Notation changes

Some changes I have introduced in notation. These changes may not be uniformely applied in the MS as I have not re-written everything yet.

- γ now stands for what used to be γ^{-1}. I chose this instead of using τ because it avoids having fractions or inverse of numbers in the exponentials. I feel like γ being the inverse of a time is reasonable.

- I have called *eigenmodes* pairs of (eigenvalue, eigenvector). Now I realize it's not ideal. Maybe *eigen-decomposition* would be better, or maybe just *eigenvalues and eigenvectors*.

On indices and writing vectors / operators

I made substantial changes, which we can discuss.
First of all, I renamed the configuration length to L (instead of R). The number of leaves of the tree stays N.

We are dealing with entities that live in different dimensions. Configurations, covariance and coupling matrices live in a space of dimension L. The joint covariance matrix $\mathbb{G}$ and its eigenvectors live in dimension $L \times N$. And finally, eigenvectors of the matrix $\mathbf{G}$ as well as the matrix itself live in dimension N. I wanted some notation that would distinguish vectors and operators as well as dimensions. This needs consistant notation on both vector or operator and indices.

On indices:

- Indices a, α, β run from 1 to L, that is in one configuration. This is not like DCA where we use i, j for this. I only used α, β for the Eulerian angles.

- Indices i, j, k run from 1 to N, that is they index the data or the leaves of the tree. I usually used i and j for components of objects (as in the ith component of a vector) and k for listing N objects, as the kth eigenvalue of a matrix.

About vectors and operators:

- Vectors in dimensions L or N are noted with an arrow and a lower case letter. Thus, eigenvectors of $\mathbf{C}$ are the $\vec{s}_a$ (L of them) and eigenvectors of $\mathbf{G}^a$ are the $\vec{u}_{ka}$ (N of them). Components of these vectors are noted with an arrow-less lower case letter, with the index of the component as an exponent. The ith component of $\vec{u}_{ka}$ is u^i_{ka}.

- Vectors in dimension $L \times N$ are noted with an arrow and an upper case letter. Thus, eigenvectors of $\mathbb{G}$ are the $\vec{S}_{ka}$.

- Operators in dimensions L or N are noted by an upper case letter in bold font. So we have matrices $\mathbf{C}$ and $\mathbf{G}$ for instance.

- Operators in dimension $L \times N$ are using the `mathbb` notation. The joint correlation matrix is $\mathbb{G}$.

Examples of this:

- Data configurations are the $\vec{x}_i$.

- Concatenated data are $\vec{X} = [\vec{x}_1, \ldots, \vec{x}_N]$.

- Eigenvalues and vectors of $\mathbf{C}$ are $\{\rho_a, \vec{s}_a\}$.

- Components of $\mathbf{C}$ are $\mathbf{C}_{ab}$.

- Eigenvalues and vectors of $\mathbf{G}^a = \mathbf{G}(\rho_a)$ are $\{\lambda_k, \vec{u}_k\}$, and its components are $\mathbf{G}_{ij}$.

Abstract

I suggest leaving it as is until we have a close to definitive MS.

Introduction

Needs global re-writing I think, haven't touched it yet.
The flow is like this: people infer pairwise boltzmann models assuming independent data points. This is not always the case, and we want to focus on the case where data is structured by a tree.

- **Q1**: is there any other application to this than phylogenetic trees of proteins? In other words, are we just solving our own particular problem, or is this a general problem that occurs in different fields? etc...
  If we only refer to proteins, people might expect that we explain why we're using a continuous model. And why we're not trying this out on protein data.

Section II

Q2: Should we mention at all the Langevin equation here? Some reasons why we should not:

- The derivation is a bit confusing, as it adds a lot of variables and notations.

- As far as I understand, we're not using it much in the rest. Edwin is using a Langevin process to sample the artificial data, but it does *not* correspond to the one in equation (2)

- Diffusion equations that result from population do not look like the classical Langevin. See Kimura. For instance, without any fitness effects, the Fokker-Planck for the probability distribution of finding a trait at frequency x, $P(x,t)$, should be $\mathrm{d}_t P \propto \mathrm{d}_x^2\{x(1-x)P\}$, so not like eq. 5. What I mean to say here is that it's maybe not so useful to justify the use of our propagator, since it's not really *justfiable*. The reason we use the OU process is that we need a propagator that looks like the DCA model at equilibrium and is analytically (and numerically) computable. I think it's important to be clear about this. We're going to assume that sequences are undergoing Brownian motion in an energy landscape defined by our Hamiltonian, and we're not assuming this for biological reasons but for practical computational reasons.
  On a side note, and for the same reason: should we speak of an *evolutive* process when referring to the OU?

Reasons why we should:

- It gives intuition maybe?

If we decide to keep the derivation of the Langevin to the OU, I could try to re-write it a bit more clearly, with a bit less notation. For instance I do not think we need Lyapunov conditions and Fokker-Planck operators in the main text, could be a supplement.

Section on computing / optimizing the likelihood

- **Q3**: Do we need $h = 0$? Is the general case too complicated for the main text?

I removed references to matrices A^a. For expressing the likelihood, I think they are a bit of unnecessary abstraction as everything can be compactly written in terms of data and the two sets of eigenvectors/values.
It was however useful for writing the equations of the gradient of the likelihood in a more compact way (see original pdf equations 25 to 27).
It's easy to re-introduce should we feel it was better before.

- **Q4**: It's not very clear to me how to construct a basis of vectors from Eulerian angles, and how to take derivatives w.r. to them. The only thing that seems obvious is the fact that $N(N-1)/2$ angles fully determine the eigenvectors.

Edwin could you write a bit more about how you do this in practice? Maybe it's super easy, I simply haven't thought much about it.

- **Q5**: A related question, maybe important: From the gradient equations, it's clear that we only use the $\vec{s}_a$ and their derivatives w.r. the angles in a scalar product with the data configurations $\vec{x}_i$. Now, is there a way to compute this product without ever having to construct the $\vec{s}_a$ vectors themselves, but directly from the values of the angles? How is this done in the code?
- **Q6**: we don't say at all what we do if eigenvalues start crossing each other. And I am not sure what we actually do...
  Again could we write something there?
- **Q7**: How do we get γ? Do we use my trick at every iteration to find the optimal value given current parameters? Or do we put it in the likelihood and compute the gradient with respect to it? If so we need to write it.
- **Q8**: Right now, what I wrote is that we compute and diagonalize a set of matrices $\mathbf{G}(\rho_a)$ for each eigenvalue ρ_a and at each iteration.
  I remember we considered pre-computing a set of $\mathbf{G}$ matrices for reasonable values of ρ and then using those to interpolate actual ρ_a that we need. Are we doing this?

Initializing parameters

Does this deserve its own subsection? I'm leaving it as is right now since it contains the calculation of γ. If this calculation is done at every iteration of the optimization, then it belongs in the previous section. If it's done once for initialization, we could maybe put the whole initialization section in a supplement. By the way, the first part section Initializing parameters is not readable. What is $S_{ij}(\theta^0)$? I imagine all this is to define initial values for the Eulerian angles but I think it has to be rewritten more clearly.

Figures

- Fig. 3: It's useless to show γ_{true}. We can just materialize the diagonal.
  What is γ_{inf} ? Is it the γ inferred with my method? Inferred using the eigenvalues of the empirical covariance matrix?
  If that's the case, this could be renamed γ_{init} maybe. Not sure about this.
- Fig. 4: This means that if we use the empirical covariance to estimate couplings in the strong phylogenetic regime (by the way we need to find some other word than phylogenetic maybe?), we always overestimage them?

It seems understandable, since the tree will generate correlations which normally aren't there, and this will tend to increase couplings in the general case. But then why aren't we also overestimating correlations (Fig. 5)?

- Fig. 5: The three figures don't share the same range either on x or y-axis. Same goes for Fig. 4
- Fig. 6: This works really well!

Overall: The figures need way bigger fonts I think. Text in the figures needs to be as big as the rest.

- **Q9**: And then more general questions about results. Should we explore bigger range of parameters ? *i.e.* change the magnitude of J? Should we try with fields or is it completely useless?
  Also, We could make histograms of the Kullback-Leibler between inferred and real model, both for the corrected parameters and the empirical ones?

Second round of notes

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. 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.
- **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.
- **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!
- **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.

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).
- 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.
- 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.
- 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...

Specific comments:
- $N = 10$ means a tree with 10 leaves right? Nothing is said on how long the sequences are.
- 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. 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.
- 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.
- 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.
- Fig. 2: What's the number called "err" in the legend?
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.
- Fig. 3: We should have the same axis ranges for all three sub figures.

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