nature portfolio

Peer Review File

Probabilistic alignment of multiple networks

Corresponding Author: Professor Marta Sales-Pardo

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
I read the manuscript with interest and I found that both clear and interesting. The idea of developing a probability matching algorithm for the alignment of multiple networks is both straightforward and clever. The article makes a very good case to support such algorithm reporting both proof of principle examples, such as the synthetic data generated by the connectome of C. elegans, but also on more complex biological data, such has the hemisphere comparison of Drosophila melanogaster larvae, where the assumption of mirroring copies supports the applicability of the method. The article also reports the interesting case of study of the email network of two research institutes, showing decent performance even when the root assumption is not properly validated. The only comment I have which may be considered major concerns the performance of the proposed algorithm.

Indeed, it is shown that in comparison with the ground truth the algorithm mostly outperforms the standard euristic approaches (at least when some anchors could be provided). However, in the conclusions is generically stated:

"The sampling of the alignment space, unfortunately, comes with the price of longer computational times than heuristic optimization approaches. Nonetheless, our results justify a more computationally costly approach."

After such a clear and explanatory paper, I find disappointing from the authors to make such a vague statement. In order to assess if the higher computational cost is justified, one would need to know how much higher the computational cost is. The authors need to make a more concrete comment on this, explicitly discussing (within the examples they made) how the accuracy of their method grows with the computational resources and how this is compared with the heuristic methods.

As minor point, I suggest the authors to enlarge panel (d) of Fig.1 to improve its readability.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
The authors present a framework and algorithm for aligning multiple networks with the same number of vertices. The combination of a probabilistic matching and the ability to align multiple networks is quite interesting, and the experimental results are compelling (though lacking comparisons). That said, there are a number of issue that could be addressed.

Main comments:

- The discussion of the benefits of the algorithmic framework and drawbacks of a QAP centered framework seem to be a bit overstated. The QAP setting can be modified to incorporate node features and similarities, and edge similarities across networks (see, for example, KerGM: Kernelized Graph Matching https://arxiv.org/abs/1911.11120). Moreover, multilayer matching approaches can incorporate the type of categorical features seen here (where a notion of categorical similarity may not be appropriate). Moreover, while many existing approaches are designed for pairwise, there are a slew of multiple graph matching approaches in the literature, and claiming "all" approaches (page 1, line 34) are pairwise is a bit misleading; comparing your algorithm to existing multiple matching approaches would be useful and appropriate (e.g., Kernelized multi-graph matching https://arxiv.org/pdf/2210.05206). Also, the idea of soft matching (i.e., not a 1-1 matching, but a probabilistic

assignment) is common in the literature.

- The model where edges and non-edges are sampled independently from the blueprint, and the connection this model allows between likelihood and matching was previously explored in the literature (see, e.g., Arroyo et al. (2021). Maximum Likelihood Estimation and Graph Matching in Errorfully Observed Networks. Journal of Computational and Graphical Statistics, 30(4), 1111–1123. https://doi.org/10.1080/10618600.2021.1872582). This should be made more explicit in the text, as the stated novelty of the proposed model is a bit misleading.

- Can your approach handle the setting where the number of nodes in each network differs, perhaps via a parameter that was used to sample nodes? I think that the Bayesian approach you adopt is flexible enough to handle this.

- When discussing how to proceed with matching in the initial experiment, you write that a pair of networks are initially chosen to align to the blueprint and then aligned one by one. This seems like the complaint against sequential QAP, and you rightly point out this initial choice is a bit arbitrary. How do you choose the initial pair to align?

- Page 5, line 158: This is only equivalent in expectation?

- Page 5, line 164: I don't think that the truth is undiscoverable, more that the chosen objective function/model is not seeking that particular truth. It may be the case that a different approach could have as its global minima the latent true assignment.

- In the initialization section, there is no mention of how to initialize L. I am assuming it is the averaged of the aligned A's at each step, but this should be made explicit.

- Practically, how fast is the MCMC algorithm? For the university networks, upwards of 80000 MCMC steps were used. What was the running time of the algorithm? In general, how would the approach scale in both N (network size) and m (the number of networks being aligned)?

Typos/errata

- Page 2, line 40: "of" should be removed
- Page 2, line 57: "and" -> "an"
- Page 3, line 81: "and" -> "an"
- footnote oddly split between pages 2 and 3
- Page 7, line 182: "not" -> "nor"
- Page 12, generative model details: Throughout the derivations, π should be $\{\pi^k\}$, and $\{A\}$ and A should be $\{A^k\}$.

(Remarks on code availability)
I did not run the code, but the code is well-documented and seems extensive. The provided Python notebooks seem to contain all relevant code and experiments.

Reviewer #3

(Remarks to the Author)
This work introduces a Bayesian approach to find optimal alignment among multiple networks. This new framework presents several theoretical advantages (clearly introduced by the authors) compared to existing approaches in the field, such as graph-matching heuristics. The mathematical framework is rigorously presented and extensively validated on both synthetic and real-work networks.
While I really enjoy the presented framework and its theoretical advantages, I have hard time understanding the extent to which the presented evidence supports its practical benefit compared to existing methods.

I have some comments that will hopefully help authors address this major point:

1. Results are mainly contrasted with the fast QAP method (Ref 35). However, a more direct comparison with the methods that allow alignment of multiple networks (cited Ref 26 and 27) should be performed.
2. More in general, one would need more, and/or maybe more clearly presented, quantitative elements to assess the performance of this new method in contrast to other approaches. Statistical hypothesis testing on the related accuracy rates (like nonparametric T-tests) would for example give more objective evaluation. Reporting accuracy values on tables (even supplementary) will give an immediate impact.
3. A more in-depth robustness analysis on the synthetic networks where one knows the ground-truth would allow to evaluate the limits of the new method. For example, this can be achieved by exploring its behavior for a broad range of edge randomization levels, eg from 5% to 95%. How accuracy evolves compared to other approaches? Is there a trade-off?
4. A question that came to my mind, is how the method treats networks with different number of nodes, as in the case of the neuronal developmental network. Please specify.
5. Pag 5. 4th paragraph. Are the two initial networks selected arbitrarily? Please specify.
6. The validation on temporal networks (such as the C.elegans) is delicate since time is not really taken into account in formula 3), and we know that consecutive networks are not independent. This should be considered and/or studied more in depth, as otherwise one is left doubtful on the interpretation of the related results.
7. While brain hemispheres can be in first approximation considered globally symmetric, there exist many local circuit

asymmetries. This should be acknowledged at some point (https://www.nature.com/articles/nrn1009). So instead of computing an accuracy looking for an exact correspondence between left and right nodes, I find much more interesting the possibility to use this method to discover new correspondences between different non spatially/functional homologous nodes in the two hemispheres.
8. Pictures 2-5 all have the same format, and one often does not see what the difference is between all of them. Some effort in reorganizing them and/or representing them differently is strongly recommended. Also, the colorbars of panels c-d should be changed (use more colors) as I cannot really spot out the matched/mismatched nodes. Finally, few improvements in the legend description would be appreciated.
9. Can authors provide a quantitative analysis on the computational time complexity of their method compared to existing ones?
10. I found many little typos throughout the manuscript. Please check them carefully.

(Remarks on code availability)
na

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
I have read the reply of the authors to all referee and I was very satisfied with it. The manuscript has greatly improved thanks to the careful considerations of all the referees' comments. I recommend the paper for publication on Nature Communications.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
The authors addressed most of my concerns with the revised manuscript. The added experiments with runtimes are appreciated as well.

A few comments on the revised draft

- There is a substantial runtime difference in your approach when going from 2-4 networks in the real data network setting (Table 1 in the revised manuscript) that is not present in the synthetic network setting (Table S1 in the supplement). In the synthetic setting, going from 2 to 4 networks saw a roughly 4x increase in time; in the real data this was at least a 10x time difference (>20x in the case where more alignments were obtained)? Do you have a sense for what would lead to this? Is it the case that most runs are fast, with a few being very slow (and most likely not as good)? In light of this, the scaling with m=number of graphs should be explored a bit more.

- I do not see comment on matching networks with different numbers of nodes? At least referencing this as a future direction would be welcome.

Small errata
- In the email section (line 375), you write "Supplementary Figure S17 illustrates the in-out degree of connections for neurons in each of the 376 connectome networks, ordered according their organizational units."

(Remarks on code availability)
Extensive Github Python code provided with a ReadMe

Reviewer #3

(Remarks to the Author)
The Authors have convincingly addressed the most critical points concerning the superiority of their method compared to existing approaches. I have no more questions.

(Remarks on code availability)
NA

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.
In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.
The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.
To view a copy of this license, visit https://creativecommons.org/licenses/by/4.0/

General Comments

In this new version of the manuscript we have introduced several changes. The most important change is that we now offer a systematic comparison of our algorithm with other alignment tools in terms of accuracy and in terms of run-times for synthetic and real-world networks. Specifically, we now compare our results against the Fast QAP method [4] and the multi-way Kernel Graph Matching (KerGM) method [2], a recent method for multiple network alignment, which can incorporate node attributes to aid the alignment process. Our results clearly show that our approach is able to recover alignments closer to the ground truth than the other approaches. We report our results in both figures and tables to help readers.

In this process, we have introduced some computational improvements to our algorithm, and, for consistency, we have re-made all the figures. For that reason, some of the results we reported in the previous version of the manuscript have changed. Additionally, during this process, we spotted a mistake in the data processing of e-mail networks that resulted in the wrong ground state permutation. Our new results show that with our approach we are able to recover accurately the ground truth alignment, even without the need to use anchors. Therefore, in this version of the manuscript, we show results for two sets of e-mail networks with different filtering protocols for the edges (sizes 170 and 356 nodes, respectively) and show that, in both cases, our approach is able to recover the ground truth alignment accurately.

Finally, for *D. melanogaster*, the results for the Fast QAP method have changed slightly because we had previously run the algorithm with weighted networks instead of with binary networks, which is the case we consider throughout this work.

Reply to Reviewer #1

COMMENT: **I read the manuscript with interest and I found that both clear and interesting. The idea of developing a probability matching algorithm for the alignment of multiple networks is both straightforward and clever. The article makes a very good case to support such algorithm reporting both proof of principle examples, such as the synthetic data generated by the connectome of C. elegans, but also on more complex biological data, such has the hemisphere comparison of Drosophila melanogaster larvae, where the assumption of mirroring copies supports the applicability of the method. The article also reports the interesting case of study of the email network of two research institutes, showing decent performance even when the root assumption is not properly validated.**

Thank you for your positive appraisal of our manuscript.

COMMENT: **The only comment I have which may be considered major concerns the performance of the proposed algorithm.**

Indeed, it is shown that in comparison with the ground truth the algorithm mostly outperforms the standard heuristic approaches (at least when some anchors could be provided). However, in the conclusions is generically stated: "The sampling of the alignment space, unfortunately, comes with the price of longer computational times than heuristic optimization approaches. Nonetheless, our results justify a more computationally costly approach."

After such a clear and explanatory paper, I find disappointing from the authors to make such a vague statement. In order to assess if the higher computational cost is justified, one would need to know how much higher the computational cost is. The authors need to make a more concrete comment on this, explicitly discussing (within the examples they made) how the accuracy of their method grows with the computational resources and how this is compared with the heuristic methods.

Following your suggestion, we have now compared run-times of the different alignment approaches. Specifically, we now compare our results against two methods: Fast QAP [4] and multi-way Kernel Graph Matching (KerGM) [2], a recent method that can align multiple networks simultaneously and include node attributes.

In Tables R1 and R2 below, we compare the accuracy and run-times of these methods in the alignment of synthetic noisy copies of the A2 connectome of *C. elegans* and of real-world networks, respectively. For Fast QAP and KerGM, we provide results from obtaining 40 alignment solutions, since there is little change in accuracy by generating a larger number of solutions. In our approach, we have two options for sampling alignments: (i) starting from an initial condition, equilibrating and sampling alignments, or (ii) initializing several times the process and sampling alignments for each initialization. We find that for synthetic networks, both strategies seem to give similar results (Table R1). However, for real-world networks, the landscape is more rugged, so that it is key to sample from different initial conditions (Table R2).

As we can see in Tables R1 and R2, for synthetic networks, run-times typically increase with the difficulty of the problem (that is, for larger copy error, number of nodes and number of networks to align). The Fast QAP method is substantially faster than both our method and KerGM. In general, the best accuracy with our sampling approach comes from obtaining a larger number of samples from different initial conditions. However, if we limit the number of initial conditions and sampled alignments, our approach still has a superior performance and the run-time is comparable to that of KerGM for the majority of synthetic and real-world networks.

CHANGES: In order to have precise comparisons of the run-time of the different methods, we have added both tables to the manuscript; Table R1 to the Supplementary Material and Table R2 to the main manuscript. We have also included explicit mentions to run-time comparison in the text and modified the discussion section.

COMMENT: **As minor point, I suggest the authors to enlarge panel (d) of Fig.1 to improve its readability.**

As per your suggestion, we have changed Fig. 1 to make it more readable.

Experiment	Sampling						QAP - 40 Algs.				KerGM - 40 algs.				
(#nets, labels)	error σ 10%			error σ 30%			error σ 10%		error σ 30%		error σ 10%		error σ 30%		Rank
	Acc(%)	**t(s)**	**Algs.**(#r)	**Acc(%)**	**t(s)**	**Algs.**(#r)	**Acc(%)**	**t(s)**	**Acc(%)**	**t(s)**	**Acc(%)**	**t(s)**	**Acc(%)**	**t(s)**	
(2, no)	99.6 ± 0.6*	555	75 (15)	92 ± 2	9000	75 (15)	99.7± 0.1	2.2	9± 1	7.6	6±1	57	1.08 ±0.02	59	20
	99.7 ± 0.5*	73	40 (1)	92.3 ± 1.9	680	40 (1)					1.7±0.2	57	0.84±0.02	67	70
(2, yes)	99.7 ± 0.5*	150	75 (15)	95 ± 1	285	75 (15)	–	–	–	–	68± 3	44	2.7 ± 0.1	61	20
	99.9 ± 0.3*	55	40 (1)	94.9 ± 1.5	52	40 (1)					45± 3	52	1.96±0.06	61	70
(3, yes)	99.7 ± 0.4*	384	75 (15)	95 ± 1	555	75 (15)		–	–	–	96.3 ± 0.2	98	6.1 ± 0.2	302	20
	99.5 ± 0.5	80	40 (1)	94.7 ± 1.6	82	40 (1)					99.5±0.1	92	4.2±0.1	313	70
(4, yes)	99.5 ±0.6*	615	75 (15)	94.6 ± 1.5	2520	75 (15)	–	–	–	–	98.1 ± 0.2	166	9.2 ±0.2	653	20
	99.5 ± 0.7*	106	40 (1)	95.1 ± 1.7	231	40 (1)					99.33± 0.08	129	6.8±0.1	602	70

Table R1: **Comparison of accuracy and run-times for synthetic networks.** We generate $K \in \{2, 3, 4\}$ synthetic copies of the *C. elegans* A2 connectome, and we swap a fraction σ of edges ($A_{ij} = 1$) with non-edges ($A_{ij} = 0$) to simulate copying errors. For σ=0.1 and 0.3, we show accuracies (**Acc.**) and run-times (**t**) for the three methods we consider: our method (Sampling), Fast QAP (QAP)[1] and the Multi-way Kernel Graph Matching Method (KerGM)[2]. For the latter we show results for two values of the dimension of the kernel, Rank $\in \{20, 70\}$: 20 is the default value used for random graphs in [2] and 70 gives better results for real networks. For QAP and KerGM we report average run-times for obtaining 40 alignments for a set of K networks. For our method we report total run-times for equilibration and sampling, and indicate the number of sample alignments (**Algs.**) and, in parenthesis, the number of different initializations (#r) . For our method and KerGM, we also consider group labels as node attributes to guide the alignment process as indicated. We report mean accuracies and the standard error of the mean for 10 sets of K networks. All the results are constrained to a maximum of 100 % accuracy; * indicates cases where the upper error bound (Acc. $+ \epsilon$) exceeds 100% so that the actual interval should be interpreted as [Acc. $- \epsilon$, 100].

	Sampling			QAP			KerGM			
Dataset (#nets, #nodes)	**Acc.**(%)	**t(s)**	**Algs.** (#runs)	**Acc.**(%)	**t(s)**	**Algs.**	**Acc.**(%)	**t(s)**	**Algs.**	**Rank**
C. elegans	**88.39**	902	200 (10)	86.4±0.4	5.4	40	39 ± 3	50	40	20
(2, 224)	87.5	222	40 (5)				34 ± 4	62	40	70
C. elegans	**93.97**	19297	400 (50)	–	–	–	56 ± 1	286	40	20
(4, 224)	91.63	2088	40 (5)				56 ± 2	302	40	70
D. melanogaster	**79.21**	20471	105 (15)	77.2	146	40	59.63 ± 0.05	2615	40	20
(2, 1235)	78.58	9538	42 (6)				69.48± 0.07	3031	40	70
Emails	**95.88**	5113	300 (20)	–	–	–	67 ± 1	162	40	20
(4, 170)	95.14	142	50 (5)				80 ± 0.5	125	40	70
Emails	**96.13**	24379	280 (20)	–	–	–	25 ± 1	1943	40	20
(4, 356)	93.96	1074	50 (5)				59± 2	1789	40	70

Table R2: **Comparison of accuracy and run-times for real-world networks.** We show results for our approach (Sampling), the Fast QAP method (QAP) and the multi-way Kernel Graph Matching Method (KerGM). The column **Algs.** indicates the number of alignments obtained with each method for the reported accuracy (**Acc.**). For our method we also report in parenthesis the number of different initializations (#runs). The column (**t**) shows run-times for in seconds. For QAP, if we find several alignments with the same score, we report the mean and standard deviation of the accuracy of those alignments. For the KerGM method, since there is no cost function associated with each solution, following the protocol in [2], we report the mean accuracy against the ground truth as well as the standard error of the mean for different dimensions of the kernel (**Rank**; 20 is the default value used for random graphs in [2] and 70 seems to give better results for real networks). Numbers in bold indicate the best alignment accuracy for each set of networks.

Reply to Reviewer #2

COMMENT: **The authors present a framework and algorithm for aligning multiple networks with the same number of vertices. The combination of a probabilistic matching and the ability to align multiple networks is quite interesting, and the experimental results are compelling (though lacking comparisons). That said, there are a number of issue that could be addressed.**

Thank you for the critical appraisal of our manuscript and for the constructive comments, which have helped us improve our work.

Main comments:

COMMENT: **The discussion of the benefits of the algorithmic framework and drawbacks of a QAP centered framework seem to be a bit overstated. The QAP setting can be modified to incorporate node features and similarities, and edge similarities across networks (see, for example, KerGM: Kernelized Graph Matching https://arxiv.org/abs/1911.11120). Moreover, multilayer matching approaches can incorporate the type of categorical features seen here (where a notion of categorical similarity may not be appropriate). Moreover, while many existing approaches are designed for pairwise, there are a slew of multiple graph matching approaches in the literature, and claiming "all" approaches (page 1, line 34) are pairwise is a bit misleading; comparing your algorithm to existing multiple matching approaches would be useful and appropriate (e.g., Kernelized multi-graph matching https://arxiv.org/pdf/2210.05206). Also, the idea of soft matching (i.e. not a 1-1 matching, but a probabilistic assignment) is common in the literature.**

Thank you for pointing out these kernel approaches, which had escaped our attention. As per your suggestion and that of Reviewer #3, we have added a systematic comparison with two tools: Fast QAP (in its Graspy implementation [1]) and KerGM [2] (implementation in `https://github.com/fxdupe/graphmatchingtools`), which, indeed, allows both to use categorical node attributes and to align multiple networks simultaneously.

First, we perform a systematic comparison of aligning noisy copies of the *C. elegans* adult connectome for different probabilities of copy error and for two, three and four copies (Fig. R1 and Table R1). For two copies, we show results with and without node attributes. In all cases, our tool is able to obtain more accurate alignment solutions than the other approaches, especially for large levels of copy error. For three and four connectome copies, we only show results for KerGM and our approach and find, again, that our approach gives accurate solutions for levels of noise for which KerGM cannot.

Second, we compare the performance of the different approaches when applied to real networks. As shown in Table R2, for the alignment of connectomes and e-mail networks, our approach also finds more accurate alignments in all cases.

CHANGES: We have added background to the manuscript to better contextualize the contributions of our approach with respect to current approaches. We have also added a systematic comparison with other network alignment approaches for synthetic experiments with noisy copies of networks

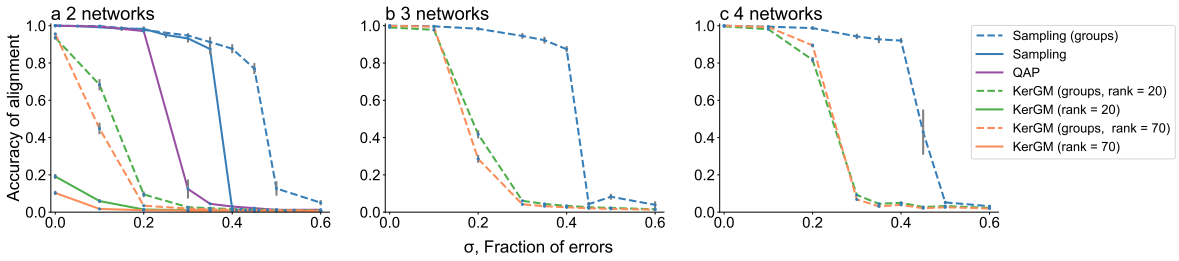

Figure R1: **Comparison of the performance of different network alignment methods for synthetic networks.** We compare our method (Sampling, blue lines) with two other alignment methods (KerGM and QAP) in the alignment of sets of: (**a**) $K = 2$; (**b**) $K = 3$; (**c**) $K = 4$ synthetic networks. These synthetic networks are copies of the connectome A2 of *C. elegans*, with varying levels of edge copying errors, σ (that is, the fraction of $A_{ij} = 1$ entries in the original adjacency matrix that are swapped with $A_{ij} = 0$ entries). The x-axis shows the fraction of errors for each set of copies. Each point is the average over 10 different sets of network copies. For KerGM and QAP, results correspond to obtaining 40 alignments for each set of copies. For our sampling approach for each set of copies we obtain the most probable alignment from 75 alignments sampled from 15 separate initializations. Dashed lines show the results where we also consider node labels (in this case node groups). We only show the result for QAP method (purple line) in (a) because it can only align pairs of networks.

(shown in Fig. 2 of the main manuscript and a Table S1 in the Supplementary Material). For real-world networks, we have included performance comparisons with the KerGM approach and QAP in a summary Table in the main manuscript.

COMMENT: **The model where edges and non-edges are sampled independently from the blueprint, and the connection this model allows between likelihood and matching was previously explored in the literature (see, e.g., Arroyo et al. (2021). Maximum Likelihood Estimation and Graph Matching in Errorfully Observed Networks. Journal of Computational and Graphical Statistics, 30(4), 1111–1123. https://doi.org/10.1080/10618600.2021.1872582). This should be made more explicit in the text, as the stated novelty of the proposed model is a bit misleading.**

Thank you for pointing this paper out. In their formulation, Arroyo *et al.* indeed assume, as we do, a copy mechanism with errors. However, there is one key difference between our respective approaches—while Arroyo *et al.* assume that one graph is a copy with errors of the other graph, in our approach, we assume that the blueprint from which all observed graphs are copied with errors is unobserved, as is the case in real-world networks. Therefore, in our inference problem we need to infer both the underlying blueprint and the correct matching from the observations.

Besides being more realistic, our model has practical advantages. For example, in Arroyo *et al.*'s approach, aligning network A to network B is not necessarily the same as aligning B to A, which in most cases is objectionable. Because of this reason, our approach is also more naturally generalizable to alignments with more than two networks.

CHANGES: In the revised version of the manuscript, we explicitly mention the work of Arroyo *et al.*, and discuss how it sets a precedent for our approach.

COMMENT: **Can your approach handle the setting where the number of nodes in each network differs, perhaps via a parameter that was used to sample nodes? I think that the Bayesian approach you adopt is flexible enough to handle this.**

We could not agree more with your observation. Our approach allows us to consider networks with a different number of nodes. To do so, we need to incorporate a generative mechanism by which nodes are copied (or not) from the blueprint to the observation. Indeed, this is a line of work that we are investigating. However, preliminary work in this direction shows that the choice of mechanism by which nodes are copied can have substantial impact in the alignment solutions the algorithm is able to sample. Therefore, this is work in progress that goes beyond the scope of the current manuscript.

CHANGES: We have included this consideration in the discussion as one of the future steps for our tool.

COMMENT: **When discussing how to proceed with matching in the initial experiment, you write that a pair of networks are initially chosen to align to the blueprint and then aligned one by one. This seems like the complaint against sequential QAP, and you rightly point out this initial choice is a bit arbitrary. How do you choose the initial pair to align?**

Note that, although we start by aligning a pair networks, the mapping of the nodes of these two initial networks is not fixed when we add additional networks. That is, when we add a third network, the algorithm samples the alignment landscape of *all three* networks, including the initial two (the only difference is that, rather than using a heuristic initialization for node identity mappings to the blueprint, for the initial networks we use the best mapping obtained in the previous step). We do this every time we add a new network, so that, in the end, we are considering the alignment of the K networks simultaneously and not by pairs.

This means that, if the networks to align are not identical, the initial pair of networks could affect perhaps the run-time of our approach, but not the ground state or the equilibrium sample of alignments.

CHANGES: We have clarified this process in Section 3 of the Supplementary Material.

COMMENT: **- Page 5, line 158: This is only equivalent in expectation?**

Yes, from the observed number of errors in copying edges and non-edges, the q and p values we report correspond to the maximum likelihood values of these parameters given the blueprint and the observation.

CHANGES: We have modified this sentence to make it more precise.

COMMENT: **Page 5, line 164: I don't think that the truth is undiscoverable, more that the chosen objective function/model is not seeking that particular truth. It may be the case that a different approach could have as its global minima the latent true assignment.**

In our synthetic experiments, we generate the noisy copies using the generative model that underlies our inference approach. In that situation, our approach is Bayes optimal, and no alternative method can do better in general. It is possible that, for a specific realization of noisy copies, one might find a particular method that finds the correct latent alignment for that specific instance.

However, in general, that method will not be able to do so for other noisy copies generated in the same way, since it would be suboptimal in Bayesian terms. It is in this sense that, when we can find a solution with an lower energy (or larger posterior) than the ground truth, we say that the ground truth is undiscoverable.

COMMENT: **In the initialization section, there is no mention of how to initialize L. I am assuming it is the averaged of the aligned A's at each step, but this should be made explicit.**

You are correct. At each step L_{ij}, is set to the majority of all A^k_{ij}. If there is no majority, we set $L_{ij} = 0$ because in sparse networks, like the ones we consider, there is a higher *a priori* chance for L_{ij} to be zero. However, setting $L_{ij} = 1$ makes no difference in terms of performance.

CHANGES: We have modified the text to clarify this issue.

COMMENT: **Practically, how fast is the MCMC algorithm? For the university networks, upwards of 80000 MCMC steps were used. What was the running time of the algorithm? In general, how would the approach scale in both N (network size) and m (the number of networks being aligned)?**

As we show in Tables R1 and R2 above, the run-times of all the approaches increase with the complexity of the alignment task, which is related to the size of the network, the number of networks to align, and the difficulty of the problem (that is, the level of noise in the networks). For a fixed complexity, run-times for our full sampling are typically longer than those of other tools; but, to a large extent, this due to the fact that we use several independent sampling runs starting from different initial conditions (this typically leads to better sampling of the alignment space, and thus to better results than a single, longer MCMC run). However, for a reduced number of sampled alignments comparable to those used to obtain KerGM solutions, run-times with our sampling approach are, in general, comparable to KerGM run-times with still a significantly higher accuracy than KerGM.

With respect to the algorithmic complexity of our approach, Fig. R2 below shows how the run-time scales with network size, N. We find that the run-time is approximately $t \sim N^{2.5}$. Since our tool executes the same algorithm for each network, the algorithm scales linearly with the number of networks K to align. Therefore, the total run-time of an iteration (MCMC step) of our algorithm for K networks of size N scales as $\sim K N^{2.5}$.

CHANGES: We have added Tables R1 and R2 to the manuscript to show precise run-time comparisons across tools and networks. We have also added a discussion about the algorithmic complexity of our tool to the Supplementary Material.

COMMENT: **Typos/errata**

- **Page 2, line 40: "of" should be removed**
- **Page 2, line 57: "and" -> "an"**
- **Page 3, line 81: "and" -> "an"**
- **footnote oddly split between pages 2 and 3**
- **Page 7, line 182: "not" -> "nor"**

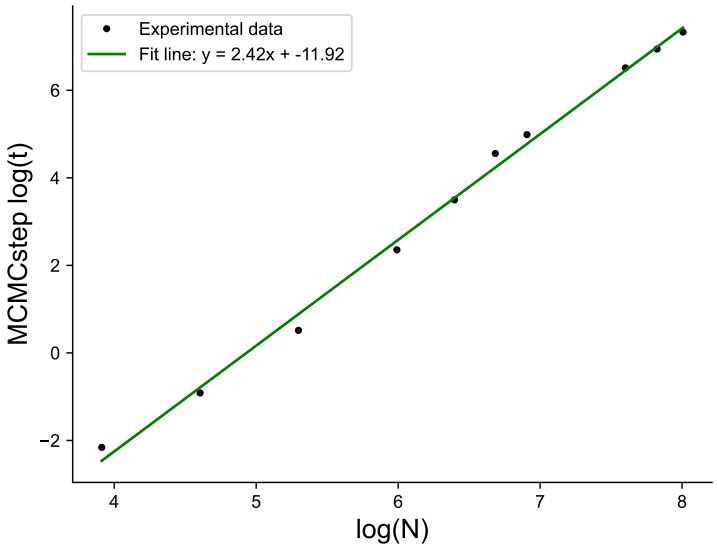

Figure R2: **Logaritmic scaling of run-time with the network size.**

- **Page 12, generative model details: Throughout the derivations, π should be $\{\pi^k\}$, and $\{A\}$ and A should be $\{A^k\}$.**

Thank you for carefully reading our manuscript and for pointing these out. We have revised the manuscript for typos and errors.

Reply to Reviewer #3

COMMENT: **This work introduces a Bayesian approach to find optimal alignment among multiple networks. This new framework presents several theoretical advantages (clearly introduced by the authors) compared to existing approaches in the field, such as graph-matching heuristics. The mathematical framework is rigorously presented and extensively validated on both synthetic and real-work networks. While I really enjoy the presented framework and its theoretical advantages, I have hard time understanding the extent to which the presented evidence supports its practical benefit compared to existing methods.**

I have some comments that will hopefully help authors address this major point:

Thank you for taking the time to critically read our manuscript and for giving us the opportunity to improve it with your comments and suggestions.

COMMENT: **Results are mainly contrasted with the fast QAP method (Ref 35). However, a more direct comparison with the methods that allow alignment of multiple networks (cited Ref 26 and 27) should be performed.**

As detailed below, we have followed your suggestion and compared systematically our approach to two benchmarks, both on synthetic and real networks.

COMMENT: **A more in-depth robustness analysis on the synthetic networks where one knows the ground-truth would allow to evaluate the limits of the new method. For example, this can be achieved by exploring its behavior for a broad range of edge randomization levels, eg from 5% to 95%. How accuracy evolves compared to other approaches? Is there a trade-off?**

As per your suggestion, we have performed a systematic comparison between our tool, the fast QAP method [4, 1] and KerGM (the Kernel Graph Matching method, as suggested by reviewer #2 [2]). Unfortunately, while there is code available for Ref. 27 [3], we could not manage to make it work for our analysis because of problems with the dependencies that we were unable to solve. Therefore, in our comparative analysis, we consider the multi-way KerGM tool and Fast QAP.

Specifically, following your suggestion we carry out synthetic experiments in which, from the adult connectome of *C. elegans*, we generate $K = \{2, 3, 4\}$ copies with different copy error probabilities—we generate errorful copies by swapping a fraction σ of edges ($A_{ij} = 1$) with non-edges ($A_{ij} = 0$). In this way, the density of edges is the same for all copies. For each value of σ we generate 10 different sets of K networks, and we compute the alignment accuracy of each set according to the different algorithms (QAP, KerGM and our approach for $K = 2$, and KerGM and our approach for $K = 3, 4$). Figure R1 above shows the average and standard error of the mean for each value of σ and algorithm. Note that KerGM can also include node attributes, therefore for $K = 3, 4$ we show the results using group labels, since the performance is superior for both KerGM and our approach. Our results show that our tool outperforms FastQAP and KerGM since it is able to recover the correct alignment permutations for larger values of copy error than the other two approaches.

CHANGES: We have included the systematic comparison for synthetically genereted connectomes (shown here in Fig. R1) to Fig. 2 in the manuscript. We have also modified the text to explain and discuss this new analysis.

COMMENT: **More in general, one would need more, and/or maybe more clearly presented, quantitative elements to assess the performance of this new method in contrast to other approaches. Statistical hypothesis testing on the related accuracy rates (like nonparametric T-tests) would for example give more objective evaluation. Reporting accuracy values on tables (even supplementary) will give an immediate impact.**

Thank you for your suggestion. We agree with you that for real networks we have a unique measurement, and it is hard to assess the statistical significance of the differences in accuracy across methods. For this reason, we have performed the synthetic experiments with noisy network copies described in our previous reply, and in which we can average over different realizations of the noise and establish differences across algorithms (Fig. R1, Table R1).

For real world networks, Table R2 summarizes the performance of the different tools on each network. Because the output of KerGM is a set of alignments with no assigned quality function, we compute the mean accuracy over alignments as well as the standard error of the mean. Additionally, for our approach, we also show accuracies obtained for 40-50 sampled alignments, and with larger sampling sets, to have an estimate of the variability that can be expected for different sampling sizes. Our results show that, while with a smaller number of sampled alignments the accuracy is slightly lower, the difference in accuracies between our approach and that of KerGM is much greater than the variability in accuracy we observe in our approach or in KerGM. Therefore, we can conclude that the performance of our approach is superior with statistical confidence.

CHANGES: We have added Table R2 to the main text, showing the performance of the different algorithms for the different real networks considered.

COMMENT: **A question that came to my mind, is how the method treats networks with different number of nodes, as in the case of the neuronal developmental network. Please specify.**

In this manuscript, we assume that the number of nodes in each network is the same. Therefore, if one of the networks we consider has a lower number of nodes, we patch the network with disconnected nodes, as usually done in other methods.

However, as noted in our response to reviewer #2 above, our approach is well suited to consider networks with a different number of nodes more rigorously. To do so, we need to incorporate a generative mechanism by which nodes are copied (or not) from the blueprint to the observation. Indeed, this is a line of work that we are investigating. However, preliminary work in this direction shows that the choice of mechanism by which nodes are copied can have substantial impact in the alignment solutions the algorithm is able to sample. Therefore, this is work in progress that goes beyond the scope of the current manuscript.

CHANGES: We have included this consideration in the discussion as one of the future steps for our tool.

COMMENT: **Pag 5. 4th paragraph. Are the two initial networks selected arbitrarily? Please specify.**

In the first synthetic experiment, all networks are identical, therefore we select two networks at random. Note that the choice of the two initial networks has no effect on the algorithm other than a slight change in the run-time.

CHANGES: We have modified the text and the Supplementary Material to clarify this issue.

COMMENT: **The validation on temporal networks (such as the C. elegans) is delicate since time is not really taken into account in formula 3), and we know that consecutive networks are not independent. This should be considered and/or studied more in depth, as otherwise one is left doubtful on the interpretation of the related results.**

We agree with you on the fact that the alignment of *C. elegans* a different stages of development is far from trivial because the density of the network changes and the generative model we assume is not realistic. Because of this, we considered only the connectomes with the largest density of connections (that is, those corresponding to later developmental stages). However, in order to consider the alignment of all connectomes for all stages of development, we would have to include temporal evolution in our generative model. We have started a line of work in this direction, in which we include the temporal dimension in the generative model, but it will require an amount of work that goes beyond the scope of the current manuscript.

CHANGES: We have included a discussion on this issue in the manuscript to clarify the limitations of our approach as well as possible future directions.

COMMENT: **While brain hemispheres can be in first approximation considered globally symmetric, there exist many local circuit asymmetries. This should be acknowledged at some point (https://www.nature.com/articles/nrn1009). So instead of computing an accuracy looking for an exact correspondence between left and right nodes, I find much more interesting the possibility to use this method to discover new correspondences between different non spatially/functional homologous nodes in the two hemispheres.**

We agree with your observation. Indeed, one of the immediate applications of our approach would be to help refine connectome annotation, as well as to identify topological equivalences between nodes and identify connection variability across hemispheres in a systematic way. In our paper, we point to possible explanations about why our results and the ground truth might not be the same.

CHANGES: We have included these aspects in the discussion of the manuscript.

COMMENT: **Pictures 2-5 all have the same format, and one often does not see what the difference is between all of them. Some effort in reorganizing them and/or representing them differently is strongly recommended. Also, the colorbars of panels c-d should be changed (use more colors) as I cannot really spot out the matched/mismatched nodes. Finally, few improvements in the legend description would be appreciated.**

CHANGES: We have modified the color scheme and the figure captions to improve readability and figure interpretation. Additionally, as pointed out before, we have summarized all performance metrics in new tables in the manuscript and supplementary information.

COMMENT: **Can authors provide a quantitative analysis on the computational time complexity of their method compared to existing ones?**

We have carried out two types of comparative analysis with the two benchmark alignment tools: fast QAP (for pairs of networks) [1] and multi-way Kernel Graph Matching (KerGM (for multiple

networks including node attributes such as group labels) [2]; and for the alignment of synthetic noisy copies of the A2 connectome of *C. elegans* and the alignment of real-world networks (2 connectomes and 2 e-mail networks). The results of these comparisons are summarized in Tables R1 and R2, respectively. Overall, we find that in terms of accuracy our sampling approach is superior to the other methods and that the run-times are comparable (for the same number of sampled alignments) to those of KerGM.

COMMENT: **I found many little typos throughout the manuscript. Please check them carefully.**

Thank you for the suggestion, we have revised the text thoroughly to correct the typos.

References

[1] J. Chung, B. D. Pedigo, E. W. Bridgeford, B. n K. Varjavand, H. S. Helm, and J. T. Vogelstein. Graspy: Graph statistics in python. *Journal of Machine Learning Research*, 20(158):1–7, 2019.

[2] F.-X. Dupé, R. Yadav, G. Auzias, and S. Takerkart. Kernelized multi-graph matching. In E. Khan and M. Gonen, editors, *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, pages 311–326. PMLR, 12–14 Dec 2023.

[3] H.-M. Park and K.-J. Yoon. Consistent multiple graph matching with multi-layer random walks synchr onization. *Pattern Recognition Letters*, 127:76–84, 2019. Advances in Visual Correspondence: Models, Algorithms and Applications (AVC-MAA).

[4] J. Vogelstein, J. Conroy, V. Lyzinski, L. Podrazik, S. Kratzer, E. Harley, D. Fishkind, R. Vogelstein, and C. Priebe. Fast approximate quadratic programming for graph matching. *PLoS ONE*, 10:e0121002, 2015.

Reply to Reviewer #1

COMMENT: **I have read the reply of the authors to all referee and I was very satisfied with it. The manuscript has greatly improved thanks to the careful considerations of all the referees' comments. I recommend the paper for publication on Nature Communications.**

Thank you for your positive appraisal of our manuscript.

Reply to Reviewer #2

COMMENT: **The authors addressed most of my concerns with the revised manuscript. The added experiments with runtimes are appreciated as well.**

Thank you for your positive appraisal of our manuscript.

COMMENT: **There is a substantial runtime difference in your approach when going from 2-4 networks in the real data network setting (Table 1 in the revised manuscript) that is not present in the synthetic network setting (Table S1 in the supplement). In the synthetic setting, going from 2 to 4 networks saw a roughly 4x increase in time; in the real data this was at least a 10x time difference (>20x in the case where more alignments were obtained)? Do you have a sense for what would lead to this? Is it the case that most runs are fast, with a few being very slow (and most likely not as good)? In light of this, the scaling with m=number of graphs should be explored a bit more.**

The variation in run-time depends not only on scale but also on the complexity of the problem. In the case of real networks, conducting extensive sampling across different alignments is more critical than for synthetic networks. This is because, while synthetic networks typically have one energy minimum, for real world networks there are typically several minima. For instance, for *C. elegans* we find that because of the left-right symmetry there are energetically good alignments in which the vast majority of neurons are wrongly aligned in one of the connectomes. In this case, is it important to perform several, longer runs to better sample the different minima in the alignment landscape. In this scenario, one might expect the run-time to increase with the number of minima in the landscape. However, with the networks that we have it is hard to assess how the number of minima would grow with the number of networks, as the number of competing minima is network dependent.

COMMENT: **I do not see comment on matching networks with different numbers of nodes? At least referencing this as a future direction would be welcome**

Thank you for the suggestion. We have included the following sentence as a future direction: "In particular, it opens the door to developing methods to align time-evolving networks by incorporating a time-dependent generative mechanism or to align networks with a different number of nodes by incorporating node copying mechanisms into the generative process."

Reply to Reviewer #3

COMMENT: **The Authors have convincingly addressed the most critical points concerning the superiority of their method compared to existing approaches. I have no more questions.**

Thank you for your positive appraisal of our manuscript.