
Identifying neurons in *C. elegans* with continuous relaxations for Bayesian permutation inference

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 The nematode *C. elegans* is a unique model organism for neuroscientists as its
2 connectome, or neural wiring diagram, has been known for at least three decades.
3 Despite this knowledge, an understanding of the functional significance of these
4 synaptic connections has remained elusive. Now several groups can routinely
5 image the activity of a large fraction of neurons in the head of the worm, providing
6 a unique opportunity to probe this organism. We propose a hierarchical Bayesian
7 framework that combines strong prior information with data from many experi-
8 ments to estimate posteriors over the functional connectivity weights. However,
9 these attempts are stifled by a major obstacle: in many cases it is not clear exactly
10 which neurons are being imaged, so to combine information across experiments
11 one must solve a matching, or permutation inference, problem.

12 In this work we introduce new variational methods that enable the joint inference
13 of connectivity weights and neural identity. Working with actual permutations
14 would involve evaluating and differentiating an intractable partition function. As an
15 alternative, we build upon recent continuous relaxation techniques [Jang et al., 2016,
16 Maddison et al., 2016], extending them from the original case of the probability
17 simplex, to the Birkhoff polytope, the convex hull of permutation matrices. We test
18 our method with simulated data from the true connectome and known covariates
19 (neural position) and show that our approach outperforms many alternatives in
20 identifying neurons.

21 1 Introduction

22 The nematode *C. elegans* plays a special role as a model organism in neuroscience since its neural
23 network is stereotyped from animal to animal and its complete neural wiring diagram is known [Varsh-
24 ney et al., 2011]. Modern calcium imaging technology enables measurements of hundreds of these
25 neurons simultaneously [Kato et al., 2015, Nguyen et al., 2016]. The time is right to employ mod-
26 ern statistical methods to learn about the functional connectome in this system and suggest new
27 experiments.

28 Ultimately, we are interested the dynamical system that governs how neural activity evolves given its
29 history and sensory inputs. Bayesian methods are ideally suited to this goal, allowing us to represent
30 hierarchical probabilistic structures and integrate our prior knowledge about the connectome, the
31 locations of neurons, etc. Bayesian learning and inference in dynamical systems with MCMC
32 methods is well-studied, even for complicated models [De Freitas et al., 2001, Paninski et al., 2010].
33 Furthermore, hierarchical models to incorporate information from many worms are easily constructed
34 in a Bayesian framework [Gelman et al., 2014].

35 However, our efforts to integrate information across worms are complicated by a major hurdle: in
36 practice, associating recorded traces to neuron names is a painstaking, manual process. Experimenters

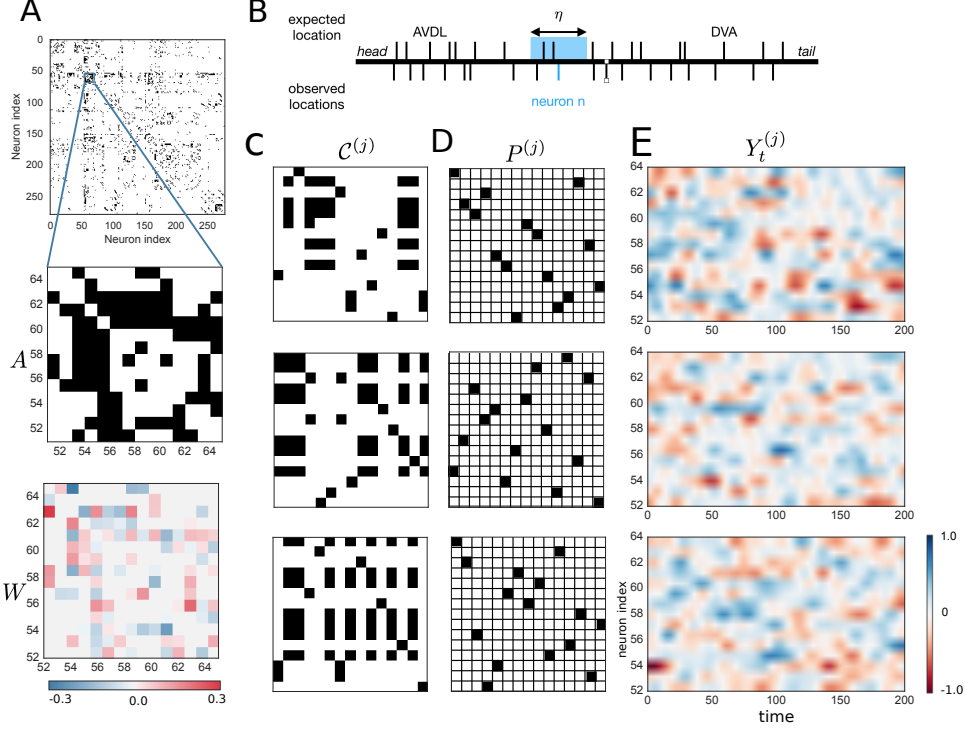


Figure 1: Hierarchical Bayesian framework. **A** We are given the actual adjacency matrix A from [Varshney et al., 2011]. The full matrix is shown (top) along with a zoom-in to 14 neurons (center). We wish to infer the corresponding weight matrix W , an example of which is shown below. **B** We also know the typical locations of the neurons [White et al., 1986, Lints et al., 2005]. Given observed locations, we constrain possible assignments to neuron identities within η of the observed location. **C** These constraints are represented as a matrix $\mathcal{C}^{(j)}$ for worm j which specifies possible assignments of observed neurons to known identities. **D** To infer the weights, we must first infer the permutation $P^{(j)}$ that matches the observed neurons in worm j to the set of known identities. **E** The observed data is a matrix $Y_t^{(j)}$ whose rows are ordered according to the order in which neurons were observed in that worm. The permutation matrix maps this to the canonical ordering of the adjacency and weight matrices. Given $\{Y_t^{(j)}\}_{j=1}^J$ and A , we infer $\{P^{(j)}\}_{j=1}^J$ and W .

consider the location of the neuron along with its pattern of activity to perform this matching, but the process is laborious and the results are prone to error. Without neuron names, we cannot represent recordings canonically or learn about how one neuron influences another. This technical problem prevents the automatic use of hierarchical methods.

We present a method for overcoming this hurdle by incorporating inference over permutations that match observed neurons (*neuron 1*, *neuron 2*, ..., *neuron N*) to known names (*AVAL*, *AVAR*, ..., *SMDR*). Once the observed neurons have been mapped to canonical names, we can learn about the shared dynamical system. To start, we focus on a simple linear autoregressive model for neural dynamics,

$$\tilde{Y}_t^{(j)} = (W \odot A) \tilde{Y}_{t-1}^{(j)} + \epsilon_t^{(j)}, \quad (1)$$

where $W \in \mathbb{R}^{N \times N}$ is the weight matrix we wish to infer; \odot denotes elementwise multiplication; $A \in \{0, 1\}^{N \times N}$ is the known adjacency matrix or connectome; and $\tilde{Y}_t^{(j)} \in \mathbb{R}^N$ is the measured neural activity at time t in worm j . The catch is that $\tilde{Y}_t^{(j)}$ is assumed to be in canonical order; i.e. in the same order as the rows and columns of W and A . What we actually observe is,

$$Y_t^{(j)} = P^{(j)} \tilde{Y}_t^{(j)}, \quad (2)$$

vectors that are permuted by matrix $P^{(j)}$. In order to learn about W , we must also infer the permutation matrices. We assume $\epsilon_t^{(j)} \sim \mathcal{N}(0, \sigma^2 I)$ with known variance, and we place a Gaussian prior on W .

The permutation matrices are constrained by side information. Specifically, we use neural position along the worm’s body to constrain the possible neural identities for a given recorded neuron. We only allow an observed neuron to be mapped to a known identity if the observed location is within η of the expected location. This is illustrated in Fig. 1B. We represent these constraints with the matrix $\mathcal{C}^{(j)}$ so that $\mathcal{C}_{mn}^{(j)} = 1$ if and only if observed neuron m is within η of where canonical neuron n ’s expected location. An example is shown in Figure 1C. We let $P^{(j)}$ have a uniform prior over the set of matrices allowable under the given constraints.

We need to perform joint posterior inference of $p(\{W, P^{(j)}\})$. MCMC with simple Metropolis-Hastings proposals is straightforward, but we found this mixed poorly in practice. Motivated by recent advances in automatic variational inference [Blei et al., 2017], we considered ways of extending this technique to permutation inference. In section 2 we detail our VI formulation and summarize the methods we developed. Then in section 3 we show that these methods outperform alternatives.

2 New methods for variational inference of latent permutations

Consider a latent variable model determined by a prior over the latent $z \sim p(z)$ and a likelihood $p(y | z)$ for the observed data y . In the VI framework, we approximate the intractable posterior $p(z | y)$ with the distribution $q \in \mathcal{Q}$ that best approximates the posterior. For tractability, we assume \mathcal{Q} is indexed by a parameter ν ; i.e. $\mathcal{Q} = \{q(z; \nu) : \nu \in \mathcal{V}\}$. The approximation is typically assessed by the Kullback-Leibler divergence between the true posterior and variational approximation,

$$\nu^* = \arg \min_{\nu \in \mathcal{V}} \text{KL}(q(z; \nu) \parallel p(z | y)). \quad (3)$$

The above problem is equivalent to maximizing the *evidence lower bound* (ELBO)

$$\mathcal{L}(\nu) \triangleq \mathbb{E}_{q(z; \nu)}[\log p(y | z)] - \text{KL}(q(z; \nu) \parallel p(z)), \quad (4)$$

with respect to ν . We typically maximize equation (4) with stochastic optimization methods [Kushner and Yin, 1987]: specifically, we approximate the expectations in (4) with Monte Carlo estimates and optimize the ELBO with stochastic gradient ascent. One critical component is the choice of the Monte Carlo approximation. Perhaps the most common choice is through the so called *score function estimator*. Unfortunately, this estimator, also referred to as REINFORCE [Williams, 1992], cannot be applied to permutations, since it involves the evaluation and differentiation of a likelihood which is intractable for any non-trivial distribution over permutations (computing the partition function involves a summation over $N!$ terms).

The re-parameterization trick Kingma and Welling [2013] offers an appealing alternatives. If z can be written as a differentiable function of a noise distribution and the parameters—i.e. if for certain f and $\xi \sim p(\xi)$ one has $z = f(\xi, \nu)$ —then we can write the expectation with respect to $q(z)$ as an expectation with respect to $p(\xi)$ and bring the gradient inside the expectation. In the case of discrete random variables a re-parameterization always exists and it is given by the *Gumbel trick* [Papandreou and Yuille, 2011, Balog et al., 2017], which states that one can sample from any discrete distribution by perturbing each potential with Gumbel i.i.d noise, and then finding the configuration with the maximum value. Unfortunately, the underlying f to this re-parameterization is the non-differentiable $\arg \max$ operator, precluding the use of gradient descent methods.

Jang et al. [2016] and Maddison et al. [2016] proposed a solution to this problem, replacing the $\arg \max$ by a temperature-dependent softmax approximation, which in the limit converges to the original $\arg \max$. By combining the Gumbel trick with the softmax approximation, they conceived the *Concrete* or *Gumbel-Softmax* distribution, and obtained explicit distribution formulae. Then, they showed how to perform variational inference in discrete latent variable models using the re-parameterization trick and gradient descent. They replaced the original ELBO with a surrogate appropriate for their continuous relaxation. As long as the temperature is chosen in a reasonable range—not too high as it would lead to a degenerate distribution in the simplex; but also not too low, to avoid too high variances of the gradients—the method works well.

We developed three methods for extending the Gumbel-softmax method to permutations. We name them *stick-breaking*, *rounding* and *Gumbel-Sinkhorn* methods. We refer the reader to sections 3.1 and 3.2 of Linderman et al. [2017] and section 4 of Anonymous [2018] for details, respectively. Here we briefly summarize them: the primary geometric object is the Birkhoff polytope, the convex

hull of permutation matrices, and the analog of the probability simplex in the discrete case. We generalize the standard stick-breaking construction on the simplex to stick-breaking of the Birkhoff polytope. We show how to consistently “break the stick” while satisfying both the row and column constraints that characterize a doubly stochastic matrix. For the rounding construction, we start with a noise distribution and force it to be close to permutation matrices by rounding them towards the extreme-points of the Birkhoff polytope (i.e. permutation matrices). Finally, for the Gumbel-Sinkhorn method we notice that the so-called *Sinkhorn operator*, or infinite and successive row and column normalization of a matrix, is a natural extension of the softmax operator. With this, we are able to conceive the Gumbel-Sinkhorn distribution, which approximates the sampling of a relevant discrete distribution. Importantly, while stick-breaking and rounding yield explicit densities, Gumbel-Sinkhorn does not. However, there are ways to circumvent this difficulty, and overall we observe the latter performs the best.

3 Results

We compared against three methods: (i) naive variational inference, where we do not enforce the constraint that $X^{(j)}$ be a permutation and instead treat each row of $X^{(j)}$ as a Dirichlet distributed vector; (ii) MCMC, where we alternate between sampling from the conditionals of W (Gaussian) and $X^{(j)}$, from which one can sample by proposing local swaps, as described in Diaconis [2009], and (iii) maximum a posteriori estimation (MAP). Our MAP algorithm alternates between the optimizing estimate of W given $\{X^{(m)}, Y^{(m)}\}$ using linear regression and finding the optimal $X^{(j)}$. The second step requires solving a quadratic assignment problem (QAP) in $X^{(j)}$; that is, it can be expressed as $\text{Tr}(AXBX^T)$ for matrices A, B . We used the QAP solver proposed by Vogelstein et al. [2015].

We found that our method outperforms each baseline. Specifically, we show that our method outperforms alternatives when there are many possible candidates (Table 1) and when only a small proportion of neurons are known with certitude (Table 2). Altogether, these results indicate our method enables a more efficient use of information than its alternatives. This is consistent with other results showing faster convergence of variational inference over MCMC [Blei et al., 2017], especially with simple Metropolis-Hastings proposals. We conjecture that MCMC could eventually obtain similar if not better results, if current local proposals—swapping pairs of labels—were replaced by more involved ones.

	10		30		45		60	
	1 worm	4 worms	1 Worm	4 worms	1 worm	4 worms	1 worms	4 worms
NAIVE VI	.34	.32	.16	.16	.13	.12	.11	.12
MAP	.34	.32	.17	.17	.14	.13	.13	.12
MCMC	.34	.65	.18	.28	.14	.17	.13	.15
VI	.79	.94	.4	.69	.25	.51	.21	.44

Table 1: Accuracy in the C.elegans neural identification problem, for varying mean number of candidate neurons (10, 30, 45, 60) and number of worms.

	40.%		30.%		20.%		10.%	
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.1$	$\eta = 0.2$
Naive VI .43	.41	.33	.31	.23	.22	.12	.1	
MAP	.42	.41	.33	.32	.23	.22	.12	.11
MCMC	.85	.80	.52	.46	.3	.26	.15	.12
VI	.97	.96	.92	.84	.74	.58	.44	.23

Table 2: Accuracy in inferring true neural identity for different of proportion of known neurons, and two values of η .

132 References

- 133 Anonymous. Learning latent permutations with gumbel-sinkhorn networks. *International Conference*
134 *on Learning Representations*, 2018.
- 135 M. Balog, N. Tripuraneni, Z. Ghahramani, and A. Weller. Lost relatives of the gumbel trick. *arXiv*
136 *preprint arXiv:1706.04161*, 2017.
- 137 D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians.
138 *Journal of the American Statistical Association*, 2017.
- 139 N. De Freitas, C. Andrieu, P. Højén-Sørensen, M. Niranjana, and A. Gee. Sequential monte carlo
140 methods for neural networks. In *Sequential Monte Carlo methods in practice*, pages 359–379.
141 Springer, 2001.
- 142 P. Diaconis. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical*
143 *Society*, 46(2):179–205, 2009.
- 144 A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data*
145 *analysis*, volume 2. CRC press Boca Raton, FL, 2014.
- 146 E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint*
147 *arXiv:1611.01144*, 2016.
- 148 S. Kato, H. Kaplan, T. Schrödel, S. Skora, T. Lindsay, E. Yemini, S. Lockery, and M. Zimmer. Global
149 brain dynamics embed the motor command sequence of *Caenorhabditis elegans*. *Cell*, 163(3):656 –
150 669, 2015. ISSN 0092-8674.
- 151 D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*,
152 2013.
- 153 H. Kushner and G. Yin. Stochastic approximation algorithms for parallel and distributed processing.
154 *Stochastics: An International Journal of Probability and Stochastic Processes*, 22(3-4):219–250,
155 1987.
- 156 S. W. Linderman, G. E. Mena, H. Cooper, L. Paninski, and J. P. Cunningham. Reparameterizing the
157 Birkhoff polytope for variational permutation inference. *arXiv preprint arXiv:1710.09508*, 2017.
- 158 R. Lints, Z. F. Altun, H. Weng, T. Stephney, G. Stephney, M. Volaski, and D. H. Hall. WormAtlas
159 Update. 2005.
- 160 C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of
161 discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
- 162 J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M.
163 Leifer. Whole-brain calcium imaging with cellular resolution in freely behaving *Caenorhabditis*
164 *elegans*. *Proceedings of the National Academy of Sciences*, 113(8):E1074–E1081, 2016.
- 165 L. Paninski, Y. Ahmadian, D. G. Ferreira, S. Koyama, K. R. Rad, M. Vidne, J. Vogelstein, and W. Wu.
166 A new look at state-space models for neural data. *Journal of computational neuroscience*, 29(1-2):
167 107–126, 2010.
- 168 G. Papandreou and A. L. Yuille. Perturb-and-map random fields: Using discrete optimization to learn
169 and sample from energy models. In *Computer Vision (ICCV), 2011 IEEE International Conference*
170 *on*, pages 193–200. IEEE, 2011.
- 171 L. R. Varshney, B. L. Chen, E. Paniagua, D. H. Hall, and D. B. Chklovskii. Structural properties of
172 the *Caenorhabditis elegans* neuronal network. *PLoS Computational Biology*, 7(2):e1001066, 2011.
- 173 J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind,
174 R. J. Vogelstein, and C. E. Priebe. Fast approximate quadratic programming for graph matching.
175 *PLOS one*, 10(4):e0121002, 2015.
- 176 J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of
177 the nematode *Caenorhabditis elegans*: the mind of a worm. *Phil. Trans. R. Soc. Lond*, 314:1–340,
178 1986.

179 R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement
180 learning. *Machine Learning*, 8(3–4):229–256, 1992.