Answering Pierre notes

May 12, 2020

1 Section on computing/optimizing the likelihood

Q3 : If $\vec{h} \neq 0$ the Langevin equation became:

$$\begin{aligned} \gamma \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= -\boldsymbol{h} - \boldsymbol{J}\mathbf{x} + \xi(t) \\ &= -\boldsymbol{J}\left(\mathbf{x} + \boldsymbol{J}^{-1}\boldsymbol{h}\right) + \xi(t) \end{aligned} \tag{1}$$

renaming $\mathbf{x}' = \mathbf{x} + \boldsymbol{J}^{-1}\boldsymbol{h}$ we obtain the Langevin equation for $\mathbf{x}'$:

$$\gamma \frac{\mathrm{d}\mathbf{x}'}{\mathrm{d}t} = -\boldsymbol{J}\mathbf{x}' + \xi(t) \tag{2}$$

which as our equation (2) has stationary distribution:

$$P_{eq}(\mathbf{x}') = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{C}|}} \exp\left\{-\frac{1}{2}\mathbf{x}'^T \boldsymbol{C}^{-1}\mathbf{x}'\right\} \tag{3}$$

with $\boldsymbol{C} = \left\langle \mathbf{x}'\mathbf{x}'^T \right\rangle$

Coming back to original variable $\mathbf{x}$ we get:

$$P_{eq}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{C}|}} \exp\left\{-\frac{1}{2}(\mathbf{x} + \boldsymbol{J}^{-1}\boldsymbol{h})^T \boldsymbol{C}^{-1}(\mathbf{x} + \boldsymbol{J}^{-1}\boldsymbol{h})\right\} \tag{4}$$

with $\boldsymbol{C} = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle - \langle \mathbf{x} \rangle \langle \mathbf{x} \rangle$, and $\langle \mathbf{x} \rangle = -\boldsymbol{J}^{-1}\boldsymbol{h}$.

In both cases (equations (3) and (4)) $\boldsymbol{J} = \boldsymbol{C}^{-1}$.

Conclusion: We could always work with $\vec{h} = 0$ equations because what is important is how to measure the covariance matrix.

Q4 : This is an important point that I would like to discuss a bit. I used two different parameterizations: the one related with the Eulerian angles and other one which is more simple related to exponentiation of a skew symmetric matrix, I implemented both but I'm using mainly the last one. Below I will briefly explain each one:

A. Parametrization in term of generalized Eulerian angles:

The idea is that each base vector $\vec{s}_a$ is parameterized according to the $a-th$ column of an arbitrary orthogonal matrix $\boldsymbol{S}$ parameterized in terms of $L(L-1)/2$ independent variables $\theta_{pq}, p = 1, 2, ..., L; q = 1, 2, ..., L; p < q$, named Generalized Eulerian angles (Eulerian angles is for L=3).

How to construct this matrix:

1- The transformation in a two-dimensional subspace of an $L - dimensional$ space is given by an L-dimensional matrix of the following form:

$$\boldsymbol{T}_{pq} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & a_{pq} & & b_{pq} & \\ & & & 1 & & \\ & & -b_{pq} & & a_{pq} & \\ & & & & & 1 \end{pmatrix} \tag{5}$$

where $a_{pq} = \cos\theta_{pq}$ and $b_{pq} = \sin\theta_{pq}$. All diagonal elements are unity except the diagonal elements in the pth column and the qth column, which are $\cos\theta_{pq}$; all off-diagonal elements are zero except the one corresponding to the intersection of the pth row and the qth column, which is $\sin\theta_{pq}$, and that on the intersection of the qth row and the pth column, which is $-\sin\theta_{pq}$.

Since there are $L(L-1)/2$ off-diagonal positions above the diagonal, it is clear that one can construct $L(L-1)/2$ matrices of the form indicated in Equation (5), corresponding to all choices of p and q with $p < q$.

2- An arbitrary orthogonal matrix $\boldsymbol{S}$ can be represented as a product of these $L(L-1)/2$ orthogonal matrices with appropriate values of the $L(L-1)/2$ independent parameters θ_{pq}.

$$\boldsymbol{S}(\boldsymbol{\theta}) = \prod_{pq} T_{pq}(\theta_{pq}) \tag{6}$$

This matrix is in charge of rotations of vectors in $L - dimensional$ space. For $L = 3$ we get the classical Eulerian matrix.

3- In paper reference [7] is explained a recurrent algorithm to efficiently perform this multiplication, as well as the recurrent construction of the matrix derivatives respect to parameters θ_{pq}. I implemented this algorithms.

Then there is another question which is also important: how determine parameters θ_{pq} for a given matrix (I tried to explain this in the initializing parameters section) .

The problem is given an orthogonal matrix $\boldsymbol{Q}$ to find a set $\boldsymbol{\theta}$, such that all equations

$$S_{ij}(\boldsymbol{\theta}) = Q_{ij}$$

are satisfied. We can't solve this system of nonlinear transcendental equations algebraically, so we address this issue finding the set of $\boldsymbol{\theta}$ which minimize the function:

$$f(\boldsymbol{\theta}) = \sum_{i,j} \left[Q_{ij} - S_{ij}(\boldsymbol{\theta})\right]^2 \tag{7}$$

This is useful when we initialize parameters $\boldsymbol{\theta}$ from the matrix formed by eigenvectors of the empirical covariance matrix.

B. Exponential Parametrization.

Another parametrization that I used and I did not include in the main text is the exponential of a skew-symmetric matrix $\boldsymbol{X} = -\boldsymbol{X}^T$:

$$\boldsymbol{S} = \exp(\boldsymbol{X}) \tag{8}$$

is easy to show orthogonality that $\exp(X)^T = \exp(X^T) = \exp(-X) = \exp(X)^{-1}$.

The derivatives on $L(L-1)/2$ independent variables can be numerically computed as

$$\frac{\partial \boldsymbol{S}}{\partial X_{jk}} = \lim_{h\to 0} \frac{1}{h}\left(\exp(\boldsymbol{X} + h\boldsymbol{E}^{jk}) - \exp(\boldsymbol{X})\right) \tag{9}$$

where $\boldsymbol{E}^{jk}$ for $j > k$ is defined as a skew-symmetric matrix that has only two nonzero elements:

$$E^{pq}_{jk} = \delta_{pj}\delta_{qk} - \delta_{pk}\delta_{qj} \tag{10}$$

In this case if for a given matrix $\boldsymbol{Q}$ we want to find the matrix $\boldsymbol{X}$ is enough invert the exponential relation with:

$$\boldsymbol{X} = \log \boldsymbol{Q}$$

Q5 :

Q6 : This is not important for our calculations right know, at least I don't see how eigenvalues crossover influence in our computations. However trying to do the interpolation of the eigenvalues I saw that there is not crossover and the behavior respect to parameter ρ is smooth.

Q7 : I explained this in the first version of MS . We put γ in the likelihood and compute the gradient with respect to it

$$\nabla L = \sum_i \frac{\partial L}{\partial \rho_i}\hat{\rho}_i + \sum_{p,q} \frac{\partial L}{\partial \theta_{pq}}\hat{\theta}_{pq} + \frac{\partial L}{\partial \gamma}\hat{\gamma} \tag{11}$$

where

$$\frac{\partial L}{\partial \gamma} = -\frac{1}{2}\left\{ \sum_{k,i} \frac{1}{\lambda_{ki}} \frac{\partial \lambda_{ki}}{\partial \gamma} + \mathbf{s}_i^T \frac{\partial \mathbf{A}^i}{\partial \gamma} \mathbf{s}_i \right\} \tag{12}$$

So your trick was used only to infer empirical value of γ using eigenvalues of the empirical covariance matrix. This value was used to set initial condition for parameter γ in the optimization scheme.

Q8 : Not, we don't do the interpolation thing, just computing eigensystem for each ρ . The problem is that we need also the eigenvectors, so I found it useless. I implemented and I saw there was not crossover and was easy to fit, but at the end I did not use it. Maybe you have any idea about, because this is the tricky part of our calculations, everything else is unimportant in term of complexity.

2 Initializing parameters

I'm not sure if this deserve it's own subsection. The most important there is your trick to infer parameter γ from empirical covariance matrix. This is useful because later in the plots when we test the prediction of γ we compare with this γ_{emp}.

3 Figures

Important note: I change plot corresponding to Fig.5 center, I did a mistake and I was plotting couplings instead of correlations in this case. The rest of plots I just change the font and some minors element of the graph to make it more clear. As you can see in each plots I only label the empirical and inferred magnitude and I always compare with the true parameter wich I used to simulate the artificial data.

- About Figure 3: γ_{inf} is the γ value obtained from the optimization procedure, as I explained in Q7. With your trick I only infer γ value from empirical covariance matrix and I named γ_{emp}.

- About Figure 4 and 5 and overestimation of correlations:

I don't appreciate overestimation of correlation in no regime. This only appear in couplings.

-About range : I did not change the range in x and y because as I'm showing different regimes and the scale is different, so if I set the same scale will be some plots where will not be possible distinguish features.

4 Second round of notes

4.1 Reply to Edwin's reply :-)

(Only for questions where I'm still not satisfied!):

Q4: Both seem fine.

-The Eulerian angles seem like a technique you could simplify, no? The only thing you need is the scalar product of the eigenvectors of C with the data. It seems to me that it should be possible to compute that directly from the angles, without having to go through all these matrices multiplications? Unless the algorithm you refer to already does something smart.

Edwin:

Let me see if I get your point. What the paper of reference do is to avoid the matrix multiplication of ec (5) transformed in the recurrence equations:

$$\boldsymbol{S}(\vec{\theta}) = \boldsymbol{T}^{(L)} \tag{13}$$

with $\theta_{pq}, p = 1, 2, ..., L; q = 1, 2, ..., L; p < q$ and

$$T^{(n)}_{kl} = \cos\theta_{kn} * t^{(n)}_{kl} - \sin\theta_{kn} * s^{(n)}_{kl} \tag{14}$$

with

$$\begin{aligned} s^{(n)}_{k+1,l} &= \sin\theta_{kn} * t^{(n)}_{kl} + \cos\theta_{kn} * s^{(n)}_{kl} \\ s^{(n)}_{k+1,n} &= \cos\theta_{kn} * s^{(n)}_{kn} \\ s^{(n)}_{1,l} &= -\delta_{ln} \\ \theta_{nn} &= \pi/2 \end{aligned} \tag{15}$$

and

$$\boldsymbol{t}^{(n)} = \begin{pmatrix} \boldsymbol{T}^{(n-1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \tag{16}$$

This mean that the eigenvectors $\vec{s}_a$ of the matrix $\boldsymbol{C}$ with dimensions $L \times L$ can be written as:

$$s^k_a = T^{(L)}_{.,a} = \cos\theta_{kL} * t^{(L)}_{ka} - \sin\theta_{kL} * s^{(L)}_{ka} \quad \text{for } k = 1, .., L \tag{17}$$

and the scalar product of the eigenvectors of $\boldsymbol{C}$ with the data became:

$$R_{ia} = \vec{x}_i * \vec{s}_a = \sum_r x^r_i s^r_a = \sum_r x^r_i \left[\cos\theta_{rL} * t^{(L)}_{ra} - \sin\theta_{rL} * s^{(L)}_{ra}\right] \tag{18}$$

So the expression (18) is what I think you said is possible to pre-compute, however this could be possible if I perform before the recurrence to get t^L_{ra} and s^L_{ra}. So to do that we must obtain a symbolic expression of t^L_{ra} and s^L_{ra}. As this is not the complex part of the algorithm I did not waste time thinking on how to do it.

- What does it mean that you "mostly" use the second method? What are its advantages? In any case, you need to write a supplementary section with the details of these methods I think.

Edwin:

The problem is that in order to gain efficiency I also implemented the gradient of the likelihood in Julia using automatic differentiation via Zygote.jl package. With this aim was more simple to do it with the exponential parametrization because the automatic differentiation does not work on the recurrence conformation of the Eularian parametrization.

Resuming I implemented the computation of the gradient with the two parametrization. In the original MS I set first the Eulerian parametrization because was the first that I used, but in practice

I'm using the exponential. Also is more simple present in the manuscript the exponential, I'm not sure if this deserve an Appendix, and If we must talk about automatic differentiation in detail etc. I presented the derivatives of the exponential as a limit, but is possible arrive to an expression for this limits I could send you this.

- **Q5**: What you're saying in answer to **Q8** is that computing the new eigenvectors of $\mathbf{C}$ after you updated the Eulerian angles (or your symmetric matrix in the exponential trick) is a negligible part of the calculation, right? If so, then this question was not so important.

Edwin: Exactly.

- **Q6**: I'm not convinced at all. First, if two eigenvalues of a matrix $\mathbf{G}(\rho)$ cross, the gradient diverges if I'm correct. So this case needs to be treated separately.

Then, saying that it doesn't happen surprises me. If the tree is for some reason balanced (or possesses a balanced subtree maybe), I imagine that it's a case that can happen!

Edwin: Why do you said that if eigenvalues cross the gradient diverge? Are you confusing cross with degeneracy?

- **Q7**: Thanks, I missed this in the initial MS. I'll add it.

- **Q8**: You're saying that the most computationally intensive part is to compute eigenvalues and eigenvectors of the L $\mathbf{G}$ matrices, which are each $N \times N$. Sounds legit.

Can you maybe not recompute those every time, but use the gradient (eqs 26 and 27 of the new ms) to re-estimate them? Basically, update ρ, and then update the λ and $\vec{u}$ with a Euler step? You could do n iterations like this, and then one where you exactly recompute them.

4.2 On figures and results:

Overall, this needs to be much more detailed. First, a bunch of general comments, then specific ones. General comments:

- Not really about the ms but more its organization: I like it much better to create panels as a unique '.png' file (or 'pdf' or whatever), and then include this in the tex. It avoids handling those complicated figure arrays in the source, and also limits the number of figures one has to deal with (subplots can also be saved separately if it's useful).

Edwin: Done

- When showing three figures on a line, it's not necessary to write the label to the y axis three times (if it's the same quantity of course). Showing it once on the left is enough.

Edwin: Done

- Captions need to give a lot more detail as to what is shown. One short sentence isn't enough, as the reader then has to guess what's happening. It's okay to have a paragraph per caption.

Edwin: I improved it a bit.

- Captions again: there are a lot of typos.

- There is almost no detail of what is done right now. One does not know whether we change the true couplings or not, whether we change the tree, how the tree is chosen, etc...

Edwin: Done

Specific comments:

- $N = 10$ means a tree with 10 leaves right? Nothing is said on how long the sequences are.

Edwin:

Not, in this case $N = 10$ refer the length of sequences (now is $L = 10$) , the number of levels of the tree was $2^{10} = 1024$.

- Where do these three values of γ come from? What is γ_c? Need much more details here. By the way, γ alone says nothing, it's $\gamma\Delta t$ that matters. And we have no info on what trees we use.

Edwin: I will assume $1/\gamma$ as mutation rate to explain this, then we could invert it in the MS. From my view what matters is $\boldsymbol{J}\Delta t/\gamma$

This is what is written in the original MS about this topic, probably we must improve it but seems that you miss it.

The phylogenetic bias present at leaves of the tree depend of the time interval Δt, the mutation rate $1/\gamma$ and the potential $\boldsymbol{J}$. Two configurations linked by OU dynamics are independent if $\boldsymbol{J}\Delta t/\gamma \to \infty$ being $\Lambda \to 0$. Let define the characteristic time Δt_c as the amount of time necessary for an exponential decay of the quantity Λ by the factor $\frac{1}{e}$, to this $\boldsymbol{J}\Delta t_c/\gamma \to 1$. As $\boldsymbol{J}$ is a matrix the previous condition became in $\lambda_{max}\Delta t_c/\gamma \to 1$, where λ_{max} is the maximum eigenvalue of $\boldsymbol{J}$, describing it's higher grow direction. We going to arbitrary define that two configuration remain correlated if the sampling interval between them satisfy $\Delta t \ll \Delta t_c$.

Similarly we assume that contemporary sequences in a tree are under a strong phylogenetic correlated regime, when the path time between most distant sequences is lower than characteristic time Δt_c . For a binary homogeneous tree with K branching events this condition became $\Delta t \ll \frac{\Delta t_c}{2*K}$.

Alternatively if we consider $\boldsymbol{J}$ and Δt fixed the same criteria allows us to define the parameter $\gamma_c \to \lambda_{max}\Delta t$ as the threshold value for γ above which two sequences are considered correlated. For a binary homoneous tree $\gamma_c \to \lambda_{max} 2 * K\Delta t$ determine the characteristic value for the inverse of mutation rate. Tunning the mutation rate respect $1/\gamma_c$ we going to generate sequences with distinct levels of phylogenetic bias.

I have to include that when the tree is non-homogeneous instead of $2K\Delta t$ was computed the higger path time between two leaves.

- By the way, shouldn't we test for many more values of γ in order to have a curve "quality of correction vs γ", as well as "quality of uncorrected couplings"? One would see that for trees with long branches (i.e. $\gamma\Delta t \gg 1$), our reconstruction is useless since the uncorrected ones are good as well. As $\gamma\Delta_t$ gets smaller, both corrected and uncorrected couplings should get worse, but uncorrected ones more so.

- Are we always using the same true coupling matrix? It's not clear at all right now. On the same note, I also wonder whether we should explore different magnitudes for couplings.

Edwin: Not we don't use the same true coupling matrix. For each of the 30 sampling inference process for each regime (strong, weak and intermediate), I generate a different coupling matrix J drawn from a normal distribution with $\mu_J = 0.8$ and $\sigma_J = 0.2$, then I compute γ_c and simulated the evolutionary process on the tree taking γ as the corresponding fraction of γ_c in dependence of the regime.

- It's not clear either what the sparsification means (it's clear to me of course, but not to any reader I think), and why we do it. Terms like "contact map" are not helping at all, since our paper is not really about proteins right now. What we're doing is trying to distinguish between actual variables which are actually interacting and those that are not.

Edwin: Yes, I agree that this could be confusing. The idea was to check if our method make a difference about the inference of topology of the graph defined by $\boldsymbol{J}$. The sparsification allow make the topology more easy to describe. I mean if J is dense we can't check this because all the variables are in contact.

- The names γ_w, γ_i and γ_s are not clear enough for us to refer directly to those in figures. We should refer to *strong*, *intermediate* and *weak* cases in the captions.

Edwin: I agree, I'm changed it.

- Fig. 2: What's the number called "err" in the legend?

Edwin: This referred to quadratic distance between parameters, but I found it useless, so I removed in the new plots.

- Can we give an explanation as to why γ is underestimated when the other parameters are not corrected?

Also, I changed the meaning of γ in the main text. It's now the inverse of a time. We should adapt figures or change my notation.

Edwin: I'm not sure. The problem is that I did inference for γ as inverse of mutation rate if I invert this number the values are to small.

- Fig. 3: We should have the same axis ranges for all three sub figures.

Edwin: Done

I personally think that we should change the tree / coupling matrix for each simulation. This would make results more robust.