \Sigma)^{-1}}^2 \right. \\ & \quad \left. - \|\hat{\mu}_{v, \text{pa}(v)}(\mu_1)\|_{\mathbf{B}_v \otimes (\ell_v \Sigma)^{-1}}^2 \right). \end{aligned} \quad (14)$$

3. Comparing the log-likelihood and factored energy surfaces

The difference equations (7) and (13) are equal for a clique tree since the factored energy of a clique tree is equal to the log-likelihood. Further, the terms in (13) do not depend on the cluster graph \mathcal{U} as long as calibration is attained. Therefore, equality must hold for any calibrated cluster graph, that is: $\delta_2^{\text{LL}} = \delta_2^{\text{FE}}$. By the same reasoning, the difference equations (8) and (14) must be equal for any calibrated cluster graph, and $\delta_1^{\text{LL}} = \delta_1^{\text{FE}}$.

Now letting $\delta(\mu, \Sigma) = \text{LL}(\mu, \Sigma) - \tilde{F}_{\mathcal{U}}(\mu, \Sigma)$, we get

$$\begin{aligned} & \delta(\mu + \Delta\mu, \mathbf{A}\Sigma) \\ &= (\text{LL}(\mu, \mathbf{A}\Sigma) + \delta_1^{\text{FE}}(\mu + \Delta\mu, \mu, \mathbf{A}\Sigma)) - \tilde{F}_{\mathcal{U}}(\mu + \Delta\mu, \mathbf{A}\Sigma) \\ &= \text{LL}(\mu, \mathbf{A}\Sigma) - \tilde{F}_{\mathcal{U}}(\mu, \mathbf{A}\Sigma) \\ &= (\text{LL}(\mu, \Sigma) + \delta_2^{\text{FE}}(\mu, \mathbf{A}\Sigma, \Sigma)) - \tilde{F}_{\mathcal{U}}(\mu, \mathbf{A}\Sigma) = \delta(\mu, \Sigma) \end{aligned}$$

which implies that δ is a constant function over μ and Σ positive definite. Thus, the FE is equal to the LL up to an additive constant as claimed in Theorem 1, and optimizing either quantity is theoretically equivalent for this homogeneous BM model. In practice however, the FE is computed from beliefs that are only approximately calibrated, and this may translate to different optimization landscapes for both quantities.

APPENDIX C COMPLEXITY ESTIMATES

We evaluate here the cost of BP, LBP, and inverting the covariance for the leaves to calculate the LL or FE, in terms of number of floating-point operations.

1. Cost of passing a message

From (6), a message $\tilde{\mu}_{i \rightarrow j}$ is obtained by marginalizing out variables x_1 from the scope $x = \text{vec}([x_S \ x_I])$ of a belief β_i with parameters

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_S & \mathbf{K}_{S,I} \\ \mathbf{K}_{S,I}^\top & \mathbf{K}_I \end{bmatrix}, \quad h = \begin{bmatrix} h_S \\ h_I \end{bmatrix} \quad \text{and } g,$$

where $\exp(g)$ is the constant of proportionality, that is, $\beta_i(x) = \exp(-\|x\|_{\mathbf{K}}^2/2 + h^\top x + g)$. Computing $\tilde{\mu}_{i \rightarrow j}$ requires computing the message parameters [5, Alg. 2]:

$$\begin{aligned} \mathbf{K}_{\text{msg}} &= \mathbf{K}_S - \mathbf{K}_{S,I} \mathbf{K}_I^{-1} \mathbf{K}_{S,I} \\ h_{\text{msg}} &= h_S - \mathbf{K}_{S,I} \mathbf{K}_I^{-1} h_I \\ g_{\text{msg}} &= g + (\log |2\pi \mathbf{K}_I^{-1}| + \|h_I\|_{\mathbf{K}_I^{-1}})/2. \end{aligned}$$

Computing \mathbf{K}_I^{-1} from $\mathbf{K}_I \in \mathbb{R}^{d_I \times d_I}$ using an LU decomposition uses $(5/3)d_I^3 + \mathcal{O}(d_I^2)$ flops [53, Sec. 3.1–3.2], where $d_I \leq kp$, k is the maximum cluster size, and p is the trait dimension. Given \mathbf{K}_I^{-1} , only the matrix product $\mathbf{K}_{S,I} \mathbf{K}_I^{-1} \mathbf{K}_{S,I}^\top$ in the computation of \mathbf{K}_{msg} may have above-quadratic complexity in kp . For $\mathbf{K}_{S,I} \in \mathbb{R}^{d_S \times d_I}$, the cost of sequentially evaluating $((\mathbf{K}_{S,I} \mathbf{K}_I^{-1}) \mathbf{K}_{S,I}^\top)$ uses $2(d_S d_I^2 + d_S^2 d_I)$ flops [53, Tab. 1.1.2], which equals $(kp)^3/2$ in the worst case when $d_S = d_I = kp/2$. Thus, the cost of computing a message is at most $(5/3 + 1/2)(kp)^3 \approx 2(kp)^3$ if the lower-order terms are dropped. This upper bound applies to BP and LBP regardless of the specific cluster and sepset involved, though the average

cost per message depends on the whole distribution of cluster and sepsset sizes and the message schedule, and is impractical to evaluate.

2. Cost of evaluating the FE

From Appendix B.2, the FE is the sum of an energy term and an entropy term. Cluster beliefs contribute to the energy and entropy, and edge beliefs only contribute to the entropy. Their contributions are independent and their costs add up to the total cost of evaluating the FE. For a given cluster or edge, let $\exp(-\|x\|_{\mathbf{K}_0}^2/2 + h_0^\top x + g_0)$ be the initial belief before regularization, $\exp(-\|x\|_{\mathbf{K}}^2/2 + h^\top x + g)$ be the final belief before the FE is computed, and d be the dimension of \mathbf{K} .

The energy of a cluster belief is $\exp(-\|\mathbf{K}^{-1}h\|_{\mathbf{K}_0}^2/2 + \text{tr}(\mathbf{K}_0\mathbf{K}^{-1}) + h_0^\top(\mathbf{K}^{-1}h) + g_0)$. Computing \mathbf{K}^{-1} from \mathbf{K} using an LU decomposition uses $(5/3)d^3 + \mathcal{O}(d^2)$ flops. Given \mathbf{K}^{-1} , only $\text{tr}(\mathbf{K}_0\mathbf{K}^{-1})$ potentially has above-quadratic complexity in d . However, this term can be computed in $\mathcal{O}(d)$ flops by considering only the diagonal entries of $\mathbf{K}_0\mathbf{K}^{-1}$. Hence, the cost of evaluating the energy term of the FE is at most $(5/3)v_{\text{cg}}(kp)^3$, where v_{cg} is the number of clusters in the cluster graph, if the lower-order terms are dropped.

The entropy of a belief is $-\log(|\mathbf{K}|/(2\pi e))/2$. Given an LU decomposition for \mathbf{K} , $|\mathbf{K}|$ can be computed in $\mathcal{O}(d)$ flops. Otherwise, computing the decomposition to obtain the determinant uses $(2/3)d^3 + \mathcal{O}(d^2)$ flops. Hence, assuming that the energy term of the FE has been computed and every cluster belief has been LU decomposed, the additional cost of evaluating the entropy term of the FE is at most $(2/3)e_{\text{cg}}(kp)^3$, where e_{cg} is the number of edges of the cluster graph, if the lower-order terms are dropped. Thus, the total cost of evaluating the FE is $\mathcal{O}((v_{\text{cg}} + e_{\text{cg}})(kp)^3)$.

APPENDIX D
SIMULATION STUDY: SUPPLEMENTAL FIGURES

We provide here more details on the simulations under the most complex Müller network, for a multivariate trait ($p = 4$).

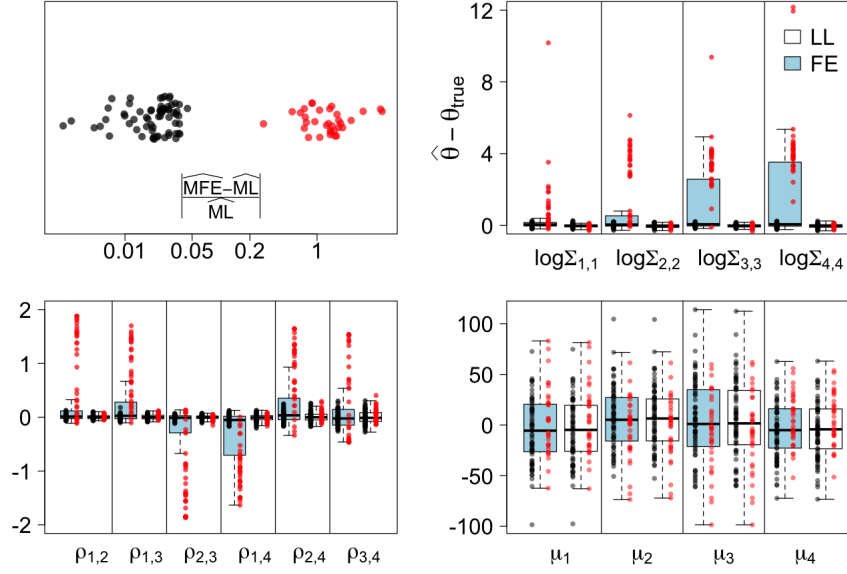


Fig. D.1. Top left: Relative deviation (on the log-scale, horizontal axis) between the maximum FE attained (\widehat{MFE}) and the theoretical maximum LL (\widehat{ML}) for the Müller network in the multivariate case, using a cluster graph with $k = 11$. The 100 replicates can be clearly divided into 67 “good” datasets (black) for which the relative deviation $|\widehat{MFE} - \widehat{ML}|/\widehat{ML}$ is below 0.05, and 33 “bad” datasets (red) for which it is large, over 0.27. Top right: estimation error from each optimization objective (LL vs FE) for the $p = 4$ variance parameters (diagonal terms in Σ), on the log-scale. Bottom: estimation error for the remaining parameters: 6 correlations ρ and 4 ancestral values μ . Within each boxplot, the 100 datasets are shown by individual points separated into the “good” (black) and “bad” (red) groups defined in the top-left panel. For the variance and correlation parameters, estimates based on numerically maximizing the FE are noticeably less precise and less accurate when the optimized FE is a poor approximation of \widehat{ML} (red points).

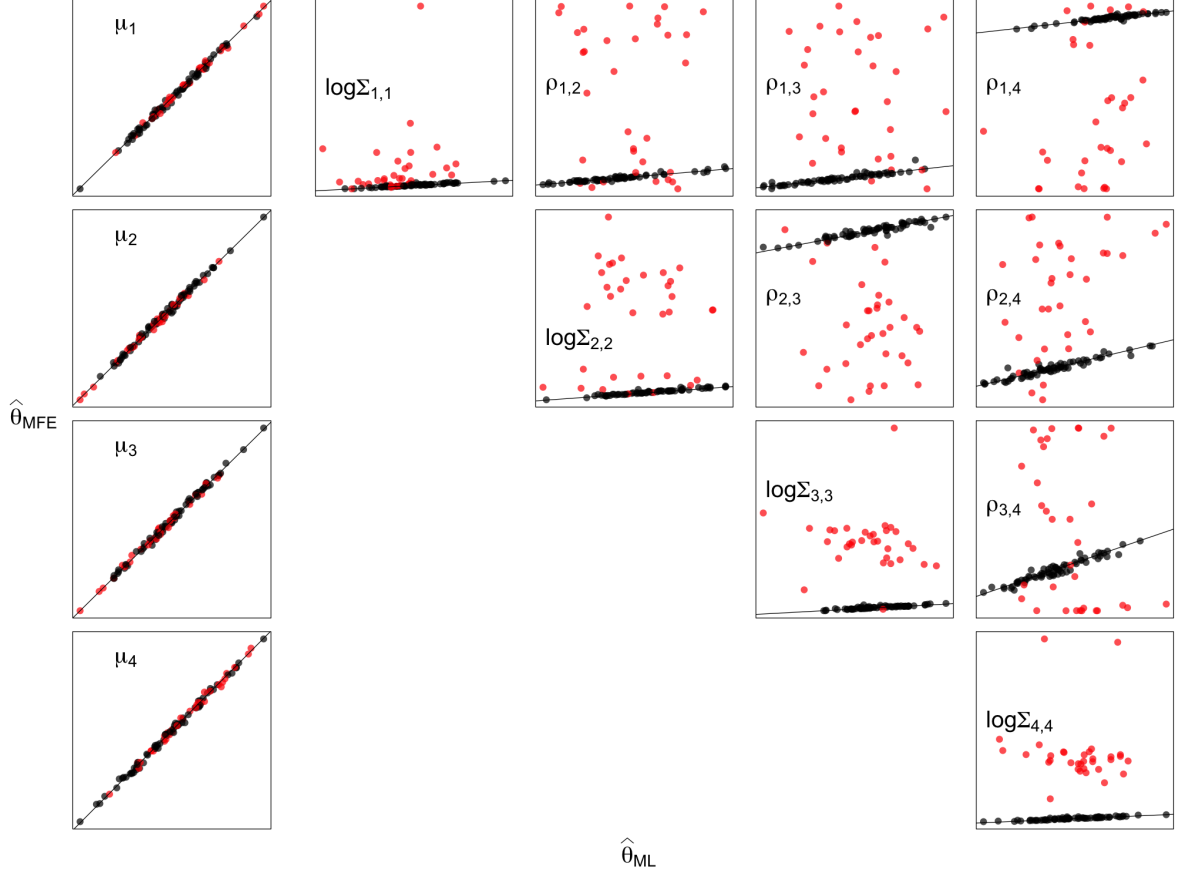


Fig. D.2. Parameter estimates for the Müller network in the multivariate case using a cluster graph with $k = 11$, estimated by either numerically maximizing the FE (vertical axis) or maximizing the exact LL (horizontal axis). Axes are not necessarily on the same scale across different plots (axes markings are suppressed for readability) but in each plot, the line corresponds to $y = x$ where both estimates are equal. Points are colored as in Fig. D.1, based on whether $|(\widehat{\text{MFE}} - \widehat{\text{ML}})/\widehat{\text{ML}}|$ is below 0.05 (black) or above 0.27 (red). The MFE estimates align well with the ML estimates in the former case, but not so in the latter case.

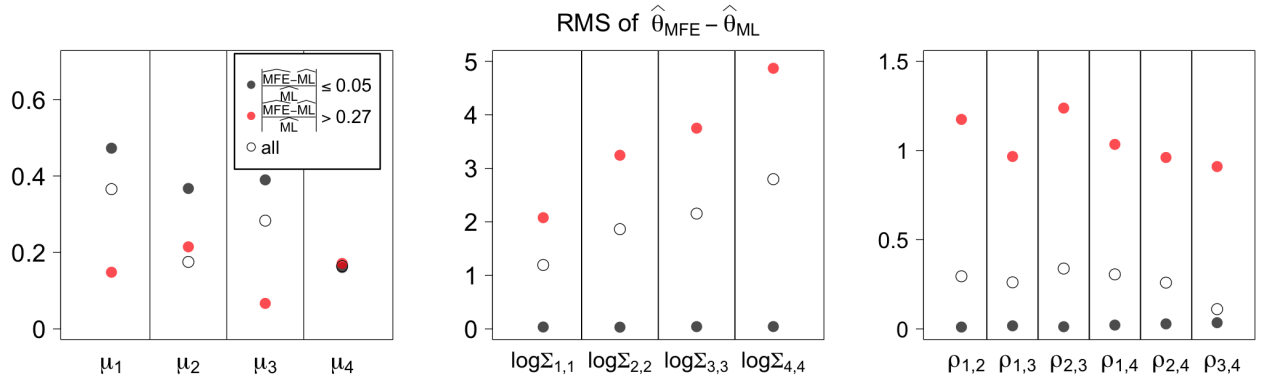


Fig. D.3. Root mean square (RMS) of the differences between the MFE ($\widehat{\theta}_{\text{MFE}}$) and ML ($\widehat{\theta}_{\text{ML}}$) estimates for the BM model parameters under the Müller network in the multivariate case, using a cluster graph with $k = 11$. $\widehat{\mu}_i$ and $\widehat{\Sigma}_{i,i}$ are the estimates of the ancestral mean and variance rate for trait i . $\rho_{i,j}$ is the evolutionary correlation between traits i and j , from Σ , and $\widehat{\rho}_{i,j}$ is its estimate. For each parameter, the RMS of $\widehat{\theta}_{\text{MFE}} - \widehat{\theta}_{\text{ML}}$ is computed from all 100 replicates, then recomputed for the “good” subset (67 replicates) and “bad” subset (33 replicates), for which $|(\widehat{\text{MFE}} - \widehat{\text{ML}})/\widehat{\text{ML}}|$ was below 0.05 or above 0.27 respectively, as in Fig. D.1. Left: for μ , the RMS difference between the two estimators is well below their RMSE (RMS between their estimate and the true value), which are all above 28. Middle: for the log variances $\log \Sigma_{i,i}$, the RMSE ranges in $[0.09, 0.11]$ for the ML estimator, and in $[1.18, 2.79]$ for the MFE estimator. The RMS difference between the two estimators is much smaller for the “good” subset than for the “bad” subset, showing that the bad subset is driving the increase in the RMSE of the FE estimator compared to the exact ML estimator. Right: for the correlations $\rho_{i,j}$, the bad subset is also driving an increase in the RMSE of the FE estimator (ranging in $[0.52, 0.73]$) compared to the exact ML estimator (whose RMSEs are in $[0.03, 0.13]$).