Expectation Propagation in Demographic Models

M. Camara Escudero[1]

[1]School of Mathematics, University of Bristol

***Abstract*—Variational methods offer a deterministic and computationally-efficient alternative to sampling methods for approximating a posterior distribution. Expectation Propagation (EP) is one such method and it is attractive due to its simplicity and intrinsically parallel computations. A major issue with EP, however, is that very little is known about its dynamic behavior and its convergence properties. Stochastic Natural-gradient Expectation Propagation (SNEP) is based on EP but it has convergence guarantees, and in this paper we explore the behavior of SNEP and Averaged-EP (aEP), a particular implementation of EP, on complex population genetic inference problems. We find that aEP outperforms SNEP in terms of the speed and reproducibility of the solution, and it rapidly learns the covariance structure. On the other hand, the aEP covariance shrinks considerably as the number of iterations increases, possibly leading to an underestimation of the tails. pSNEP, a robust version of SNEP, bridges the gap between SNEP and aEP by learning a sensible covariance structure with a much less severe underestimation of the tails.**

I. Classical Genetics Background

Every organism has measurable and observable **traits**, which depend on a combination of inherited information and the environment they live in, such as their nutrition. Inherited information is passed down as a set of instructions, collectively called the **genome**, coded via **DNA** molecules.

The DNA has a double-stranded shape [1] and it is composed by sequences of 4 basic building blocks called **nucleotides**, comprising a backbone and one of 4 **bases** A, T, C or G. Bases on one strand of DNA bond together with complementary bases on the opposite strand, forming A-T or C-G bonds.

This double-helix structure is packaged up and organized in units called **chromosomes**, along which we find regions of DNA that provide instructions on how to build a specific protein or RNA molecule, called **genes**.

Using a popular analogy among genetics textbooks, one can think of the genome as a library containing all the hereditary information of an organism. The books represent chromosomes, and their chapters are genes. Finally, one can think of the 4 bases as letters of the alphabet, with which all the information is encoded.

Humans, and most sexually-reproducing organisms, are **diploid**, meaning that their chromosomes come in pairs, called **homologous** chromosomes, having the same genes in the same places but possibly coding slightly different instructions, due to different bases at the same places, called **loci**. Carrying on the analogy, the library has 2 copies of each volume. The chapters in such copies have the same titles, but they might differ in how such chapters are written. Different variants of each gene are called **alleles**, and they account for the genetic variation underpinning evolution.

In general, there are two main ways in which different versions of a gene can arise.

- **homologous recombination**: exchange of nucleotide sequences between maternal and paternal homologous chromosomes during meiosis (summarized in Figure 1), the process of replication of reproductive cells.
- **mutation**: errors in nucleotide sequences occurring when copying DNA molecules during either meiosis or mitosis, cell division for non-reproductive cells.

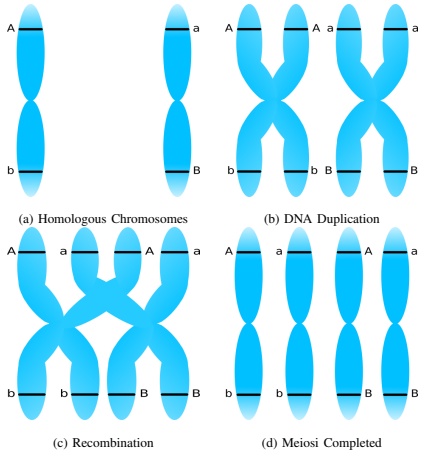

Figure 1: Recombination of a pair of homologous chromosomes to produce gametes. a) A pair of homologous chromosomes having different allele states ((A, a) or (b, B)) on different sites, marked by black lines. b) Each chromosome duplicates, obtaining two sister chromatids. c) Recombination exchanges bits of DNA across sister chromatids. d) Meiosis ends with 4 chromosomes.

II. Population Genetics Background

The mechanism of recombination illustrated above means that loci closer together on a chromosome are more likely to be inherited together than loci that are further apart. This fundamental property is called **genetic linkage** [2]. For the purpose of this paper homologous recombination will not be considered and, rather, it will be assumed that genetic diversity has arisen solely from mutation. In particular, the mutations of interest are **Single-Nucleotide Polymorphisms** (SNPs), whereby the error consists in having a different base at the same locus in a specific gene. Sticking with the analogy, one assumes that two copies of the same book might differ, in the same chapter, by a single-letter typo such as between "bear" and "beer".

The exclusion of homologous recombination, in addition to making the mathematical model simpler, is motivated by

considering a DNA region small enough for recombination to be very unlikely over relatively small time scales [3]. Furthermore, if recombination is not present, every chromosome in the offspring of the population descends from exactly one parent. For this reason, we can consider them as **haploid** organisms, where each cell has only one copy of each chromosome. Notably, this approach is very realistic when considering a specific kind of DNA molecule in humans, called **mitochondrial DNA**, whose genetic material is inherited from the mother for the vast majority, so recombination can be neglected [4].

Imagine now that a sample of $n = 5$ **haplotypes**[1] has been sampled from 5 different individuals in a population[2] of size $2N$, and that the aim is to understand the demographic history of that gene across generations. This is very useful to perform **disease gene mapping**, the study of which gene alleles contribute to individuals having particular diseases [6], [7]. One of the central ideas in population genetics is that, starting from our sample and going backwards in time, we can expect to find ancestral haplotypes that are common to all such individuals and, importantly, one of them will be the **Most Recent Common Ancestor** (MRCA).
The demographic history of the samples will be modelled using a stochastic process and we will make use of the popular **infinite sites** approximation, whereby we assume that the mutation rate is low enough, compared to the length of the sequence, that we don't need to worry about backmutations and, therefore, one and only one mutation can occur at each nucleotide of the sequence.
From this assumption one can conclude that every haplotype sampled from the population can be written as a long string of 0s and 1s, where 0 means that the sampled sequence has the same nucleotide as the MRCA, whereas 1 means that there has been a mutation, see Table I.

	Nucleotide Sequence	Encoding
Sample 1	ACT	010
Sample 2	AGT	000
Sample 3	TGT	100
Sample 4	TCA	110
Sample 5	TCT	111

Table I: Encoding of sampled haplotypes based on MRCA with sequence AGT.

III. Genealogical Trees

The relationship between the sampled sequences and the MRCA can be illustrated with a **typed ancestry** $\mathcal{A}$, as shown in Figure 2.
A typed ancestry comprises multiple elements:

- **Genealogical tree** $\mathcal{G}$ consisting of all horizontal and vertical lines, together with the times of the mutations.
- **History** (also called Genealogy) $\mathcal{H}$ the set of haplotype configurations across all generations, from the current sample, denoted H_0, all the way to the MRCA denoted H_{-m}, where m is the number of events happened up until the MRCA is achieved.

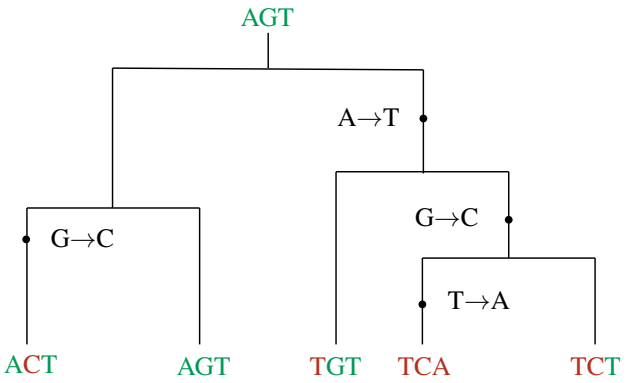

Figure 2: Typed ancestry for sampled haplotypes. Dots represent mutation events, horizontal lines are coalescence events, when considered backwards in time.

The possible events that we consider are either mutations, denoted by full dots, or **coalescence**, which are represented by horizontal lines. Going backward in time, from the current sample to the MRCA, we say that two sequences coalesce when their lineages (vertical lines) combine through a horizontal line, becoming the same sequence. A coalescent event backwards in time is equivalent to an individual having more than one offspring forward in time.
In this example, the history has 9 elements because there have been 4 mutations, 4 coalescence events and the state of the current sample

$$\mathcal{H} = \{H_0, H_{-1}, \dots, H_{-8}\},$$

where each H_j is called an **ancestral configuration**. The full list of ancestral configurations is given below, together with a flag on the side indicating whether it was a mutation or a coalescence event.

$$\begin{aligned}
H_0 &= \{ACT, AGT, TGT, TCA, TCT\} & \\
H_{-1} &= \{ACT, AGT, TGT, TCT, TCT\} & \text{m} \\
H_{-2} &= \{ACT, AGT, TGT, TCT\} & \text{c} \\
H_{-3} &= \{AGT, AGT, TGT, TCT\} & \text{m} \\
H_{-4} &= \{AGT, AGT, TGT, TGT\} & \text{m} \\
H_{-5} &= \{AGT, TGT, TGT\} & \text{c} \\
H_{-6} &= \{AGT, TGT\} & \text{c} \\
H_{-7} &= \{AGT, AGT\} & \text{m} \\
H_{-8} &= \{AGT\} & \text{c}
\end{aligned}$$

In the next few sections we will summarize ideas about how to simulate genealogies under different settings, with the goal of developing an intuition for how important certain parameters can be in studying the demographic history of a population through a sample. We will start by looking at models with discrete generations, constant population size and no mutation events, and we will then end up defining a continuous-time model with both mutation and coalescence events, with exponential population growth.

[1] sequences of loci where SNPs happen in a gene [5].
[2] We use $2N$ to compare against models of N diploids individuals.

In what follows, we will assume that all mutations are **selectively neutral**: the particular genetic sequence of a parent does not have any influence on the rate at which its descendants will develop mutations. Practically, this implies that we can consider the coalescent events and the mutation events separately. In other words, we can construct the genealogical tree first and only subsequently add mutations along the branches to obtain a full typed ancestry.

Lastly, we will assume throughout that the number of sampled individuals is much smaller than the size of the population, i.e. $n \ll N$ so that at any one time, only one coalescence event can happen.

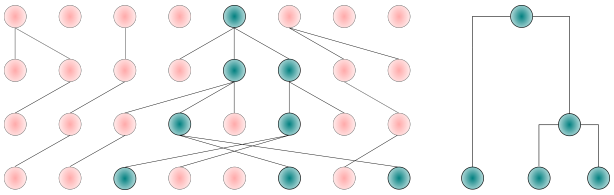

Figure 3: Example of Wright-Fisher generations for a constant population of $2N = 8$. On the left, each circle represents an individual. A row of circles represents a generation, with the oldest one at the top and the most recent one at the bottom. The bottom $n = 3$ teal circles are sampled from the population, the other teal circles represent their ancestors. On the right, a genealogical tree for the sample.

IV. Discrete-Time Coalescent Model

The Wright-Fisher model (see [8], [9] and references therein) is one of the most basic models to study the demographic history of a sample. It's main assumptions are:

- Constant population size $2N$.
- No mutation events.
- No population structure: individuals in the previous generation are equally likely to have been the parents of any descendant in the current generation.

Figure 3 shows an example of a population and a sample typical of the Wright-Fisher model, together with its genealogical tree.

Appendix A demonstrates how the random variable T_n, counting the number of generations needed for two randomly chosen individuals within a sample of size n to coalesce, follows a **geometric distribution**.

$$\mathbb{P}[T_n = t] = \left[1 - \binom{n}{2}\frac{1}{2N}\right]^{t-1}\binom{n}{2}\frac{1}{2N} \qquad t \in \mathbb{N}_+ \quad (1)$$

Discrete mutation-free genealogies can then be sampled using Algorithm 1.

Algorithm 1: Simulation of Discrete Genealogies

1: Set $k = n$
2: **while** $k > 1$ **do**
3: Draw time to next coalescence.
$$T_k \sim \text{Geom}\left(\binom{k}{2}\frac{1}{2N}\right)$$
4: Choose coalescing individuals (i, j) among $\binom{k}{2}$.
5: Merge i and j. Set $k \leftarrow k - 1$.
6: **end while**

V. Continuous Time Coalescent Model

Since the limit of a geometric distribution, as its success probability goes to 0, is an exponential distribution (see Appendix B) we can consider a continuous-time approximation to the Wright-Fisher model, the so-called **basic coalescent model**, in which time is scaled by $2N$, the population size.

$$\frac{T_n}{2N} \sim \text{Exp}\left(\binom{n}{2}\right) \quad (2)$$

As explained in Appendix B, scaling the time by $2N$ is convenient because then the time needed for a sample of size n to reach the MRCA is independent of population size. It's straightforward to generalize Algorithm 1 to the basic coalescent model by sampling from an exponential distribution with mean $\binom{k}{2}$ in step 3.

VI. Continuous-Time Model with Mutations

Consider the genealogy of a sample of n sequences as generated by the continuous-time version of Algorithm 1. We show in Appendix C that, assuming selectively neutral mutations and that N is large, on a branch of the genealogy with length (measured in scaled time) ℓ, the random variable M_ℓ counting the number of mutations along it follows a **Poisson distribution**

$$\mathbb{P}[M_\ell = m] = \frac{(\ell\theta)^m}{m!2^m}\exp\left\{-\frac{\theta}{2}\ell\right\}. \quad (3)$$

Where $\theta = 4Nu$ is the **scaled mutation rate**, and u the mutation rate. Since mutations on each branch are independent, we can use Algorithm 2 to sample continuous-time genealogies with mutations for every branch, and then distribute them randomly along such branch.

Algorithm 2: Simulation of Mutation and Coalescence

1: Use Algorithm 1 to generate genealogy of n individuals.
2: **for** every branch **do**
3: Draw number of mutations on it.
$$M_\ell \sim \text{Po}\left(\ell\frac{\theta}{2}\right) \quad \text{where } \ell \text{ branch length}$$
4: Choose timings for the M_ℓ mutations randomly on the branch.
5: **end for**

VII. Continuous-Time Exponential-Growth Model

In this section we assume that the population size is not constant across populations. Instead, we assume an exponential growth model backward in time from the current population size N_0, to a (scaled) time t_f where the population reaches a constant ancestral size N_f, i.e.

$$N_t = \begin{cases} N_0 \exp\{-\beta t\} & 0 \le t \le t_f \\ N_f & t > t_f \end{cases} \quad (4)$$

where $\beta = 4N_0 b$ is the **scaled growth rate**, b is the growth rate and t is scaled time. We can re-parametrize this expression

by defining the ratio between the current and the ancestral population size

$$r = \frac{N_0}{N_f}, \tag{5}$$

and then expressing β in terms of r and t_f

$$\beta = \frac{\log(r)}{t_f}.$$

One can then modify Algorithm 2 by shrinking or stretching the probability of coalescence at each generation. Notice how we can use r to determine whether the population has expanded ($r > 1$), contracted ($r < 1$) or maintained stable ($r = 1$). Of course, in practice we don't really have r or t_f but we will see in the next section how one can go about inferring various population parameters from the data.

VIII. Inference in Population Genetics

It should be clear that the parameters θ, r and t_f allow us to sample continuous-time genealogies with mutations where the underlying population changes exponentially. For this reason, we focus here on Bayesian inference of such parameters through $\boldsymbol{\phi}$,

$$\boldsymbol{\phi} = (\log\theta, \log r, \log t_f).$$

Our aim is to obtain a **posterior distribution** over $\boldsymbol{\phi}$ given some data $\mathcal{D} = H_0$, the haplotypes of the observed sample,

$$p(\boldsymbol{\phi} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\phi}) p(\boldsymbol{\phi}). \tag{6}$$

One of the major challenges with this approach is that in order to evaluate the likelihood we need to integrate over all the unknown ancestral configurations[3] $\mathcal{H} = \{H_{-1}, \ldots, H_{-m}\}$

$$p(\mathcal{D} \mid \boldsymbol{\phi}) = \int p(\mathcal{D} \mid \mathcal{H}, \boldsymbol{\phi}) p(\mathcal{H} \mid \boldsymbol{\phi}) d\mathcal{H} \tag{7}$$

Therefore Bayesian inference in this context requires tackling a **missing data problem**.

In order to extract more information about $\boldsymbol{\phi}$ from the data, instead of sampling a single sequence across a population, we will instead look at D **independent** regions of the DNA, and for each of them sample $n^{(d)}$ individuals from the population. This is illustrated in Figure 4.

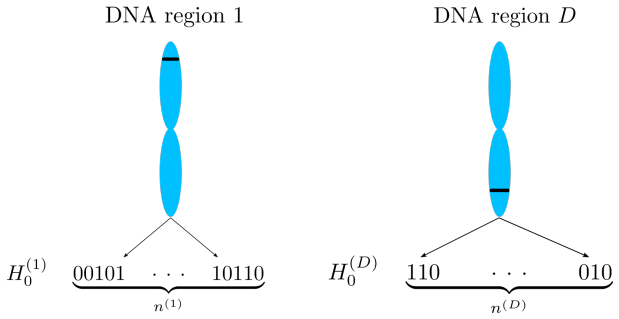

Figure 4: Illustration of the data set. There are D DNA regions of variable length, and for each of them we sample $n^{(d)}$ individuals having possibly different alleles. The set of all sample haplotypes from region d is denoted $H_0^{(d)}$.

[3]Recall that initially we defined $\mathcal{H}$ to also contain H_0 for ease of exposition. In practice, however, one uses H_0 as available data and the rest of the history as missing data or latent variables.

The full likelihood is then found by multiplying all the independent likelihood factors together

$$p(\mathcal{D} \mid \boldsymbol{\phi}) = \prod_{d=1}^{D} p(H_0^{(d)} \mid \boldsymbol{\phi}) \tag{8}$$

where $H_0^{(d)}$ is the sample of haplotypes for DNA region d and its likelihood factor is given by

$$p(H_0^{(d)}) = \int p(H_0^{(d)} \mid \mathcal{H}^{(d)}, \boldsymbol{\phi}) p(\mathcal{H}^{(d)} \mid \boldsymbol{\phi}) d\mathcal{H}^{(d)}$$

and the unknown history of sample configurations for the d^{th} DNA region is

$$\mathcal{H}^{(d)} = \left\{H_{-1}^{(d)}, \ldots, H_{-m^{(d)}}^{(d)}\right\}$$

In the next few sections we will build up the necessary theory to approximate $p(\boldsymbol{\phi} \mid \mathcal{D})$ using the Stochastic Natural-gradient Expectation Propagation (SNEP) algorithm [10]. The motivation behind considering SNEP lies in the fact that it is distributed, so that a split-data approach would take advantage of limited computational resources, and, most importantly, it promises convergence guarantees (see Section XIII), contrary to standard Expectation Propagation. In general, variational methods are deterministic methods that find an approximation to the posterior trading off the exactness of stochastic methods like MCMC for computational speed [11]. SNEP is therefore attractive, compared to more direct methods, in that while being a variational inference method, it is also convergent.

IX. Exponential Family of Distributions

The Exponential Family of Distributions is well-studied and has many appealing properties but, maybe, the most interesting fact is that it arises as the unique solution to the **Maximum Entropy** problem [12], [13]. Loosely speaking, having obtained IID observations, a member of the exponential family will maximize entropy across all distributions over the random variable that are, in some sense, "consistent" with the data. Motivated by this fact, many inference methods have been built to exploit the numerous properties of the Exponential Family, a very rigorous and thorough treatment can be found in [12].

A random vector Φ follows a distribution in the exponential family if it can be written as

$$p_{\boldsymbol{\vartheta}}(\boldsymbol{\phi}) = \exp\left\{\langle \boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi})\rangle - A(\boldsymbol{\vartheta})\right\}$$

where $\boldsymbol{\vartheta}$ is the **natural parameter**, $\boldsymbol{\rho}(\boldsymbol{\phi})$ is a **sufficient statistics** of $\boldsymbol{\phi}$, $\langle\cdot\rangle$ denotes an inner product, and $A(\boldsymbol{\vartheta})$ is called **log-partition function**

$$A(\boldsymbol{\vartheta}) = \log \int \exp\left\{\langle \boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi})\rangle\right\} d\boldsymbol{\phi}$$

and is the logarithm of the normalization constant of the distribution. Below we give a brief summary of the main characterization of the Exponential Family.

1) Sufficient Statistics: $\boldsymbol{\rho}(\boldsymbol{\phi})$ determines the distributional form of the particular pdf parametrized by $\boldsymbol{\vartheta}$. In other words if $\rho(\boldsymbol{\phi}) = (\boldsymbol{\phi}, \text{vec}(\boldsymbol{\phi}\boldsymbol{\phi}^\top))$ are the sufficient statistics of the multivariate normal (see Appendix D), then any multivariate normal distribution can be written with the same sufficient statistics but with different parameters $\boldsymbol{\vartheta}$ determining the mean and the variance-covariance matrix.

2) Mean Parametrization: The sufficient statistics also give us access to an alternative parametrization of the exponential family in terms of **mean parameters**[4], defined as

$$\boldsymbol{\eta} = \mathbb{E}_p[\boldsymbol{\rho}(\boldsymbol{\phi})] \tag{9}$$

Computing the mean parameters from the natural parameters is called **forward mapping** and, as we show in Appendix E, can be performed by taking the gradient of the log-partition function

$$\boldsymbol{\vartheta} \xrightarrow{\nabla_{\boldsymbol{\vartheta}} A} \boldsymbol{\eta}$$

The mapping from mean parameters to natural parameters is called **backward mapping** and can be performed using the convex conjugate $A^*(\boldsymbol{\eta})$ of $A(\boldsymbol{\vartheta})$

$$\boldsymbol{\vartheta} \xleftarrow{\nabla_{\boldsymbol{\eta}} A^*} \boldsymbol{\eta} \tag{10}$$

3) Convex Conjugacy: The log-partition function $A(\boldsymbol{\vartheta})$ is a **convex** function of $\boldsymbol{\vartheta}$ so it admits a convex conjugate [15]. This is found by solving an optimization problem, called the **dual** problem

$$A^*(\boldsymbol{\eta}) = \sup_{\boldsymbol{\vartheta} \in \boldsymbol{\Theta}} \boldsymbol{\vartheta}^\top \boldsymbol{\eta} - A(\boldsymbol{\vartheta}),$$

where $\boldsymbol{\Theta}$ is the natural parameter domain: the set of $\boldsymbol{\vartheta}$ for which the log partition is finite. Conversely, the convex conjugate of $A^*(\boldsymbol{\eta})$ is $A(\boldsymbol{\vartheta})$ and it is the solution of

$$A(\boldsymbol{\vartheta}) = \sup_{\boldsymbol{\eta} \in \boldsymbol{\mathcal{M}}} \boldsymbol{\eta}^\top \boldsymbol{\vartheta} - A^*(\boldsymbol{\eta}) \tag{11}$$

where $\boldsymbol{\mathcal{M}}$ is the mean parameter domain. Equation (11) is known as **variational** problem, and it's solution is found at the mean parameters corresponding to $\boldsymbol{\vartheta}$

$$\boldsymbol{\eta} = \mathbb{E}_{p_{\boldsymbol{\vartheta}}}[\boldsymbol{\rho}(\boldsymbol{\phi})].$$

Practically, this means that by solving (11) one can obtain the log-partition function $A(\boldsymbol{\vartheta})$ (and thus, the normalization constant) and the mean parameters $\boldsymbol{\eta}$ of $p_{\boldsymbol{\vartheta}}$, which can be converted to natural parameters using the forward mapping. This will be the driving idea behind the SNEP algorithm described in Section XIII, where we will relax the variational problem to make it more tractable.

4) Products and Ratios: Suppose that we have two members of the exponential family with the same sufficient statistics but different natural parameters

$$p_{\boldsymbol{\vartheta}_1}(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_1, \boldsymbol{\rho}(\boldsymbol{\phi})) \qquad p_{\boldsymbol{\vartheta}_2}(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_2, \boldsymbol{\rho}(\boldsymbol{\phi}))$$

[4]Notice that the parametrization in terms of $\boldsymbol{\vartheta}$ or $\boldsymbol{\eta}$ is unique only if the exponential family is *minimal* [14], [12]. In practice, we always assume that this is the case.

Then, under certain conditions, their product has the same distributional form and its parameter is given by the sum of the respective natural parameters

$$p_{\boldsymbol{\vartheta}_1}(\boldsymbol{\phi}) p_{\boldsymbol{\vartheta}_2}(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_1 + \boldsymbol{\vartheta}_2, \boldsymbol{\rho}(\boldsymbol{\phi})). \tag{12}$$

Similarly, their ratio, under certain non-trivial conditions, has the same distributional form with natural parameter given by the difference of the respective parameters

$$\frac{p_{\boldsymbol{\vartheta}_1}(\boldsymbol{\phi})}{p_{\boldsymbol{\vartheta}_2}(\boldsymbol{\phi})} = \text{EF}(\boldsymbol{\vartheta}_1 - \boldsymbol{\vartheta}_2, \boldsymbol{\rho}(\boldsymbol{\phi})). \tag{13}$$

Properties (12) and (13) are exploited in Expectation Propagation.

X. Expectation Propagation

Suppose that our aim is to approximate the posterior distribution $p(\boldsymbol{\phi} \mid \mathcal{D})$ in equation (6).

$$p(\boldsymbol{\phi} \mid \mathcal{D}) \propto \prod_{d=1}^{D} p(H_0^{(d)} \mid \boldsymbol{\phi}) p(\boldsymbol{\phi})$$

MCMC sampling would seem to be a sensible choice. However, to find a single likelihood term $p(H_0^{(d)} \mid \boldsymbol{\phi})$ we need to integrate over all possible histories $\mathcal{H}^{(d)}$ (see Equation (7)), each of them having a (possibly) exponentially large number of ancestral configurations. As such, the full likelihood $p(\mathcal{D} \mid \boldsymbol{\phi})$ is very intractable and MCMC would likely be inefficient.

Expectation Propagation assumes that, while the full likelihood $p(\mathcal{D} \mid \boldsymbol{\phi})$ is too intractable to be tackled altogether, individual likelihood terms $p(H_0^{(d)} \mid \boldsymbol{\phi})$ are more tractable. The strategy of EP is then to approximate each likelihood term in turn with the hope that this will lead to a good global approximation.

EP was initially introduced in [16] and since then it has gained a lot of popularity due to its intrinsic simplicity and effectiveness in tackling hard problems. In spite of its traction, there have been very few theoretical results regarding its convergent properties [17], although some promising work has shown that its dynamic behavior matches that of Newton's method in the limit of infinite data [18].

EP works by considering a posterior distribution that can be factorized into a product of $D+1$ terms

$$p(\boldsymbol{\phi} \mid \mathcal{D}) \propto \prod_{d=0}^{D} f_d(\boldsymbol{\phi})$$

where, in this case, one factor will be the prior and the remaining ones, called **sites**, will be likelihood terms

$$f_d(\boldsymbol{\phi}) = \begin{cases} p(\boldsymbol{\phi}) & \text{if } d = 0 \\ p(H_0^{(d)} \mid \boldsymbol{\phi}) & \text{if } d \in \{1, \dots, D\}. \end{cases}$$

It then introduces a **global approximation** to the posterior that has a similar factorization structure

$$q(\boldsymbol{\phi}) \propto \prod_{d=0}^{D} g_d(\boldsymbol{\phi}), \tag{14}$$

where one factor will be the prior, as before, and the remaining ones, called **site approximations**, will be members of the exponential family with the same sufficient statistics but with different natural parameters

$$g_d(\boldsymbol{\phi}) = \begin{cases} p(\boldsymbol{\phi}) & \text{if } d = 0 \\ \text{EF}(\boldsymbol{\vartheta}_d, \boldsymbol{\rho}(\boldsymbol{\phi})) & \text{if } d \in \{1, \dots, D\}. \end{cases}$$

As it was already hinted at before, EP then proceeds by iteratively cycling through each site $f_1(\boldsymbol{\phi}), \dots, f_D(\boldsymbol{\phi})$ and approximate it while keeping the other ones fixed, in a way that is reminescent of coordinate descent optimization. Specifically, to approximate site $f_j(\boldsymbol{\phi})$, with $1 \leq j \leq D$, it proceeds by taking the full global approximation and replacing $g_j(\boldsymbol{\phi})$ with $f_j(\boldsymbol{\phi})$. The aim here is to obtain a hybrid distribution that is easier to approximate than the full posterior $p(\boldsymbol{\phi} \mid \mathcal{D})$. The construction of this hybrid distribution, called **tilted distribution**, follows two steps. Firstly, it uses property (13) to remove $g_j(\boldsymbol{\phi})$ from the global approximation, forming the so-called **cavity distribution**

$$q_{-j}(\boldsymbol{\phi}) \propto \prod_{\substack{d=0 \\ d \neq j}}^{D} g_d(\boldsymbol{\phi}).$$

Secondly, it uses the cavity distribution as a "prior" and multiplies it by the possibly intractable likelihood term that we want to approximate

$$q_{/j}(\boldsymbol{\phi}) \propto f_j(\boldsymbol{\phi}) q_{-j}(\boldsymbol{\phi}). \tag{15}$$

By assumption, the tilted distribution in equation (15) is more tractable than the full posterior and we can use a variational approach to approximate it.

We now aim to update $g_j(\boldsymbol{\phi})$ to some new term $g_j^{\text{new}}(\boldsymbol{\phi})$ so that the new global approximation $g_j^{\text{new}}(\boldsymbol{\phi}) q_{-j}(\boldsymbol{\phi})$ is closest, in some sense, to the tilted distribution. In Variational Inference there are two main methods for measuring distance between a target distribution and an approximation, both defined in terms of the KL divergence between two distributions p_1 and p_2

$$\text{KL}\left(p_1 \parallel p_2\right) = \int p_1(\boldsymbol{\phi}) \log\left(\frac{p_1(\boldsymbol{\phi})}{p_2(\boldsymbol{\phi})}\right) d\boldsymbol{\phi}.$$

Inference via Variational Message Passing (VMP) minimizes the KL divergence shown below [19]

$$q_{\text{vb}}^{\text{new}}(\boldsymbol{\phi}) = \underset{q \in \mathcal{Q}}{\arg\min}\ \text{KL}\left[g_j^{\text{new}}(\boldsymbol{\phi}) q_{-j}(\boldsymbol{\phi}) \parallel q_{/j}(\boldsymbol{\phi})\right]$$

while EP minimizes the reverse KL divergence.

$$q_{\text{ep}}^{\text{new}}(\boldsymbol{\phi}) = \underset{q \in \mathcal{Q}}{\arg\min}\ \text{KL}\left[q_{/j}(\boldsymbol{\phi}) \parallel g_j^{\text{new}}(\boldsymbol{\phi}) q_{-j}(\boldsymbol{\phi})\right] \tag{16}$$

The differences between these two divergence measures are reflected in the differences between the approximations that the two algorithms will lead to. In particular, VMP avoids regions where the true distribution $q_{/j}(\boldsymbol{\phi})$ has little mass, whereas EP seeks to cover all the regions where $q_{/j}(\boldsymbol{\phi})$ has any positive mass. This results in the former focusing on one mode, depending on initialization, and underestimating the tails, while the latter finds an approximation that covers all the modes, but is more spread out and has thicker tails [20].

Algorithm 3: Expectation Propagation

Initialize prior.

$$p(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_0, \boldsymbol{\rho}(\boldsymbol{\phi}))$$

Initialize site approximations.

$$g_d^{(1)}(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_d^{(1)}, \boldsymbol{\rho}(\boldsymbol{\phi})) \quad \forall\, d = 1, \dots, D$$

Initialize global approximation.

$$q^{(1)}(\boldsymbol{\phi}) = \text{EF}\left(\boldsymbol{\vartheta}_0 + \sum_{d=1}^{D} \boldsymbol{\vartheta}_d^{(1)}, \boldsymbol{\rho}(\boldsymbol{\phi})\right)$$

Choose damping $\delta \in (0, 1]$.

for every iteration $i = 1, \dots, n_{\text{sweeps}}$ **do**

  for every site $j = 1, \dots, D$ **do**

    Form cavity distribution.

$$q_{-j}^{(i)}(\boldsymbol{\phi}) = \text{EF}\left(\boldsymbol{\vartheta}_0 + \sum_{\substack{d=1 \\ d \neq j}}^{D} \boldsymbol{\vartheta}_d^{(i)}, \boldsymbol{\rho}(\boldsymbol{\phi})\right)$$

    Form tilted distribution.

$$q_{/j}^{(i)}(\boldsymbol{\phi}) \propto f_j(\boldsymbol{\phi}) q_{-j}^{(i)}(\boldsymbol{\phi})$$

    Find new global mean parameters by matching moments.

$$\boldsymbol{\eta}^{(i+1)} \longleftarrow \mathbb{E}_{q_{/j}^{(i)}(\boldsymbol{\phi})}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$$

    Convert mean parameters to natural parameters.

$$\boldsymbol{\vartheta}^{(i+1)} \longleftarrow \nabla_{\boldsymbol{\eta}^{(i+1)}} A^*(\boldsymbol{\eta}^{(i+1)})$$

    Find new natural parameter for the site.

$$\boldsymbol{\vartheta}_j^{(i+1)} \longleftarrow (1 - \delta)\boldsymbol{\vartheta}_j^{(i)} + \delta\left[\boldsymbol{\vartheta}^{(i+1)} - \boldsymbol{\vartheta}_{-j}^{(i)}\right]$$

  end for

end for

It turns out that using the reverse **KL divergence** as a measure of distance leads to a relatively simple solution: the global approximation with factorization (14) that minimizes the reverse KL-divergence to the tilted distribution is found by **moment-matching**.

$$\mathbb{E}_{q^{\text{new}}(\boldsymbol{\phi})}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right] = \mathbb{E}_{q_{/j}(\boldsymbol{\phi})}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$$

That is, to find the new natural parameter $\boldsymbol{\vartheta}^{\text{new}}$ for the global approximation we first take the expectation of the sufficient statistics $\boldsymbol{\rho}(\boldsymbol{\phi})$ with respect to the tilted distribution $q_{/j}(\boldsymbol{\phi})$. Then we assign this value to the new mean parameters of the global approximation $\boldsymbol{\eta}^{\text{new}} = \mathbb{E}_{q^{\text{new}}(\boldsymbol{\phi})}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$. Finally, we use the backward mapping in equation (10) to convert the mean parameter to the natural parameter $\boldsymbol{\vartheta}^{\text{new}}$.

Since each likelihood term is intractable, moment-matching often reduces to obtaining samples $\boldsymbol{\phi}^{[1]}, \dots, \boldsymbol{\phi}^{[m]}$ from the tilted distribution, approximating the global mean parameters

with via Monte Carlo

$$\boldsymbol{\eta}^{\text{new}} \longleftarrow \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\rho}(\boldsymbol{\phi}^{[i]}),$$

and finally converting the mean parameters to natural parameters

$$\boldsymbol{\vartheta}^{\text{new}} \longleftarrow \nabla_{\boldsymbol{\eta}^{\text{new}}} A^*(\boldsymbol{\eta}^{\text{new}}).$$

Cycling through every site $j = 1, \dots, D$ is called a sweep through the sites and, normally, multiple sweeps are needed in order to achieve convergence. Algorithm 3 summarizes Expectation Propagation for $i = 1, \dots, n_{\text{sweeps}}$ sweeps through the sites, and the superscript $^{(i)}$ indicates that we are considering distributions at the i^{th} sweep.
Notice that if we choose a prior distribution of the form

$$p(\boldsymbol{\phi}) = \text{EF}(\boldsymbol{\vartheta}_0, \boldsymbol{\rho}(\boldsymbol{\phi}))$$

then using property (12) and definition (14) the global approximation becomes

$$q(\boldsymbol{\phi}) = \text{EF}\left(\sum_{d=0}^{D}\boldsymbol{\vartheta}_d, \boldsymbol{\rho}(\boldsymbol{\phi})\right) := \text{EF}(\boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi}))$$

where we have defined $\boldsymbol{\vartheta}$ to be the sum of all site approximation and prior natural parameters. Similarly, the cavity will be found by subtracting the j^{th} natural parameter.

$$q_{-j}(\boldsymbol{\phi}) = \text{EF}\left(\boldsymbol{\vartheta} - \boldsymbol{\vartheta}_j, \boldsymbol{\rho}(\boldsymbol{\phi})\right)$$

When the likelihood is particularly ill-behaved, it can be helpful to use some **damping** [17], [10] by choosing a value $\delta \in (0, 1]$ and update the natural parameter at site j and iteration i as

$$\boldsymbol{\vartheta}_j^{(i+1)} \longleftarrow (1-\delta)\boldsymbol{\vartheta}_j^{(i)} + \delta\left[\boldsymbol{\vartheta}^{(i+1)} - \boldsymbol{\vartheta}_0 - \sum_{\substack{d=1\\ d\neq j}}^{D}\boldsymbol{\vartheta}_d^{(i)}\right]$$

Essentially the equation above can be read as follows: to update the j^{th} site approximation subtract the current cavity parameter from the new global parameter to get the raw update, then take a weighted average with the previous site parameter using δ and $(1-\delta)$ as weights.
In Table II we summarize the natural and mean parametrization of the various terms involved in the EP algorithm.

	Natural	Mean
j^{th} Site	$\boldsymbol{\vartheta}_j$	$\boldsymbol{\eta}_j$
j^{th} Cavity	$\boldsymbol{\vartheta}_{-j}$	$\boldsymbol{\eta}_{-j}$
Global	$\boldsymbol{\vartheta}$	$\boldsymbol{\eta}$

Table II: Natural and Mean parameters for the j^{th} site approximation, the j^{th} cavity and the global approximation. Subscripts $^{(i)}$ indicating the iteration number have been omitted in this table to simplify notation.

XI. Averaged Expectation Propagation

One of the issues with EP is that as the number of sites D grows, one has to keep in memory a large number of natural parameters $\boldsymbol{\vartheta}_1, \dots, \boldsymbol{\vartheta}_D$. To decrease the parameter space and therefore obtain a memory saving of order D, a generalization called Stochastic EP [21] has been proposed. In this paper, we will consider a specific case of Stochastic EP, **averaged Expectation Propagation** (aEP) [18], that was previously developed by Dehaene and Barthelmé.
This simplified version sets all the site approximations to have the same natural parameter, i.e. a fraction of the global natural parameter

$$\boldsymbol{\vartheta}_j = \frac{1}{D}\boldsymbol{\vartheta} \qquad \text{for } j = 1, \dots, D$$

consequently the cavity distribution has parameter

$$\boldsymbol{\vartheta}_{-j} = \boldsymbol{\vartheta}_0 + \frac{D-1}{D}\boldsymbol{\vartheta} \qquad j = 1, \dots, D \tag{17}$$

This simplified version is more computationally efficient and sometimes can lead to qualitatively similar results to classic EP (see Section 6 in [18]), so we will compare this memory-efficient version of EP, using an particle-based sampling algorithm to perform moment-matching, against the SNEP algorithm, introduced in Section XIII.

XII. Power Expectation Propagation

Another extension to standard EP consists in replacing the KL divergence in equation (16) with the α-divergence [22]

$$D_\alpha\left(q_{/j} \parallel q^{\text{new}}\right) = \frac{4}{1-\alpha}\int q_{/j}(\boldsymbol{\phi})^{\frac{1+\alpha}{2}} q^{\text{new}}(\boldsymbol{\phi})^{\frac{1-\alpha}{2}} d\boldsymbol{\phi}, \tag{18}$$

where we have dropped the dependency on $\boldsymbol{\phi}$ and defined $q^{\text{new}} = g_j^{\text{new}}(\boldsymbol{\phi})q_{-j}(\boldsymbol{\phi})$ for ease of notation.Then, in a similar way as to EP, one can formulate a fixed-point algorithm that at each iteration minimizes α-divergence between the tilted distribution and the new global approximation, obtaining **power EP** (pEP) [23]. The α-divergence is a generalization of the KL-divergence seen above and it can be equivalent to both EP or VMP depending on the value of α.

$$D_\alpha\left(q_{/j} \parallel q^{\text{new}}\right) = \begin{cases}\text{KL}\left(q_{/j} \parallel q^{\text{new}}\right) & \text{if } \alpha = 1\\ \text{KL}\left(q^{\text{new}} \parallel q_{/j}\right) & \text{if } \alpha = -1\end{cases}$$

It leads to a novel algorithm when $\alpha \neq \{1, -1\}$, and it can be obtained by re-parametrizing the α-divergence at each locus

$$\gamma_j = \frac{1+\alpha_j}{2}$$

and setting to zero the derivative of (18). The resulting algorithm can be viewed as standard EP applied to $\frac{1}{\gamma_j}$-rooted sites, since usually $\gamma_j \in (0, 1]$. For instance, when computing the cavity distribution, rather than removing the j^{th} site completely, we remove it's $\frac{1}{\gamma_j}$-root

$$q_{-j}(\boldsymbol{\phi}) \propto \frac{q(\boldsymbol{\phi})}{g_j(\boldsymbol{\phi})^{\gamma_j}} = \left[\prod_{\substack{d=0\\ d\neq j}}^{D} g_d(\boldsymbol{\phi})\right] g_j(\boldsymbol{\phi})^{1-\gamma_j}$$

which translates to changing less the global parameter when obtaining the parameter for the cavity distribution

$$\boldsymbol{\vartheta}_{-j} = \boldsymbol{\vartheta} - \gamma_j \boldsymbol{\vartheta}_j$$

Similarly, the tilted distribution is found by replacing the $\frac{1}{\gamma_j}$-root of $g_j(\boldsymbol{\phi})$ with the $\frac{1}{\gamma_j}$-root of the likelihood site.

$$q_{/j} \propto f_j(\boldsymbol{\phi})^{\gamma_j} q_{-j}(\boldsymbol{\phi})$$

thus introducing a possibly less intractable term since on can match the exponent γ_j to cancel out any exponents in the expression of the likelihood factor, making the algorithm more robust and stable [23]. As a consequence, when we estimate the natural parameter of the tilted distribution $\boldsymbol{\vartheta}^{\text{new}}$ by matching moments, we use this to find the new natural parameter for the $\frac{1}{\gamma_j}$-rooted site

$$\gamma_j \boldsymbol{\vartheta}_j^{\text{new}} \longleftarrow \boldsymbol{\vartheta}^{\text{new}} - \boldsymbol{\vartheta}_{-j}$$

and then we replicate this update, in proportion, to the whole site approximation.

$$\boldsymbol{\vartheta}_j^{\text{new}} \longleftarrow \frac{1}{\gamma_j} \left[\boldsymbol{\vartheta}^{\text{new}} - \boldsymbol{\vartheta}_{-j}\right]$$

A complete description of pEP is found in Algorithm 4, where notation is consistent with Table II.

Algorithm 4: Power EP

Initialize prior natural parameter $\boldsymbol{\vartheta}_0$

Initialize site approximations natural parameters $\boldsymbol{\vartheta}_j^{(1)}$ for $j \in \{1, \ldots, D\}$.

Initialize global approximation natural parameter

$$\boldsymbol{\vartheta}^{(1)} \longleftarrow \boldsymbol{\vartheta}_0 + \sum_{j=1}^{D} \boldsymbol{\vartheta}_j^{(1)}$$

Choose $\delta \in (0, 1]$, and powers $\gamma_1, \ldots, \gamma_D \in (0, 1]$.

for every iteration $i = 1, \ldots, n_{\text{sweeps}}$ **do**

  for every site $j = 1, \ldots, D$ **do**

    Form cavity distribution natural parameters

$$\boldsymbol{\vartheta}_{-j}^{(i)} \longleftarrow \boldsymbol{\vartheta}^{(i)} - \gamma_j \boldsymbol{\vartheta}_j^{(i)}$$

    Form tilted distribution.

$$q_{/j}^{(i)}(\boldsymbol{\phi}) \propto f_j(\boldsymbol{\phi})^{\gamma_j} q_{-j}(\boldsymbol{\phi})$$

    Find new global mean parameters by matching moments.

$$\boldsymbol{\eta}^{(i+1)} \longleftarrow \mathbb{E}_{q_{/j}^{(i)}(\boldsymbol{\phi})} \left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$$

    Convert mean parameters to natural parameters.

$$\boldsymbol{\vartheta}^{(i+1)} \longleftarrow \nabla_{\boldsymbol{\eta}^{(i+1)}} A^*(\boldsymbol{\eta}^{(i+1)})$$

    Find new natural parameter for the site.

$$\boldsymbol{\vartheta}_j^{(i+1)} \longleftarrow (1-\delta)\boldsymbol{\vartheta}_j^{(i)} + \frac{\delta}{\gamma_j}\left[\boldsymbol{\vartheta}^{(i+1)} - \boldsymbol{\vartheta}_{-j}^{(i)}\right]$$

  end for

end for

XIII. Stochastic Natural-Gradient Expectation Propagation

The problematic step in EP, aEP and pEP is moment-matching because all three of them are fixed-point algorithms and therefore they are not very robust against the stochastic noise due to MCMC sampling [10]. Accordingly, often one needs to obtain a prohibitively large number of samples to suitably approximate the moments of the tilted distribution. Many alternative ways have been proposed to simplify the moment-matching step, such as approximating the tilted distribution with a Laplace approximation [24]. In this paper, however, we will present a different algorithm that is suitable for distributed Bayesian learning: **Stochastic Natural-gradient Expectation Propagation** (SNEP) [10].

In order to develop the SNEP algorithm one needs to define the variational problem corresponding to pEP. Recall that, for a general member of the exponential family $\text{EF}(\boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi}))$ one can define the variational problem associated with the log-partition function, as seen in equation (11)

$$A(\boldsymbol{\vartheta}) = \sup_{\boldsymbol{\eta} \in \boldsymbol{\mathcal{M}}} \boldsymbol{\eta}^\top \boldsymbol{\vartheta} - A^*(\boldsymbol{\eta}).$$

The key here is to view the posterior distribution in (6) as a member of the **extended exponential family** of distributions, so that it is then possible to define an "extended" variational problem similar to the one above. This new family of distributions is essentially just a notational trickery used to write the posterior in a form that resembles that of an exponential family distribution. Before going ahead and writing the posterior distribution in that form, we imagine that our data set $\mathcal{D}$ is partitioned into $i = 1, \ldots, N_{\text{workers}}$ **workers**

$$\mathcal{D}_i := \left\{ H_0^{(k)} : k \in \mathcal{P}_i \subseteq \{1, \ldots, D\} \right\}$$

$$\{1, \ldots, D\} := \bigcup_{i=1}^{N_{\text{workers}}} \mathcal{P}_i \quad \text{and} \quad \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \forall\, i \neq j.$$

This split of the data is reminiscent of our original factorization assumption in EP, but it generalizes so that now at each site the log-likelihood is a sum of terms

$$\ell_i(\boldsymbol{\phi}) := \sum_{k \in \mathcal{P}_i} \log p(H_0^{(k)} \mid \boldsymbol{\phi}).$$

This is quite useful for developing a general distributed algorithm that can work with a number of computer nodes that is different than the number of likelihood terms.

Coming back to the posterior distribution, we re-write it as follows

$$\begin{aligned} p(\boldsymbol{\phi} \mid \mathcal{D}) &\propto \exp\left\{ \sum_{i=1}^{N_{\text{workers}}} \ell_i(\boldsymbol{\phi}) \right\} p(\boldsymbol{\phi}) \\ &\propto \exp\left\{ \sum_{i=1}^{N_{\text{workers}}} \ell_i(\boldsymbol{\phi}) \right\} \exp\left\{ \boldsymbol{\vartheta}_0^\top \boldsymbol{\rho}(\boldsymbol{\phi}) \right\} \\ &\propto \exp\left\{ \widetilde{\boldsymbol{\vartheta}}^\top \widetilde{\boldsymbol{\rho}}(\boldsymbol{\phi}) \right\}, \end{aligned}$$

where we have defined the following terms

$$\widetilde{\boldsymbol{\vartheta}} := [\boldsymbol{\vartheta}_0, \mathbf{1}]^\top$$
$$\widetilde{\boldsymbol{\rho}}(\boldsymbol{\phi}) := [\boldsymbol{\rho}(\boldsymbol{\phi}), \ell_1(\boldsymbol{\phi}), \dots, \ell_{N_{\text{workers}}}(\boldsymbol{\phi})]^\top$$
$$\mathbf{1} := (1, \dots, 1)^\top \in \mathbb{R}^{N_{\text{workers}}}.$$

Essentially we have absorbed the likelihood chunks $\ell_i(\boldsymbol{\phi})$ into the sufficient statistics each with natural parameter 1, so that we can write the corresponding (approximate) variational problem

$$\widetilde{A}(\widetilde{\boldsymbol{\vartheta}}) = \max_{\widetilde{\boldsymbol{\eta}} \in \widetilde{\boldsymbol{\mathcal{M}}}} \widetilde{\boldsymbol{\eta}}^\top \widetilde{\boldsymbol{\vartheta}} - \widetilde{A}^*(\widetilde{\boldsymbol{\eta}}). \tag{19}$$

Here $\widetilde{\boldsymbol{\mathcal{M}}}$ is some outer bound on the extended mean domain[5], for more details see the original paper [10] and [12]. The approximate variational problem above is a constrained maximization[6] problem and one can introduce **Lagrange multipliers** to solve it, thus obtaining the power EP updates, as seen in Section XII, where it turns out that each Lagrange multiplier corresponds to the natural parameter of a likelihood chunk.

To go from pEP to SNEP, we follow the EM observation in Appendix G and we aim to construct an auxiliary problem whose solution is the same as (19) and that can be solved using **coordinate maximization**. Accordingly, we introduce $i = 1, \dots, N_{\text{workers}}$ auxiliary variables $\boldsymbol{\vartheta}'_i$ each one representing the natural parameter of global approximation kept by each worker locally, whose role will be clearer later, which we call **local-global approximation**. Then, for each worker, we subtract from the objective function in (19) the **KL divergence** between a the corresponding local tilted distribution and the local-global approximation.

Overall, the **auxiliary variational problem** obtained can be solved, similarly to EM, by coordinate maximization. In practical terms this means that we alternate between the following two steps:

- keeping the natural parameters of the local-global approximation fixed, while computing the moments of the local tilted distribution.
- updating the natural parameters of the local-global approximation to perform moment matching.

The first step is implemented by leveraging convex duality and solving the dual using Lagrange multipliers. To minimize with respect to the Lagrange multipliers the authors use **natural stochastic gradient** descent rather than Euclidean stochastic gradient descent. This is convenient because while it improves performance, it is also easy to implement as it's just a matter of reparametrization [10].

Practically, the algorithm can be implemented in a **distributed** way by letting the different workers learn the natural parameter of the local-global approximation and then synchronizing with a master node, called **posterior server** by sharing with it their most recent natural parameter for the likelihood chunk approximation.

[5]Since they are outer bounds, we can replace sup with max in the optimization problem.

[6]Unfortunately, since it's an approximation, it is not guaranteed to be concave.

In a Multivariate Gaussian setting, the algorithm starts by choosing a mean vector $\boldsymbol{\mu}_0$ and a variance-covariance matrix $\boldsymbol{\Sigma}_0$ for the prior and transforming them into natural parameters

$$\boldsymbol{\vartheta}_0 = \begin{pmatrix} \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_0^{-1}) \end{pmatrix}$$

Similarly, it chooses Gaussian parameters for each worker, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$, and transforms them into natural and mean parameters.

$$\boldsymbol{\vartheta}_j^{(1)} = \begin{pmatrix} \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_j^{-1}) \end{pmatrix} \qquad \boldsymbol{\eta}_j^{(1)} = \begin{pmatrix} \boldsymbol{\mu}_j \\ \text{vec}(\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j\boldsymbol{\mu}_j^\top) \end{pmatrix}$$

The natural parameter $\boldsymbol{\vartheta}_j^{(1)}$ is then also stored in a different variable, $\boldsymbol{\vartheta}_j^{\text{old}}$ and it will be used for synchronization later on. These parameters are then combined together to obtain a natural parameter for the cavity distribution and an auxiliary natural parameter $\boldsymbol{\vartheta}'_j$, representing the local-global approximation, for each worker.

$$\boldsymbol{\vartheta}_{-j} = \boldsymbol{\vartheta}_0 + \sum_{\substack{\mathfrak{z}=1 \\ \mathfrak{z}\neq j}}^{N_{\text{workers}}} \boldsymbol{\vartheta}_{\mathfrak{z}}^{(1)} \qquad \text{and} \qquad \boldsymbol{\vartheta}'_j = \sum_{\mathfrak{z}=1}^{N_{\text{workers}}} \boldsymbol{\vartheta}_{\mathfrak{z}}^{(1)}$$

Finally, each worker initializes the starting point of a Markov Chain by sampling from a distribution parametrized by the auxiliary parameter

$$\boldsymbol{\phi}_j^{[0],(1)} \sim p_{\boldsymbol{\vartheta}'_j}$$

At this point the algorithm starts its main loop[7], which corresponds to learning at the workers' level. Firstly, each worker samples N_{samples} times from the SNEP tilted distribution, defined as

$$q_{/j}^{(i)}(\boldsymbol{\phi}) \propto \exp\left(\gamma_j \ell_j(\boldsymbol{\phi})\right) \text{EF}\left(\boldsymbol{\vartheta}'_j - \gamma_j \boldsymbol{\vartheta}_j^{(i)}, \boldsymbol{\rho}(\boldsymbol{\phi})\right)$$

where $\ell_j(\boldsymbol{\phi})$ is the log-likelihood of the data partition seen by worker j. These samples are then used to build a Monte Carlo estimate of the expected value of the sufficient statistics, i.e. of the mean parameter.

$$\widehat{\boldsymbol{\eta}}_{q_{/j}^{(i)}} = \frac{1}{N_{\text{samples}}} \sum_{k=1}^{N_{\text{samples}}} \boldsymbol{\rho}(\boldsymbol{\phi}_j^{[k],(i)}) \approx \boldsymbol{\eta}_{q_{/j}^{(i)}}$$

where $\boldsymbol{\phi}_j^{[k],(i)}$ is the k^{th} sample from the tilted distribution at worker j and iteration i. Next, it replaces the MCMC starting point with the last sample $\boldsymbol{\phi}_j^{[0],(i+1)} \longleftarrow \boldsymbol{\phi}_j^{[N_{\text{samples}}],(i)}$.

Subsequently, it uses this Monte Carlo estimate to do a stochastic update of the mean parameter of the worker, in the same way that we used the difference between the natural parameter of the tilted and the natural parameter of the global approximation to find the new natural parameter of the site in EP. In theory, here one should update the mean parameter of the site approximation $\boldsymbol{\eta}_j^{(i)}$ using the difference between the tilted mean parameter and the mean parameter corresponding to the auxiliary natural parameter $\boldsymbol{\eta}'_j = \nabla_{\boldsymbol{\vartheta}'_j} A(\boldsymbol{\vartheta}'_j)$. However, one can decide to update the local-global parameter $\boldsymbol{\vartheta}'_j$ only

[7]Superscript $^{(i)}$ denotes the i^{th} main-loop iteration in any given worker.

Algorithm 5: Stochastic Natural-Gradient EP

Initialize prior $\boldsymbol{\vartheta}_0$.
for every worker $j = 1, \ldots, N_{\text{workers}}$ **do**
  Initialize $\boldsymbol{\vartheta}_j^{(1)}$, $\boldsymbol{\eta}_j^{(1)}$, $\boldsymbol{\vartheta}_{-j}$, $\boldsymbol{\vartheta}'_j$ and store natural parameter $\boldsymbol{\vartheta}_j^{\text{old}} = \boldsymbol{\vartheta}_j^{(1)}$.
  Sample MCMC starting point $\boldsymbol{\phi}_i^{[0],(1)} \sim p_{\boldsymbol{\vartheta}'_j}$.
  for every iteration $i = 1, \ldots, N_{\text{iter}}$ **do**
    Sample $\boldsymbol{\phi}_j^{[1],(i)}, \ldots, \boldsymbol{\phi}_j^{[N_{\text{samples}}],(i)}$ from $q_{/j}^{(i)}(\boldsymbol{\phi})$.
    Update mean parameter of the worker

$$\boldsymbol{\eta}_j^{(i+1)} \longleftarrow \boldsymbol{\eta}_j^{(i)} + \epsilon_j^{(i)} \left(\widehat{\boldsymbol{\eta}}_{q_{/j}^{(i)}} \; - \; \boldsymbol{\eta}_j'^{(i)} \right)$$

    Convert mean parameters to natural parameters.

$$\boldsymbol{\vartheta}_j^{(i+1)} \longleftarrow \nabla_{\boldsymbol{\eta}_j^{(i+1)}} A^*(\boldsymbol{\eta}_j^{(i+1)})$$

    Update worker state $\boldsymbol{\phi}_j^{[0],(i+1)} \longleftarrow \boldsymbol{\phi}_j^{[N_{\text{samples}}],(i)}$
    if $i \equiv 0 \pmod{N_{\text{outer}}}$ **then**
      Update auxiliary $\boldsymbol{\vartheta}'_j \leftarrow \boldsymbol{\vartheta}_{-j} + \boldsymbol{\vartheta}_j^{(i)}$
    end if
    if $i \equiv 0 \pmod{N_{\text{sync}}}$ **then**
      Send difference in natural parameter of the worker to server.

$$\Delta_j \longleftarrow \boldsymbol{\vartheta}_j^{(i)} - \boldsymbol{\vartheta}_j^{\text{old}}$$

      Update $\boldsymbol{\vartheta}_j^{\text{old}} = \boldsymbol{\vartheta}_j^{(i)}$
      Update cavity $\boldsymbol{\vartheta}_{-j} \longleftarrow \boldsymbol{\vartheta}_{\text{global}} - \boldsymbol{\vartheta}_j^{\text{old}}$
    end if
  end for
end for
for posterior server **do**
  Initialize global natural parameter

$$\boldsymbol{\vartheta}_{\text{global}} \longleftarrow \boldsymbol{\vartheta}_0 + \sum_{j=1}^{N_{\text{workers}}} \boldsymbol{\vartheta}_j^{(1)}$$

  for each Δ_j in order of arrival **do**
    Update global natural $\boldsymbol{\vartheta}_{\text{global}} \longleftarrow \boldsymbol{\vartheta}_{\text{global}} + \Delta_j$
    Send $\boldsymbol{\vartheta}_{\text{global}}$ to j^{th} worker.
  end for
end for

every N_{outer} iterations and instead update $\boldsymbol{\eta}_j^{(i)}$ using the current value of $\nabla_{\boldsymbol{\vartheta}_{-j}+\boldsymbol{\vartheta}_j^{(i)}} A(\boldsymbol{\vartheta}_{-j} + \boldsymbol{\vartheta}_j^{(i)})$

$$\boldsymbol{\eta}_j^{(i+1)} \longleftarrow \boldsymbol{\eta}_j^{(i)} + \epsilon_j^{(i)} \left(\widehat{\boldsymbol{\eta}}_{q_{/j}^{(i)}} \; - \; \nabla_{\boldsymbol{\vartheta}_{-j}+\boldsymbol{\vartheta}_j^{(i)}} A(\boldsymbol{\vartheta}_{-j} + \boldsymbol{\vartheta}_j^{(i)}) \right)$$

where $\epsilon_j^{(i)}$ is the step size used by the j^{th} worker, at iteration i. The new mean parameter is then transformed to natural parameter using the backward mapping. To simplify the notation, at every iteration i, we denote by $\boldsymbol{\vartheta}_j'^{(i)}$ the **(possibly unupdated) local-global** natural parameter

$$\boldsymbol{\vartheta}_j'^{(i)} = \boldsymbol{\vartheta}_{-j} + \boldsymbol{\vartheta}_j^{(i)},$$

the notation for SNEP is summarized in Table III.

	Natural	Mean
j^{th} Site	$\boldsymbol{\vartheta}_j$	$\boldsymbol{\eta}_j$
j^{th} Cavity	$\boldsymbol{\vartheta}_{-j}$	$\boldsymbol{\eta}_{-j}$
Global	$\boldsymbol{\vartheta}_{\text{global}}$	
j^{th} Tilted (MC estimate)		$\widehat{\boldsymbol{\eta}}_{q_{/j}^{(i)}}$
Local-Global	$\boldsymbol{\vartheta}'_j$	$\boldsymbol{\eta}'_j$
Possibly-Unupdated Local-Global	$\boldsymbol{\vartheta}_j'^{(i)}$	$\boldsymbol{\eta}_j'^{(i)}$

Table III: Natural and Mean parameters for the j^{th} site approximation, the j^{th} cavity, the global approximation, the local-global approximation and the (possibly-unupdated) local-global approximation. Subscripts $^{(i)}$ indicating the iteration number have been omitted in this table to simplify notation.

If N_{outer} iterations have passed by, the algorithm then updates the auxiliary parameter

$$\boldsymbol{\vartheta}'_j \leftarrow \boldsymbol{\vartheta}_{-j} + \boldsymbol{\vartheta}_j^{(i)},$$

and, similarly, if N_{sync} iterations have passed by it synchronizes with the server. When synchronizing with the server, the worker calculates the difference between the previous natural parameter and the new one and sends it to the server.

$$\Delta_j \longleftarrow \boldsymbol{\vartheta}_j^{(i)} - \boldsymbol{\vartheta}_j^{\text{old}}$$

Then, it updates $\boldsymbol{\vartheta}_j^{\text{old}}$, which contains the most recently sent natural parameter to the server. The worker then **asynchronously** receives a new value for the natural parameter of the global approximation kept by the server, and uses it to update its prior, the cavity distribution.

$$\boldsymbol{\vartheta}_{-j} \longleftarrow \boldsymbol{\vartheta}_{\text{global}} - \boldsymbol{\vartheta}_j^{\text{old}}$$

Finally, the last few lines of Algorithm 5 describe how the posterior server updates the natural parameter of the global approximation. During initialization, the server initializes the global natural parameter to be equal to the starting auxiliary variable. Then it updates $\boldsymbol{\vartheta}_{\text{global}}$ by summing to it, in order of arrival, the difference Δ and sending it to the corresponding worker.

XIV. Experiments

A. Experimental Setup

In this paper, we have compared SNEP and aEP on the population genetics problem described in Section VIII. Data sets were generated using the Genetree software by Bahlo and Griffiths [25] for three different scenarios: growing, contracting and stable population, summarized in Table IV.

	θ	r	t_f
Contracting	1.0	0.05	1.0
Stable	10.0	1.0	1.0
Growing	20.0	20.0	0.05

Table IV: Parameter values used to generated the data sets used in this paper. In practice we work on the log-scale and we find distributions over $\boldsymbol{\phi} = (\log\theta, \log r, \log t_f)^\top$.

The 3 generated data sets consists of $D = 100$ files each, where the j^{th} file corresponds to the j^{th} sampled haplotype for that population $H_0^{(j)}$. The likelihood is estimated using a modification of Stephens and Donnelly's algorithm in [3], adapted for population with varying size, that for a parameter value $\boldsymbol{\phi}$ and a data file $H_0^{(j)}$ it returns an unbiased[8] estimate of the likelihood factor $\mathbb{E}[\widehat{p}(H_0^{(j)} \mid \boldsymbol{\phi})] = p(H_0^{(j)} \mid \boldsymbol{\phi})$.

B. Motivation for aEP

Due to the complexity of the algorithm used to estimate the likelihood, each likelihood evaluation requires a non-trivial computation time, so that standard EP using a vanilla sampler such as Random Walk Metropolis Hasting (RWMH) takes a prohibitively large number of hours to run in order to achieve convergence. This is because a very large number of samples are needed by the RWMH sampler in order to get an accurate approximation of the moments of the tilted distribution. For these reasons, we adopt instead a **particle based method** described in Appendix H which turns out to need much fewer samples and obtain more accurate results. An interesting direction for future research would be to look into alternative **pseudo-marginal samplers** that can control the acceptance probability and the variance of the samples by averaging multiple estimators [26], [27].

C. Varying the number of Workers

For SNEP we have examined its behavior as N_{workers} increases and across multiple runs, to see if its results are more reproducible compared to aEP, whose convergence is not always guaranteed. In addition, we have explored it's behavior when $\gamma = 1$ (similar to EP) and when $\gamma = N_{\text{workers}}$, i.e. what the authors call **pSNEP**.

The evolution of $\boldsymbol{\vartheta}_{\text{global}}$ in SNEP as the number of workers increases is shown in Figure 8, for the case of growing population. Convergence is monitored on a running average of the $\boldsymbol{\vartheta}_{\text{global}}$ found at each iteration, where the average is taken over a user-defined percentage of latest iterations, by default we take the average of the last 20% of the iterations. The motivation for monitoring the average is that it has lower variance and, if we take the running average of all available iterations, then we are essentially using **Polyak averaging** [28], which has good convergence guarantees on Robbins-Monro stochastic approximations [29] provided that one sets the step size $\epsilon_j^{(i)}$ to decrease slower than $1/i$, usually to $\epsilon_j^{(i)} = i^{-\frac{2}{3}}$. Unfortunately, setting the step size in this way slows down learning a lot and, in our experiments, SNEP and pSNEP barely move away from the initial $\boldsymbol{\vartheta}_{\text{global}}$, so instead we use a constant step size $\epsilon_j^{(i)} = 0.05$ for all i and j, found by trial and error.

In Figure 8 we can see how only a very small number of workers are needed to get a good approximation to the truth. Moreover, by increasing the number of workers learning is slowed down, possibly because every worker has fewer data to learn from and therefore the combined effect of all the workers learning is reduced. However, the path followed by the algorithm to reach the truth also seems to be different, compared to $N_{\text{workers}} = 2, 4, 6$. This is possibly due to the **bias-variance trade-off** [30], whereby when we initialize the algorithm to have smaller variance, we also condemn it to a more biased evolution of $\boldsymbol{\vartheta}_{\text{global}}$. Importantly, we can see that the variance-covariance matrix of the starting global approximation reduces with an increased number of workers. This is due to the very set up of SNEP, where we set the initial natural parameter for the global approximation as[9]

$$\boldsymbol{\vartheta}_{\text{global}} = \begin{pmatrix} \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + N_{\text{workers}}\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j \\ -\frac{1}{2}\text{vec}(\boldsymbol{\Sigma}_0^{-1}) - \frac{N_{\text{workers}}}{2}\text{vec}(\boldsymbol{\Sigma}_j^{-1}) \end{pmatrix},$$

meaning that we are expanding the precision matrix and therefore contracting the variance-coviariance matrix. It could be useful to scale the initial $\boldsymbol{\vartheta}_{\text{global}}$ by $\frac{1}{N_{\text{workers}}}$ to be able to compare the different runs more coherently. On the other hand, having a larger precision matrix means that at each worker, and at each iteration the cavity (representing the prior) will be larger and, depending on the sampling strategy at hand and on the likelihood landscape, this might be more or less preferred.

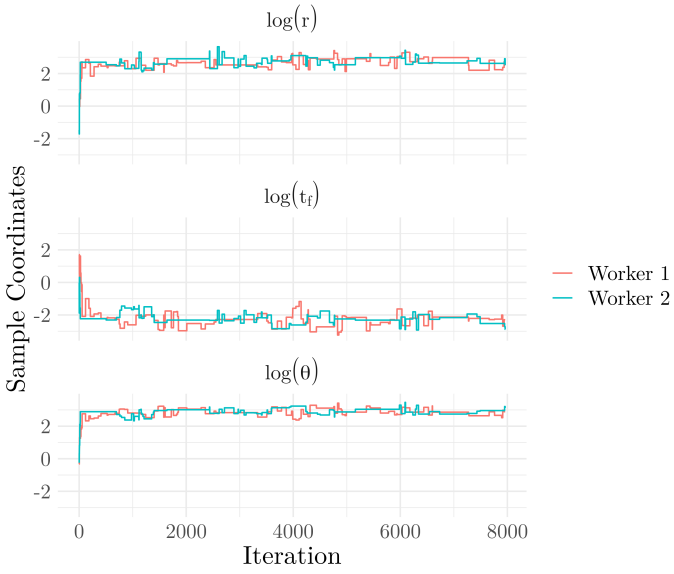

Figure 5: SNEP traces for each dimension of $\boldsymbol{\phi}$ over 200 iterations, each drawing 40 samples for a growing population, using 2 workers.

Finally, in Figure 13 in Appendix J we compare the evolution of $\boldsymbol{\vartheta}_{\text{global}}$ of SNEP and pSNEP on all data sets. We see that while SNEP seem to capture the true value of $\boldsymbol{\phi}$ more accurately with its mean, it doesn't seem to capture much of the covariance structure, which is instead steadily captured by pSNEP.

D. Sampling Diagnostic

At every iteration, each worker uses $N_{\text{samples}} = 40$ samples from the tilted distribution to approximate the sufficient

[8]Therefore MCMC samplers using such unbiased likelihood estimates will be pseudo-marginal algorithms [26].

[9]Here the index j is somehow an abuse of notation, but it indicates that we are summing up the precision matrices of the prior and of all the workers.

statistics. Such a small number of samples usually would give a poor approximation, however, since at every iteration we are only perturbing the previous parameter values $\boldsymbol{\vartheta}_j^{(i)}$ and $\boldsymbol{\eta}_j^{(i)}$ by a small amount (recall $\epsilon_j^{(i)} = 0.05$) and we are starting the successive sampling from the last sample obtained, essentially we can imagine that we are always sampling from the same distribution. Nevertheless, as can be seen in Figure 5 where traces for $N_{\text{workers}} = 2$ for a growing population are shown, the chain is rejecting very often thus leading to a lot of "sticking" that slows learning down.

The authors of SNEP suggest to counteract this behavior by modifying the MCMC state of the worker after it synchronizes with the server. The motivation behind this is that synchronization updates the cavity distribution, thus changing the tilted distribution and requiring the chain to use some "burn-in". The shift in MCMC described in [10] was implemented here but, in this particular example, seemed to bear little or no difference when using full-covariance Gaussian approximations, whereas noticeably less "stickiness" happens when shifting MCMC states with a diagonal-covariance Gaussian approximation. Unfortunately, the covariance structure of $\boldsymbol{\phi}$ is very important in population genetics problems, thus a diagonal variance-covariance matrix would not model the correlation appropriately.

The motivation for using pSNEP is that it is more robust under nastier models. At a first glance at the trace plot in Figure 6, though, it would seem that pSNEP is instead less robust and more vulnerable to noise than SNEP. However, if we consider Figure 12 in Appendix J we can see how pSNEP is actually learning the covariance structure of the underline posterior distribution better. One reason for this apparently contradicting behavior is that in our code, when the Cholesky decomposition fails (during Matrix inversion), we use the Moore-Penrose pseudoinverse. We conjecture that SNEP is much more sensitive to noise than pSNEP and therefore uses the pseudoinverse more often, thus loosing the important covariance structure. Instead pSNEP, being more robust, requires taking fewer pseudoinverses thus capturing the underlying distribution better. To compare pSNEP and SNEP traces as the number of workers increases, we refer the reader to Figure 11 in the same Appendix.

E. Running Time Diagnostic and Reproducibility

Due to the time constraints of this project, SNEP was implemented using a sequential architecture and therefore without leveraging its computational advantage. However, we have divided the estimated average iteration time by the number of workers, to get a crude approximation, although fairly reasonable since synchronization with the server can be performed asynchronously. We can see in Table V that even when implemented in a sequential manner SNEP performs better than aEP and, as expected, as the number of workers increases the average iteration time decreases.

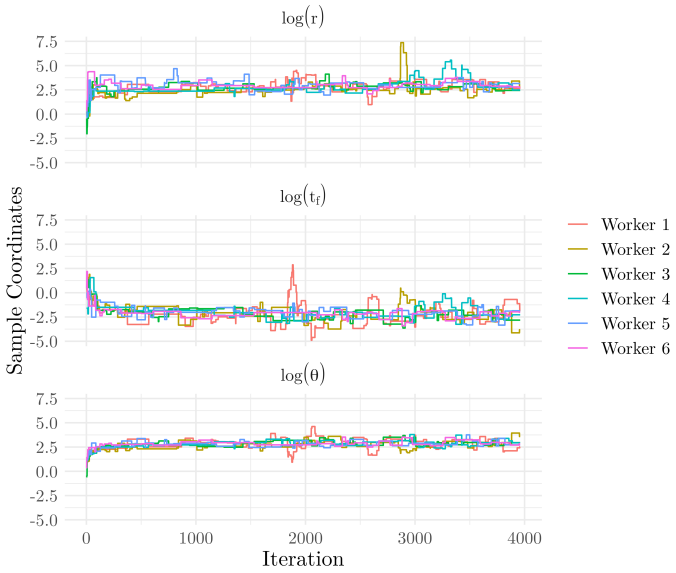

Figure 6: pSNEP traces for each dimension of $\boldsymbol{\phi}$ over 100 iterations, each drawing 40 samples for a growing population, using 6 workers.

	SNEP-64W	SNEP-8W	SNEP	aEP
C	—	—	457.50	827.58
S	—	—	458.37	562.67
G	7.85	64.72	234.12	262.57

Table V: Estimated average time per iteration of SNEP and aEP for Growing (G), Contracting (C) and Stable (S) populations. Notice that scenarios with more than 2 workers have been tested only on (G).

In Figure 7 we can also see the path followed by SNEP and aEP in two different runs for each of the parameter dimensions for the Contracting population. It seems that aEP not only is faster at reaching a solution, but also more reproducible while SNEP seems to have more variability. In order to study this behavior more carefully, more runs and more tuning is needed.

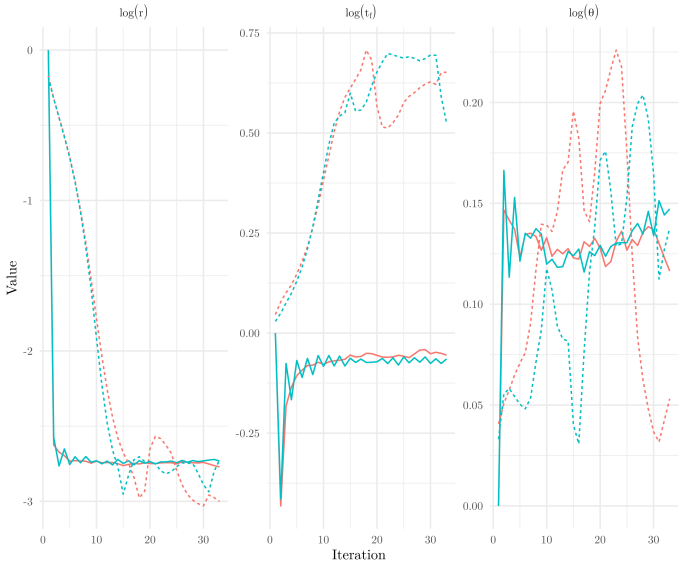

Figure 7: Dotted lines represent SNEP runs, full lines represent aEP runs. Different colors represent two different runs for a contracting population.

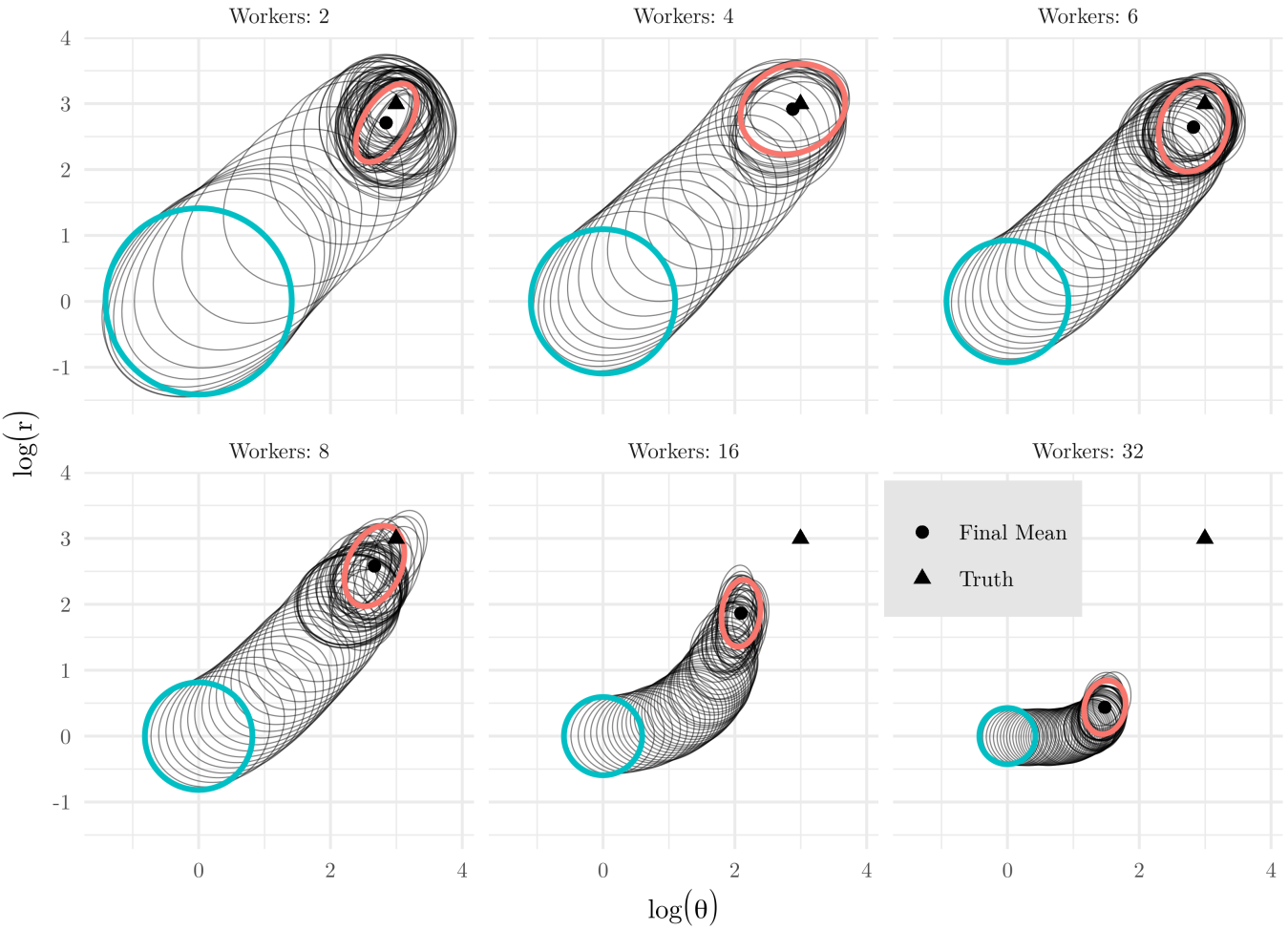

Figure 8: Evolution of $\boldsymbol{\vartheta}_{\text{global}}$ during SNEP with varying number of workers $N_{\text{workers}} = 2, 4, 6, 8, 16, 32$. Each ellipse represents 95% of the mass of the 3-dimensional Multivariate Normal distribution over $\boldsymbol{\phi} = (\log\theta, \log r, \log t_f)$. In this particular plot, only the joint over $\log\theta$ and $\log r$ is displayed. Light blue ellipses represent the initial distributions, the grey ellipses represent all iterations and the orange ellipses are found by averaging the final 20% iterations.

F. Averaged-EP Performance

Finally, we have also implemented aEP with a particle-based sampler, described in Appendix H. The evolution of $\boldsymbol{\vartheta}_{\text{global}}$ for a growing population is displayed in Figure 9, while all the evolution for all populations and all parameter dimensions is shown in Figure 14. Averaged-EP seems to find the solution much quicker than both SNEP and pSNEP, although one potential issue is that, being a fixed point algorithm, the variance shrinks as the number of iterations increases. This could lead to a severe under-estimation of the tails of the distributions, which is a common problem with variational methods [31]. Fortunately, in our experiments, this doesn't seem to happen with pSNEP/SNEP, further experimentation is needed to determine whether this might be a common behavior of SNEP compared to EP.

XV. Conclusion

In this paper we have studied the behavior of Stochastic Natural-gradient Expectation Propagation (SNEP) and Averaged EP (aEP) on three complex population genetics inference problems. The former is attractive because, while being a Variational method, it guarantees convergence and can be distributed. The latter, doesn't share as much of a strong theoretical motivation, yet it has been found to perform well on many problems and can also be parallelized. Our results suggest that SNEP is a valid alternative to aEP and we recognize the following areas for further work:

- **Reproducibility**: aEP outperforms SNEP in terms of reproducibility across runs, however, our experiments have been limited so further work is needed to examine this behavior across a larger number of runs and with more tuning of the various SNEP hyperparameters. In particular, it would be interesting to see if the reduced variance of a large number of workers (see Figure 8) would allow SNEP to obtain equally or more reproducible results, by choosing a large enough number of workers.
- **Posterior Tails and Covariance Structure**: One problem with aEP is that it could possibly underestimate the posterior tails, as often happens with variational methods. Luckily, pSNEP (a particular, robust, implementation of SNEP) seems to overcome this issue, while at the same time capturing a similar covariance structure as aEP, which was missed by SNEP.

- **Multimodal posterior**: In our tasks the posterior was unimodal, however, it would be interesting to explore if one assign the sites to different clusters (and re-normalize) thus obtaining a mixture of Gaussians as a global approximation.

XVI. Acknowledgement

I want to thank my supervisors Christophe Andrieu and Mark Beaumont for helping me make sense of the problem and guiding me towards a sensible research question. The code used here to obtain likelihood estimates uses a version of Griffiths' Genetree program modified by Mark Beaumont, Sorina Maciuca and myself.

Appendix A
Coalescence of a sample of n individuals

In the discrete-time coalescent model each generation consists of a population of size $2N$ and each individual in the current generation has a parent in the previous generation. This means that, looking backwards in time, the probability that a chosen individual in the current generation "chooses" another individual as its parent in the previous generation is $\frac{1}{2N}$. Consider now two individuals in the current generation. The probability that they will coalesce in the previous generation, i.e. that they have the same parent is given by

$$\mathbb{P}\left[2 \text{ individuals coalesce}\right] = \frac{1/(2N)^2}{1/(2N)} = \frac{1}{2N}.$$

Since at each generation children choose their parents independently of the previous generation [9], the number of generations needed for any two individuals to coalesce T_2 follows a **geometric distribution**

$$T_2 \sim \text{Geom}\left(\frac{1}{2N}\right).$$

Now consider a sample of n individuals within the population of size $2N$. Assuming that $n \ll N$, i.e. that at most only one pair of individuals will coalesce in any given generation, the probability that 2 individuals out of n will coalesce is found by multiplying the number of possible pairs by the probability that such a pair will coalesce

$$\mathbb{P}\left[2 \text{ out of } n \text{ coalesce}\right] = \binom{n}{2}\frac{1}{2N}.$$

It follows that the probability that two genes out of n will coalesce T_n generations ago is also given by a **geometric distribution**

$$T_n \sim \text{Geom}\left(\binom{n}{2}\frac{1}{2N}\right).$$

Appendix B
Limit of Geometric Distribution

In this appendix we show the limit relationship between the geometric distribution and the exponential distribution that was used to derive the continuous-time coalescent model.
Let T follow a geometric distribution with probability $p = \frac{1}{2N}$

$$T \sim \text{Geom}\left(\frac{1}{2N}\right)$$

This means that the cumulative distribution function of T is given by

$$\mathbb{P}\left[T \leq t\right] = 1 - \left(1 - \frac{1}{2N}\right)^t \qquad t \in \{1, 2, \dots, \}$$

and it's probability mass function is given below

$$\mathbb{P}\left[T = t\right] = \left(1 - \frac{1}{2N}\right)^{t-1}\frac{1}{2N} \qquad t \in \{1, 2, \dots\}$$

We now define a new random variable $\Omega = \frac{T}{2N}$. It's CDF can be found below for any $\omega \in [0, \infty)$

$$\begin{aligned}\mathbb{P}\left[\Omega \leq \omega\right] &= \mathbb{P}\left[\frac{T}{2N} \leq \omega\right] \\ &= \mathbb{P}\left[T \leq 2N\omega\right] \\ &= \mathbb{P}\left[T \leq \lfloor 2N\omega \rfloor\right] \\ &= 1 - \left(1 - \frac{1}{2N}\right)^{\lfloor 2N\omega \rfloor}\end{aligned}$$

As $N \to \infty$ then

$$\lim_{N\to\infty} 1 - \left(1 - \frac{1}{2N}\right)^{\lfloor 2N\omega \rfloor} = 1 - e^{-\omega}$$

which is the CDF of an exponential distribution with parameter $\lambda = 1$

$$\Omega \sim \text{Exp}(1) \tag{20}$$

Notice how Ω is time scaled by $2N$ which, according to Appendix A, is the expected time needed for two individuals to coalesce and find an MRCA.

Appendix C
Mutations along a branch

Suppose that the probability that a particular individual will develop a child with a mutation is given by u. Then number of generations T_m needed by a particular lineage, backwards in time, to develop a mutation follows a geometric distribution

$$T_m \sim \text{Geom}\left(u\right)$$

When N is large, we can use the result in Appendix B, scale the random variable T_m by $2N$ and approximate it's distribution with an Exponential Distribution.

$$\begin{aligned}\mathbb{P}\left[\frac{T_m}{2N} \leq t_m\right] &= 1 - (1-u)^{\lfloor 2Nt_m \rfloor} \\ &\approx 1 - e^{-2Nut_m} \\ &= 1 - e^{-\frac{\theta}{2}t_m}\end{aligned}$$

where $\theta = 4Nu$ is the scaled mutation rate. Therefore it is exponential with parameter $\frac{\theta}{2}$

$$\frac{T_m}{2N} \sim \text{Exp}\left(\frac{\theta}{2}\right)$$

Essentially the waiting time between mutations on a particular lineage is distributed exponentially. The Poisson distribution

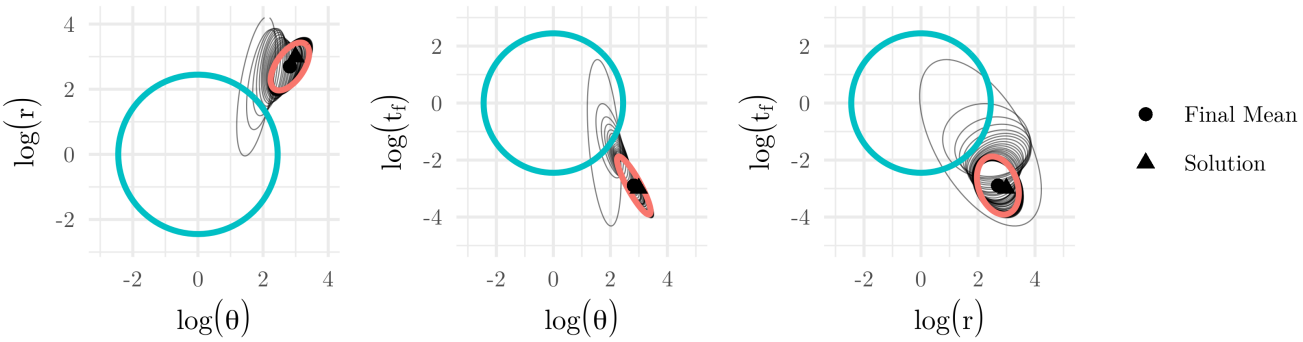

Figure 9: Evolution of $\boldsymbol{\vartheta}_{\text{global}}$ during SNEP with varying number of workers $N_{\text{workers}} = 2, 4, 6, 8, 16, 32$. Each ellipse represents 95% of the mass of the 3-dimensional Multivariate Normal distribution over $\boldsymbol{\phi} = (\log\theta, \log r, \log t_f)$. In this particular plot, only the joint over $\log\theta$ and $\log r$ is displayed. Light blue ellipses represent the initial distributions, the grey ellipses represent all iterations and the orange ellipses are found by averaging the final 20% iterations.

can equivalently describe this process by modelling the number of mutations M_ℓ occurring on a lineage of length ℓ.

$$\mathbb{P}\left[M_\ell = m\right] = \frac{(\ell\theta)^m}{m!2^m}\exp\left\{-\frac{\theta}{2}\ell\right\}$$

Indeed the probability that there are no mutations, on a single lineage, during $\ell = t_m$ generations is given by

$$\mathbb{P}\left[M_\ell = 0\right] = e^{-\frac{\theta}{2}t_m} = \mathbb{P}\left[\frac{T_m}{2N} > t_m\right],$$

which leads to the CDF of an exponential distribution with the correct parameter

$$\mathbb{P}\left[\frac{T_m}{2N} \leq t_m\right] = 1 - e^{-\frac{\theta}{2}t_m}.$$

Appendix D
Multivariate Normal as an Exponential Family Distribution

Suppose that the d-dimensional random variable $\boldsymbol{\phi}$ follows a multivariate normal distribution $\boldsymbol{\phi} \sim \mathbb{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ whose pdf,

$$f(\boldsymbol{\phi}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{\phi}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi}-\boldsymbol{\mu})\right\},$$

can be rearranged into the following expression

$$f(\boldsymbol{\phi}) \propto \exp\left\{\boldsymbol{\phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi}\right\}.$$

Consider the second term in the exponent now.

$$-\frac{1}{2}\boldsymbol{\phi}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi} = -\frac{1}{2}\sum_{k=1}^{d}\sum_{j=1}^{d}\phi_k\Sigma^{-1}_{kj}\phi_j.$$

This expression can equivalently be written in terms the trace of the matrix $\boldsymbol{\Gamma} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\boldsymbol{\phi}\boldsymbol{\phi}^\top$ as shown below

$$\begin{aligned}
\text{tr}\left[\boldsymbol{\Gamma}\right] &= -\frac{1}{2}\text{tr}\left[\begin{pmatrix}\Sigma^{-1}_{11} & \cdots & \Sigma^{-1}_{1d}\\ \vdots & \ddots & \vdots \\ \Sigma^{-1}_{d1} & \cdots & \Sigma^{-1}_{dd}\end{pmatrix}\begin{pmatrix}\phi_1^2 & \cdots & \phi_1\phi_d\\ \vdots & \ddots & \vdots \\ \phi_d\phi_1 & \cdots & \phi_d^2\end{pmatrix}\right] \\
&= -\frac{1}{2}\text{tr}\left[\begin{pmatrix}\sum_{j=1}^d\Sigma^{-1}_{1j}\phi_j\phi_1 & \cdots & \sum_{j=1}^d\Sigma^{-1}_{1j}\phi_j\phi_d\\ \vdots & \ddots & \vdots \\ \sum_{j=1}^d\Sigma^{-1}_{dj}\phi_j\phi_1 & \cdots & \sum_{j=1}^d\Sigma^{-1}_{dj}\phi_j\phi_d\end{pmatrix}\right] \\
&= -\frac{1}{2}\sum_{k=1}^{d}\sum_{j=1}^{d}\phi_k\Sigma^{-1}_{kj}\phi_j
\end{aligned}$$

Equivalently, the expression on the left-hand side is nothing but the Frobenius inner product since both $\boldsymbol{\phi}\boldsymbol{\phi}^\top$ and $\boldsymbol{\Sigma}^{-1}$ are symmetric and real matrices

$$\left\langle -\frac{1}{2}\boldsymbol{\Sigma}^{-1}, \boldsymbol{\phi}\boldsymbol{\phi}^\top\right\rangle_F = \text{tr}\left(\boldsymbol{\Gamma}\right) = \text{vec}\left(-\frac{1}{2}\boldsymbol{\Sigma}^{-1}\right)\text{vec}\left(\boldsymbol{\phi}\boldsymbol{\phi}^\top\right),$$

where the vectorizing operation stacks rows one after the other into a column vector, so that $\text{vec}(\boldsymbol{\phi}\boldsymbol{\phi}^\top) \in \mathbb{R}^{d^2\times 1}$. As a consequence, we can write the pdf as

$$f(\boldsymbol{\phi}) \propto \exp\left\{\langle\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\phi}\rangle + \left\langle -\frac{1}{2}\boldsymbol{\Sigma}^{-1}, \boldsymbol{\phi}\boldsymbol{\phi}^\top\right\rangle_F\right\}.$$

The sufficient statistics, the natural parameter and the mean parameters are then shown below

$$\begin{aligned}
\boldsymbol{\rho}(\boldsymbol{\phi}) &= \begin{pmatrix}\boldsymbol{\phi}\\ \text{vec}(\boldsymbol{\phi}\boldsymbol{\phi}^\top)\end{pmatrix} \\
\boldsymbol{\vartheta} &= \begin{pmatrix}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\\ -\frac{1}{2}\text{vec}\left(\boldsymbol{\Sigma}^{-1}\right)\end{pmatrix} \\
\boldsymbol{\eta} &= \begin{pmatrix}\boldsymbol{\mu}\\ \text{vec}(\boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^\top)\end{pmatrix}.
\end{aligned}$$

Appendix E
Gradient of the Log Partition Function

The log-partition function of a member of the exponential family $p_{\boldsymbol{\vartheta}} = \text{EF}(\boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi}))$ is the cumulant generating function

of that distribution. Here we only show that the first order derivative of $A(\boldsymbol{\vartheta})$ is the expected value of the sufficient statistics (see Figure 10)

$$\begin{aligned}
\nabla_{\boldsymbol{\vartheta}} A(\boldsymbol{\vartheta}) &= \nabla_{\boldsymbol{\vartheta}} \log \int \exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi})\right\} d\boldsymbol{\phi} \\
&= \frac{\int \nabla_{\boldsymbol{\vartheta}} \exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi})\right\} d\boldsymbol{\phi}}{\int \exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi})\right\} d\boldsymbol{\phi}} \\
&= \int \boldsymbol{\rho}(\boldsymbol{\phi}) \frac{\exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi})\right\}}{\int \exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi})\right\} d\boldsymbol{\phi}} d\boldsymbol{\phi} \\
&= \int \boldsymbol{\rho}(\boldsymbol{\phi}) \exp\left\{\boldsymbol{\vartheta}^\top \boldsymbol{\rho}(\boldsymbol{\phi}) - A(\boldsymbol{\vartheta})\right\} d\boldsymbol{\phi} \\
&= \mathbb{E}_{p_{\boldsymbol{\vartheta}}}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right] \\
&= \boldsymbol{\eta}.
\end{aligned}$$

In the derivation above we have followed the treatment of [14] and used the dominated convergence theorem.

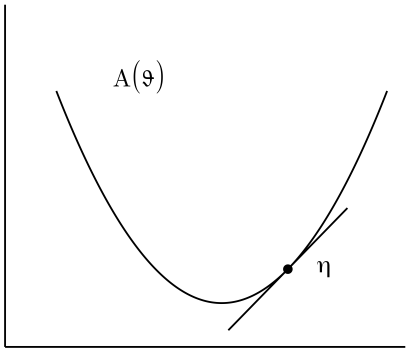

Figure 10: The log partition function $A(\boldsymbol{\vartheta})$ is convex and at every value of $\boldsymbol{\vartheta}$ in its domain its slope is given by the mean parameter $\boldsymbol{\eta}$.

Appendix F
KL divergence between Exponential Family members

Suppose we have two exponential family distributions sharing the same sufficient statistics $\boldsymbol{\rho}(\boldsymbol{\phi})$ but having different sets of natural and mean parameters, $(\boldsymbol{\vartheta}_1, \boldsymbol{\eta}_1)$ and $(\boldsymbol{\vartheta}_2, \boldsymbol{\eta}_2)$ respectively. Following the result in [10] we show how to write the KL divergence between these two members in a convenient form

$$\begin{aligned}
\mathrm{KL}\left(p_{\boldsymbol{\vartheta}_1} \parallel p_{\boldsymbol{\vartheta}_2}\right) &= \mathbb{E}_{p_{\boldsymbol{\vartheta}_1}}\left[\log p_{\boldsymbol{\vartheta}_1} - \log p_{\boldsymbol{\vartheta}_2}\right] \\
&= \left[\boldsymbol{\vartheta}_1^\top \boldsymbol{\eta}_1 - A(\boldsymbol{\vartheta}_1)\right] - \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta}_1 + A(\boldsymbol{\vartheta}_2) \\
&= A^*(\boldsymbol{\eta}_1) - \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta}_1 + A(\boldsymbol{\vartheta}_2) \\
&= A^*(\boldsymbol{\eta}_1) - \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta}_1 + (\boldsymbol{\vartheta}_2 - \boldsymbol{\vartheta}_2)^\top \boldsymbol{\eta}_2 + A(\boldsymbol{\vartheta}_2) \\
&= A^*(\boldsymbol{\eta}_1) - \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta}_1 - A^*(\boldsymbol{\eta}_2) + \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta}_2 \\
&= A^*(\boldsymbol{\eta}_1) - A^*(\boldsymbol{\eta}_2) + (\boldsymbol{\eta}_2 - \boldsymbol{\eta})^\top \boldsymbol{\vartheta}_2,
\end{aligned}$$

where on the third line we have used the fact that the variational problem

$$A^*(\boldsymbol{\eta}) = \sup_{\boldsymbol{\eta} \in \mathcal{M}_1} \boldsymbol{\vartheta}_1^\top \boldsymbol{\eta} - A(\boldsymbol{\vartheta}_1)$$

has solution $\boldsymbol{\eta}_1 = \mathbb{E}_{p_{\boldsymbol{\vartheta}}}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$, on the fourth line we have just added and subtracted the same quantity, and on the fifth line we have use the same reasoning as before, i.e. that

$$\boldsymbol{\eta}_2 = \mathbb{E}_{p_{\boldsymbol{\vartheta}_2}}\left[\boldsymbol{\rho}(\boldsymbol{\phi})\right]$$

is the solution to the variational problem

$$A^*(\boldsymbol{\eta}) = \sup_{\boldsymbol{\eta} \in \mathcal{M}_2} \boldsymbol{\vartheta}_2^\top \boldsymbol{\eta} - A(\boldsymbol{\vartheta}_2)$$

where $\boldsymbol{\mathcal{M}}_1$ and $\boldsymbol{\mathcal{M}}_2$ are the mean parameter domain for the first and the second distribution respectively.

Appendix G
EM Algorithm

In this appendix, we recall the Expectation Maximization algorithm [32] and how it can be framed as a maximization-maximization procedure, following the work in [33]. Notice that the notation in this appendix is independent of the notation in the rest of the paper, and it is rather lose for ease of exposition.

Let $\mathbf{y}$ be some observed data, and $\mathbf{x}$ be some unobserved data. We call $p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta})$ the complete-data likelihood. Our aim is, given the observed data $\mathbf{y}$ to find a value of the parameter $\boldsymbol{\theta}^*$ that maximizes the incomplete-data likelihood $p(\mathbf{y} \mid \boldsymbol{\theta})$. The EM algorithm starts with a guess $\boldsymbol{\theta}^{(0)}$ and then for $t = 1, 2, \ldots$ alternates between these two steps:

1) **E-step**: Compute conditional distribution of the unobserved data given the observed data $p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)})$.
2) **M-step**: Choose new parameter value $\boldsymbol{\theta}^{(t+1)}$ so that it maximizes $\mathbb{E}_{p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)})}[\log p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta})]$.

Given some fixed unobserved variables, we can define a joint distribution between the conditional distribution found in the E-step, which we denote $\widetilde{p}$, and the parameter

$$\begin{aligned}
F(\widetilde{p}, \boldsymbol{\theta}) &= \mathbb{E}_{\widetilde{p}}[\log p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta})] - \mathbb{E}_{\widetilde{p}}[\log \widetilde{p}] \\
&= \mathbb{E}_{\widetilde{p}}[\log p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}) p(\mathbf{y} \mid \boldsymbol{\theta})] - \mathbb{E}_{\widetilde{p}}[\log \widetilde{p}] \\
&= \log p(\mathbf{y} \mid \boldsymbol{\theta}) - \mathrm{KL}\left(\widetilde{p} \parallel p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta})\right).
\end{aligned}$$

The function F is similar to the well-known **variational free-energy**, and we can write the EM algorithm as a coordinate-maximization algorithm.

1) **E-step**:

$$p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)}) = \arg\max_{p} F(p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}), \boldsymbol{\theta}^{(t-1)})$$

2) **M-step**:

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} F(p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(t)}), \boldsymbol{\theta})$$

In summary, we started off wishing to find $\boldsymbol{\theta}^*$ by solving

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \ \log p(\mathbf{y} \mid \boldsymbol{\theta}),$$

but by subtracting a KL-divergence term between an auxiliary distribution and a distribution depending on $\boldsymbol{\theta}$ we can solve the Maximum Likelihood problem by coordinate maximization.

APPENDIX H
AVERAGED-EP PARTICLE-BASED SAMPLER

Moment matching in our aEP experiments has been performed using a particle-based sampler working as follows. Suppose we need to sample the tilted distribution at worker j. Firstly, we sample N_{samples} parameter values from the cavity distribution in equation (17)

$$\boldsymbol{\vartheta}^{[1]}, \dots, \boldsymbol{\vartheta}^{[N_{\text{samples}}]} \sim \text{EF}\left(\boldsymbol{\vartheta}_0 + \frac{D-1}{D}\boldsymbol{\vartheta}, \boldsymbol{\rho}(\boldsymbol{\phi})\right).$$

Next, for each sampled particle $\boldsymbol{\vartheta}^{[k]}$ we uses Stephens and Donnelly's (SD) algorithm [3] to produce an unbiased estimate for the likelihood $w_k = \widehat{p}(H_0^{(j)} \mid \boldsymbol{\phi})$

$$\boldsymbol{\vartheta}^{[k]} \xrightarrow{\text{SD}} w_k.$$

Since our aim is to obtain samples that are proportional to the tilted distribution, which can be seen as the local posterior distribution with prior equal to the cavity and likelihood $p(H_0^{(j)} \mid \boldsymbol{\phi})$, we reweight the prior particles by their corresponding likelihood estimate, so that an estimate of the mean of such samples (re-normalized) is given below

$$\widehat{\boldsymbol{\mu}} = \frac{\sum_{k=1}^{N_{\text{samples}}} \boldsymbol{\vartheta}^{[k]} w_k}{\sum_{k=1}^{N_{\text{samples}}} w_k},$$

and, similarly, an estimate of the variance-covariance matrix of such samples if found as follows

$$\widehat{\boldsymbol{\Sigma}} = \frac{\text{ESS}}{\text{ESS}-1} \frac{\sum_{k=1}^{N_{\text{samples}}} w_k (\boldsymbol{\vartheta}^{[k]} - \widehat{\boldsymbol{\mu}})(\boldsymbol{\vartheta}^{[k]} - \widehat{\boldsymbol{\mu}})^\top}{\sum_{k=1}^{N_{\text{samples}}} w_k}, \quad (21)$$

where we have kept the estimate unbiased using Kish's Effective Sample Size (ESS) [34]

$$\text{ESS} = \frac{\left(\sum_{k=1}^{N_{\text{samples}}} w_k\right)^2}{\sum_{k=1}^{N_{\text{samples}}} w_k}$$

following the EP-ABC work in [35]. In practice we monitor the ESS and only compute (21) when ESS is large enough, otherwise we use the precision matrix and the rescaled mean of the cavity to estimate those of the tilted, and try again at the next iteration.

Of course what we actually need are the natural parameters, so we need to transform $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ into an estimated precision matrix $\widehat{\boldsymbol{\Sigma}^{-1}}$ and an estimated rescaled mean $\widehat{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}$. Unfortunately, naively inverting $\widehat{\boldsymbol{\Sigma}}$ does not lead to an unbiased estimate of the precision matrix, so instead we use a correction that has been shown to work well [36], [17]

$$\widehat{\boldsymbol{\Sigma}^{-1}} = \frac{\text{ESS} - N_{\text{params}} - 2}{\text{ESS}-1}\widehat{\boldsymbol{\Sigma}}^{-1},$$

where N_{params} is just the dimensionality of $\boldsymbol{\phi}$, i.e. 3. Then it's straightforward to obtain an estimate for the rescaled mean

$$\widehat{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}} = \widehat{\boldsymbol{\Sigma}^{-1}}\widehat{\boldsymbol{\mu}}$$

APPENDIX I
ADDITIONAL TRACE PLOTS

In the following Appendix we collect additional trace plots. In particular, in Figure 11, we consider a growing population and we let SNEP run for 200 iterations. At each iteration, each worker samples 40 times from the tilted distribution, so that the total number of samples is 8000. In these trace plots the reduced variance of a large number of workers is more evident, together with the severely slower learning done by workers with a high number of workers.

In Figure 12, the setting is the same, but we consider pSNEP and only 100 iterations, with the same number of samples per iteration. We can see that in pSNEP sometimes very odd samples are accepted, this seems to be contradicting the conjecture that pSNEP is more stable than SNEP in [10].

APPENDIX J
ADDITIONAL PATH PLOTS

Similarly, in this Appendix we display more plots showing the convergence of SNEP and pSNEP, displayed in Figure 13, and aEP, displayed in Figure 14. We can see how generally SNEP is better at capturing the true solution with it's mean but might fail to learn the covariance structure of the true distribution. On the other hand, aEP seems to be reaching similar (possibly more narrower) covariance structures as pSNEP, but much quicker. One potential issue with aEP is that, being a fixed point algorithm, as the number of iterations increases the variance shrinks considerably, which could lead to an under-representation of the tails, typical of Variational Methods.

REFERENCES

[1] J. D. Watson and F. H. C. Crick, "Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[2] R. C. Punnett, "Linkage in the sweet pea (lathyrus odoratus)," *Journal of Genetics*, vol. 13, no. 1, pp. 101–123, 1923.

[3] M. Stephens and P. Donnelly, "Inference in molecular population genetics," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 4, pp. 605–635, 2000.

[4] R. C. Griffiths and S. Tavare, "Ancestral inference in population genetics," *Statist. Sci.*, vol. 9, pp. 307–319, 08 1994.

[5] N. A. Rosenberg and M. Nordborg, "Genealogical trees, coalescent theory and the analysis of genetic polymorphisms," *Nature Reviews Genetics*, vol. 3, no. 5, pp. 380–390, 2002.

[6] S. Zöllner and J. K. Pritchard, "Coalescent-based association mapping and fine mapping of complex trait loci," *Genetics*, vol. 169, pp. 1071–1092, Feb 2005. 15489534[pmid].

[7] A. Morris, J. Whittaker, and D. Balding, "Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies," *American journal of human genetics*, vol. 70, pp. 686–707, 04 2002.

[8] P. Tataru, M. Simonsen, T. Bataillon, and A. Hobolth, "Statistical Inference in the Wright–Fisher Model Using Allele Frequency Data," *Systematic Biology*, vol. 66, pp. e30–e46, 08 2016.

[9] J. Sigwart, "Gene Genealogies, Variation and Evolution: A Primer in Coalescent Theory.—Jotun Hein, Mikkel H. Schierup, and Carsten Wiuf. 2004. Oxford University Press, Oxford. xiii + 276 pp. ISBN 0-19-852996-1, £29.95 (paperback); ISBN 0-19-852995-3, £65.00 (hardback).," *Systematic Biology*, vol. 54, pp. 986–987, 12 2005.

[10] L. Hasenclever, S. Webb, T. Lienart, S. Vollmer, B. Lakshminarayanan, C. Blundell, and Y. W. Teh, "Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server," *J. Mach. Learn. Res.*, vol. 18, p. 3744–3780, Jan. 2017.

[11] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, p. 859–877, Feb 2017.

[12] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, p. 1–305, Jan. 2008.

[13] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[14] L. D. Brown, "Fundamentals of statistical exponential families with applications in statistical decision theory," *Lecture Notes-Monograph Series*, vol. 9, pp. i–279, 1986.

[15] D. Bertsekas, *Convex Optimization Theory*. Athena Scientific optimization and computation series, Athena Scientific, 2009.

[16] T. P. Minka, "Expectation propagation for approximate bayesian inference," in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, (San Francisco, CA, USA), p. 362–369, Morgan Kaufmann Publishers Inc., 2001.

[17] A. Vehtari, A. Gelman, T. Sivula, P. Jylänki, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. Robert, "Expectation propagation as a way of life: A framework for bayesian inference on partitioned data," 2014.

[18] G. Dehaene and S. Barthelmé, "Expectation propagation in the large-data limit," 2015.

[19] J. Winn and C. M. Bishop, "Variational message passing," *J. Mach. Learn. Res.*, vol. 6, p. 661–694, Dec. 2005.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] Y. Li, J. M. Hernández-Lobato, and R. E. Turner, "Stochastic expectation propagation," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2323–2331, Curran Associates, Inc., 2015.

[22] S.-I. Amari, O. E. Barndorff-Nielsen, R. E. Kass, S. L. Lauritzen, and C. R. Rao, *Chapter 2: Differential Geometrical Theory of Statistics*, vol. Volume 10 of *Lecture Notes–Monograph Series*, pp. 19–94. Hayward, CA: Institute of Mathematical Statistics, 1987.

[23] T. Minka, "Power ep," Tech. Rep. MSR-TR-2004-149, January 2004.

[24] A. J. Smola, V. Vishwanathan, and E. Eskin, "Laplace propagation.," in *NIPS* (S. Thrun, L. K. Saul, and B. Schölkopf, eds.), pp. 441–448, MIT Press, 2003.

[25] M. Bahlo and R. Griffiths, "Coalescence time for two genes from a subdivided population," *Journal of mathematical biology*, vol. 43, pp. 397–410, 12 2001.

[26] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient monte carlo computations," *Ann. Statist.*, vol. 37, pp. 697–725, 04 2009.

[27] C. Andrieu, A. Doucet, S. Yıldırım, and N. Chopin, "On the utility of metropolis-hastings with asymmetric acceptance ratio," 2018.

[28] B. Polyak, "New stochastic approximation type procedures," *Avtomatica i Telemekhanika*, vol. 7, pp. 98–107, 01 1990.

[29] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 400–407, 09 1951.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.

[31] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman, "Yes, but did it work?: Evaluating variational inference," 2018.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[33] R. M. Neal and G. E. Hinton, *A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants*, p. 355–368. Cambridge, MA, USA: MIT Press, 1999.

[34] H. Wiegand, "Kish, l.: Survey sampling. john wiley & sons, inc., new york, london 1965, ix + 643 s., 31 abb., 56 tab., preis 83 s.," *Biometrische Zeitschrift*, vol. 10, no. 1, pp. 88–89, 1968.

[35] S. Barthelmé, N. Chopin, and V. Cottet, "Divide and conquer in abc: Expectation-progagation algorithms for likelihood-free inference," 2015.

[36] R. A. Wijsman *The Annals of Statistics*, vol. 12, no. 3, pp. 1145–1150, 1984.

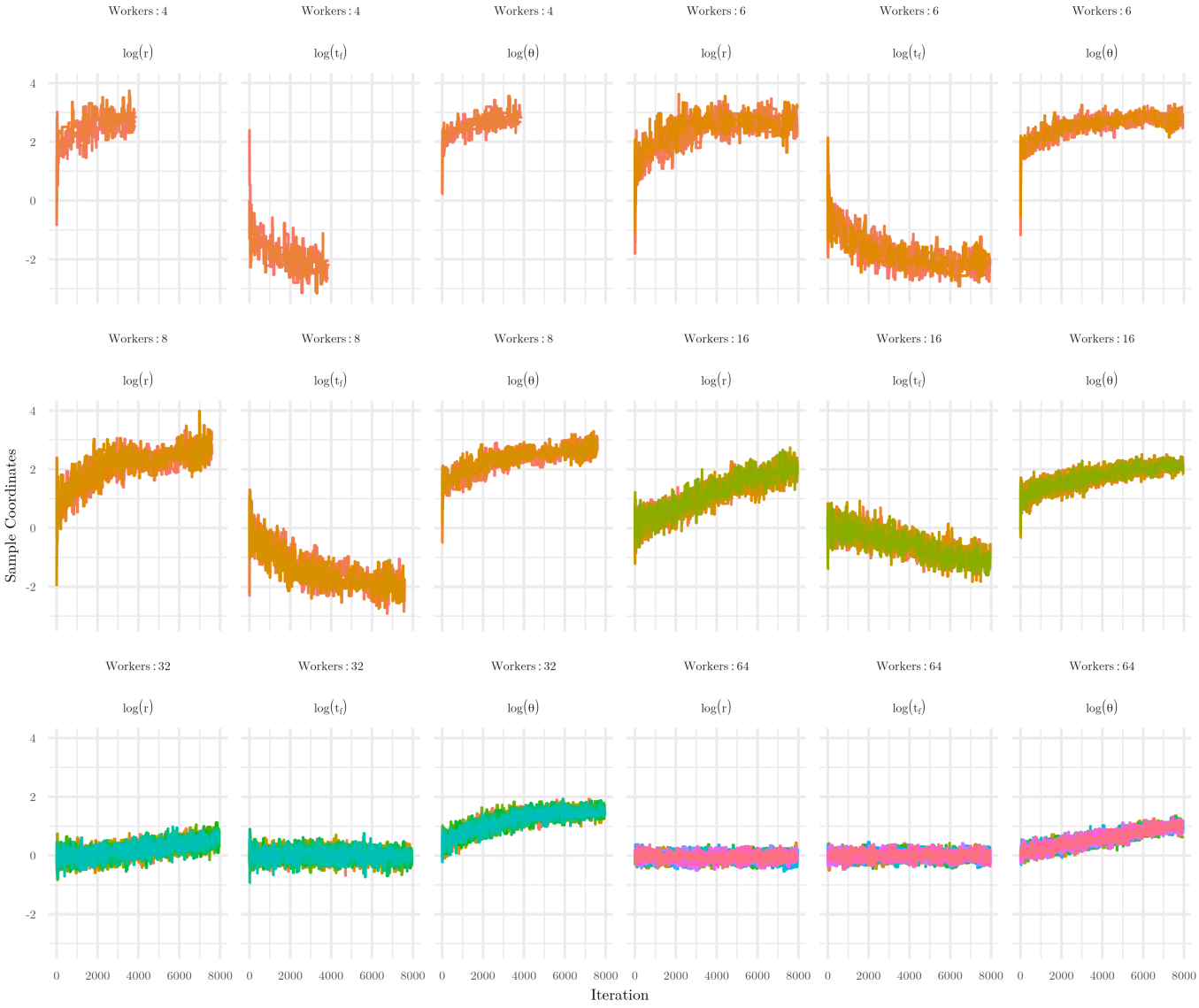

Figure 11: SNEP traces for a growing population as the number of workers increases over 200 iterations. Notice that SNEP with 4 workers actually converged in just over 100 iterations (see Figure 8) so it's traces are half as long.

Figure 12: pSNEP traces for a growing population as the number of workers increases over 100 iterations. Here we can see a poorer mixing compared to standard SNEP.

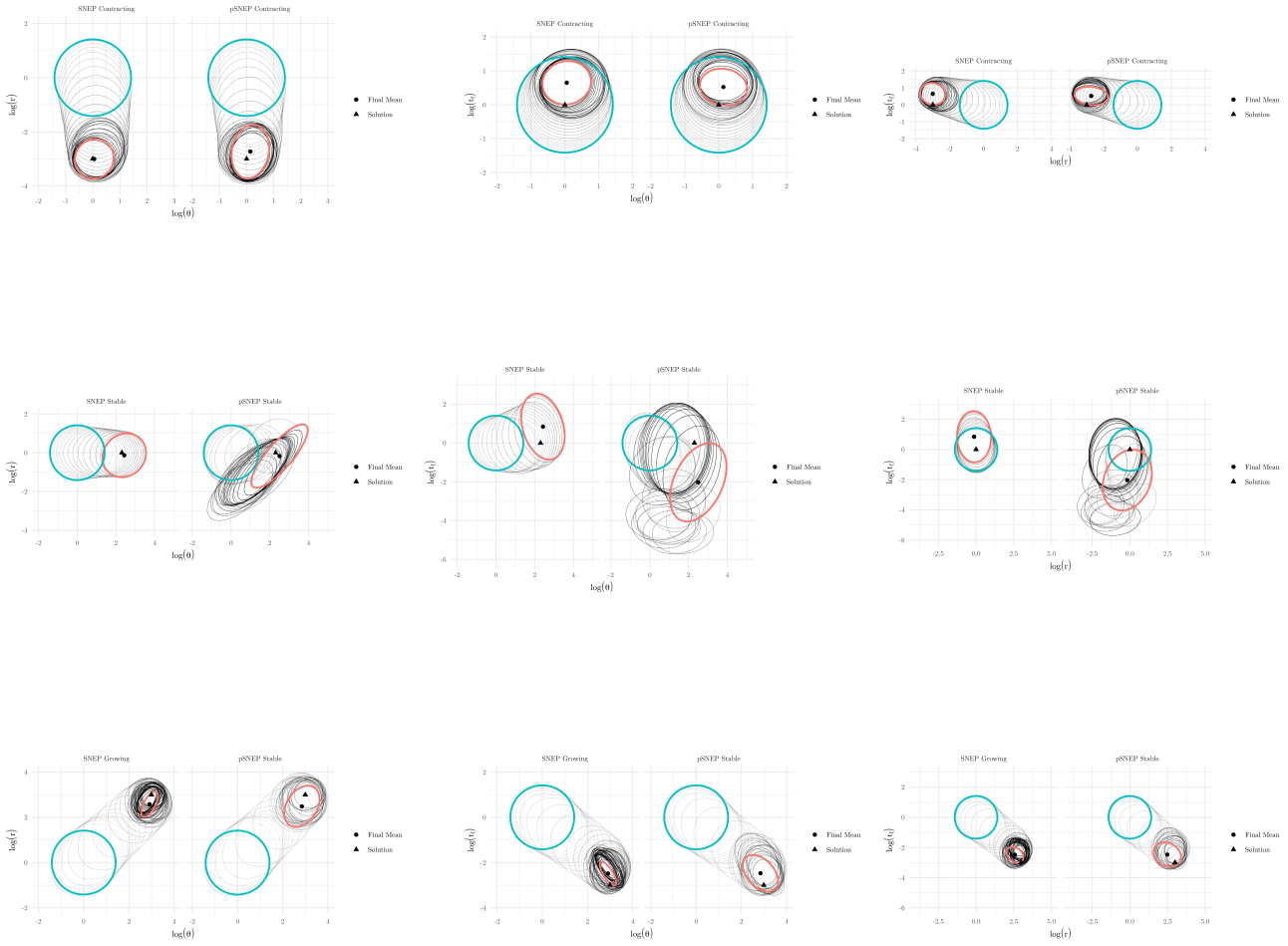

Figure 13: Comparing $\boldsymbol{\vartheta}_{\text{global}}$ evolution between SNEP and pSNEP for all data sets and parameter dimension combinations. Contracting population is in the first row, stable is in the second and growing is in the third. The three columns represent $(\log\theta, \log r)$, $(\log\theta, \log t_f)$ and $(\log r, \log t_f)$ respectively.

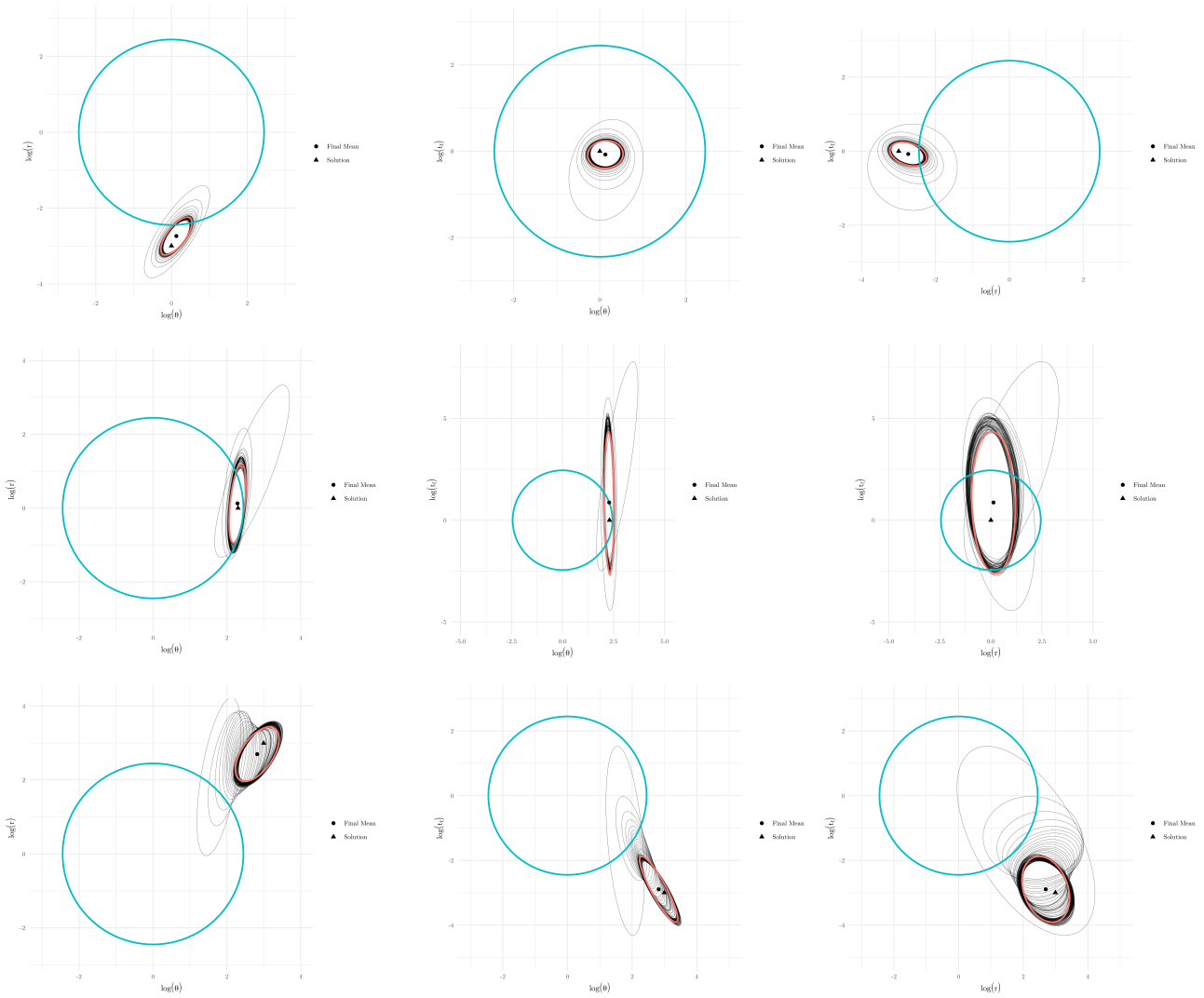

Figure 14: Evolution of $\boldsymbol{\vartheta}_{\text{global}}$ using aEP for all data sets and parameter dimension combinations. Contracting population is in the first row, stable is in the second and growing is in the third. The three columns represent $(\log\theta, \log r)$, $(\log\theta, \log t_f)$ and $(\log r, \log t_f)$ respectively.