Cambridge Hackathon

Benjamin D. Pedigo

(he/him)

NeuroData lab

Johns Hopkins University - Biomedical Engineering

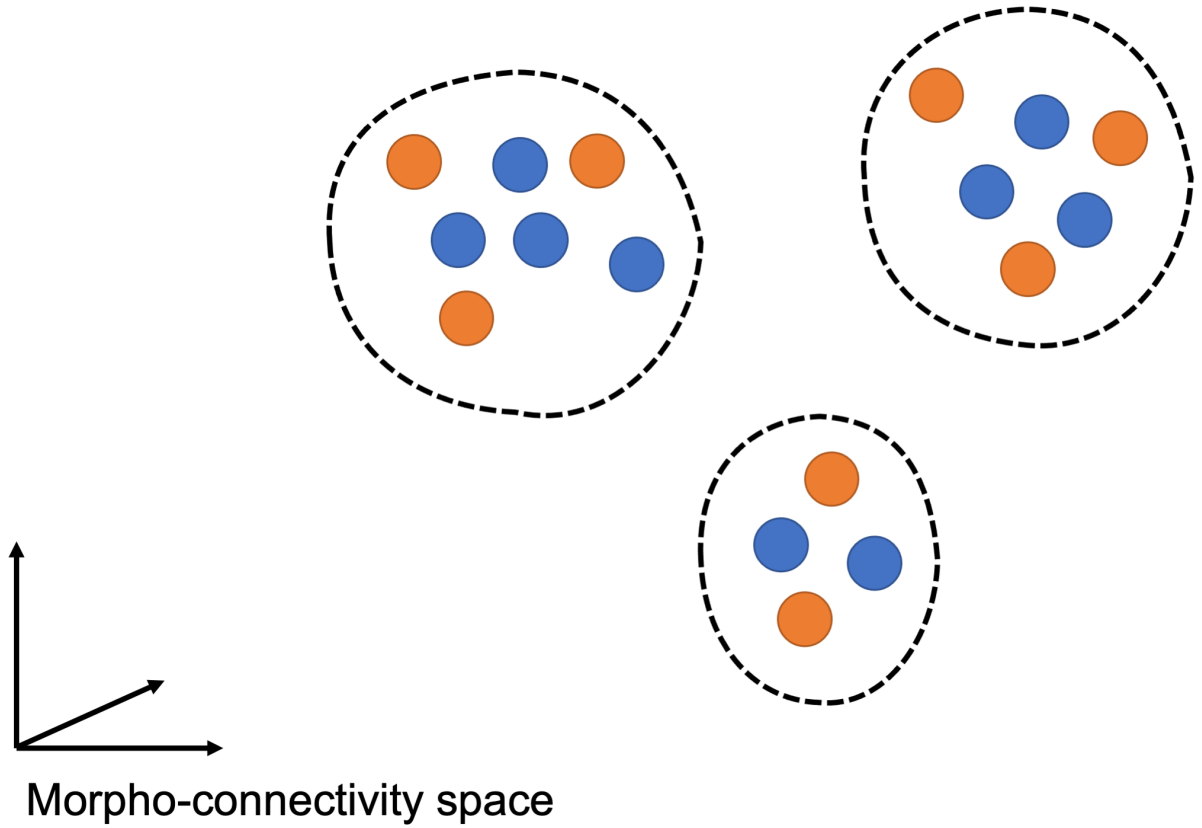
 *bpedigo@jhu.edu*

 *@bdpedigo (Github)*

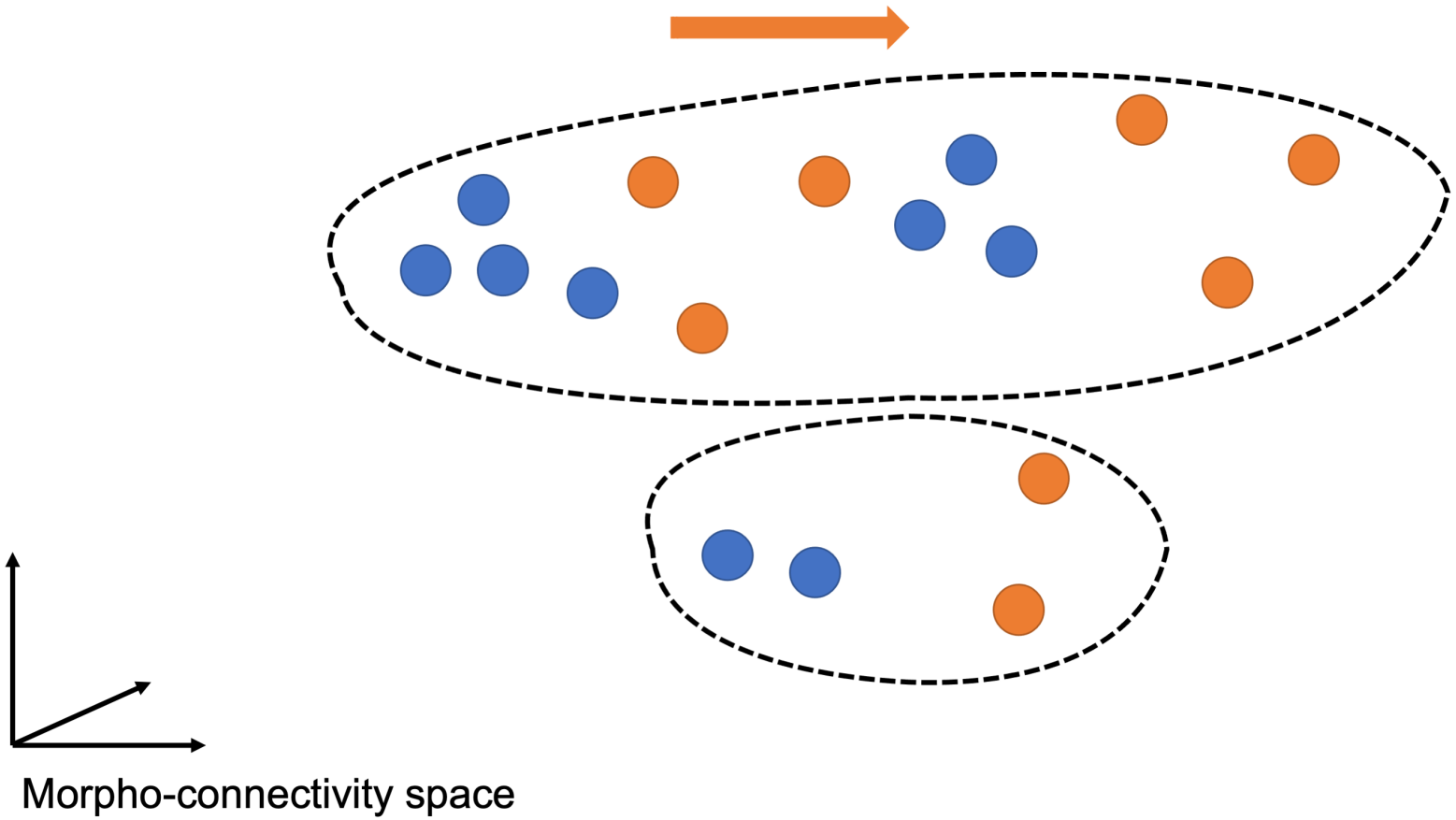
 *@bpedigod (Twitter)*

 *bdpedigo.github.io*

Cell types



Cell types and "distribution shift"



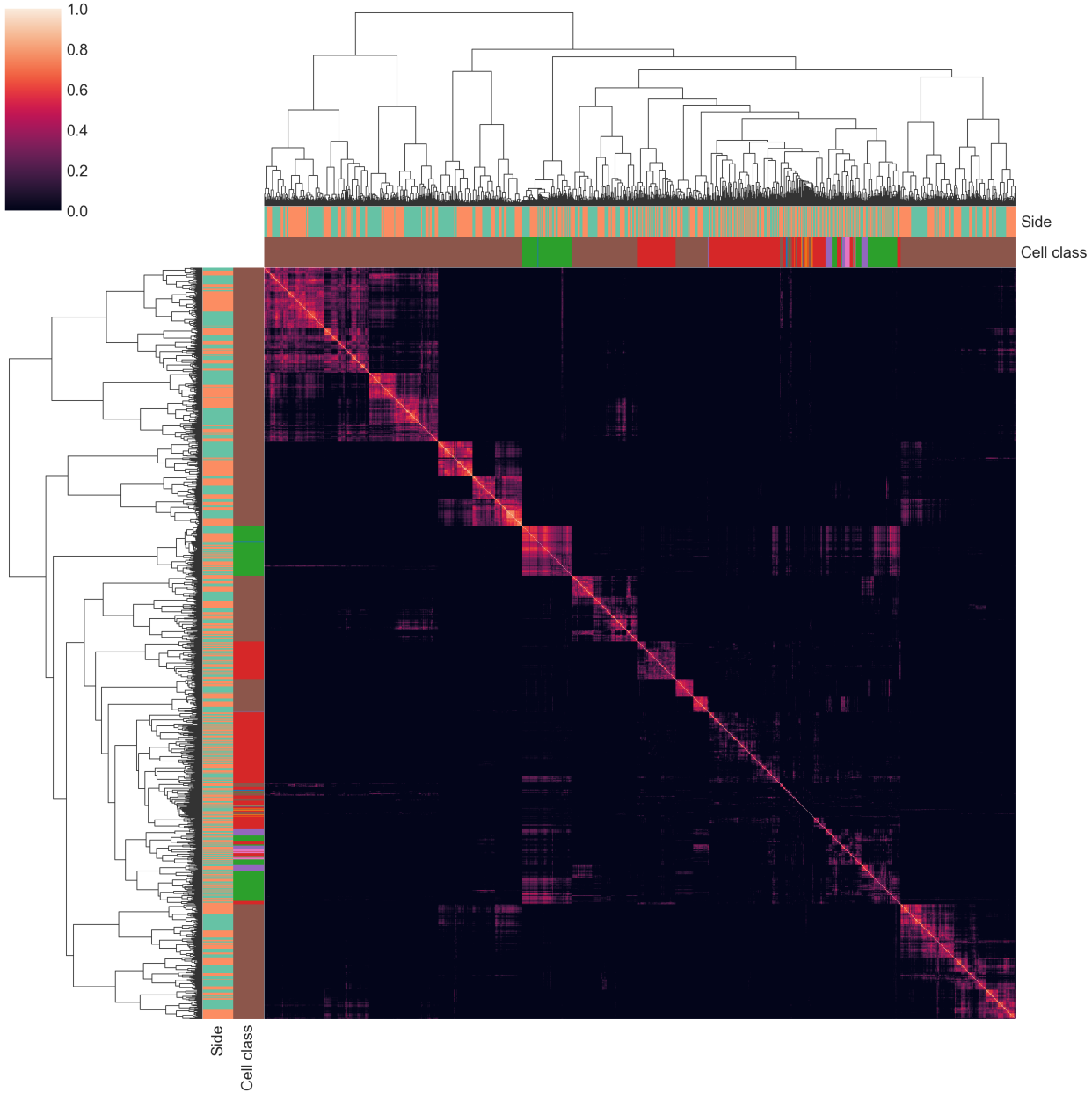
Matters for both "inductive" and "de novo" cell typing

- E.g. mapping labels from one dataset onto another via a matching, or
- Creating new labels by clustering on a unified representation of datasets

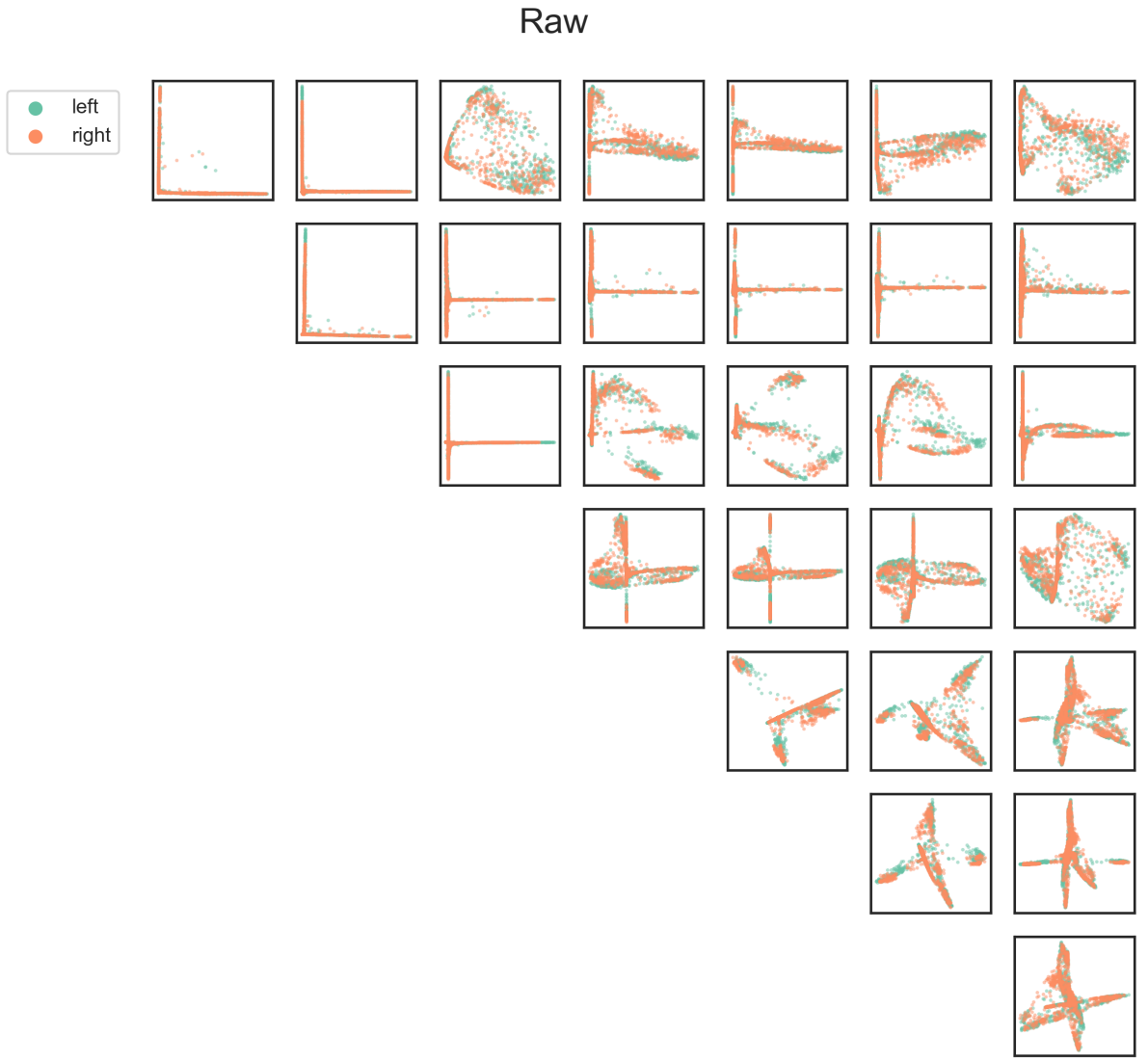
Antennal lobe as a test case

- Using FlyWire data
- Trying to match left/right
- ~1750 neurons per side
- Has good labels to use for evaluation

NBLAST clustering suggests this could be happening



NBLAST embedding



How to line things up, based on NBLAST?

Say we have sets of objects $I = \{i : i = 1, \ldots, n\}$ and $J = \{j : j = 1, \ldots, m\}$

For example, I is the set of neurons on the left, J those on the right.

Say S is a $n \times m$ matrix such that S_{ij} has the NBLAST score between neuron i (left) and neuron j (right)

Linear assignment problem

$$\max_P trace(SP^T) = \max_P \sum_{ij} S_{ij}P_{ij}$$

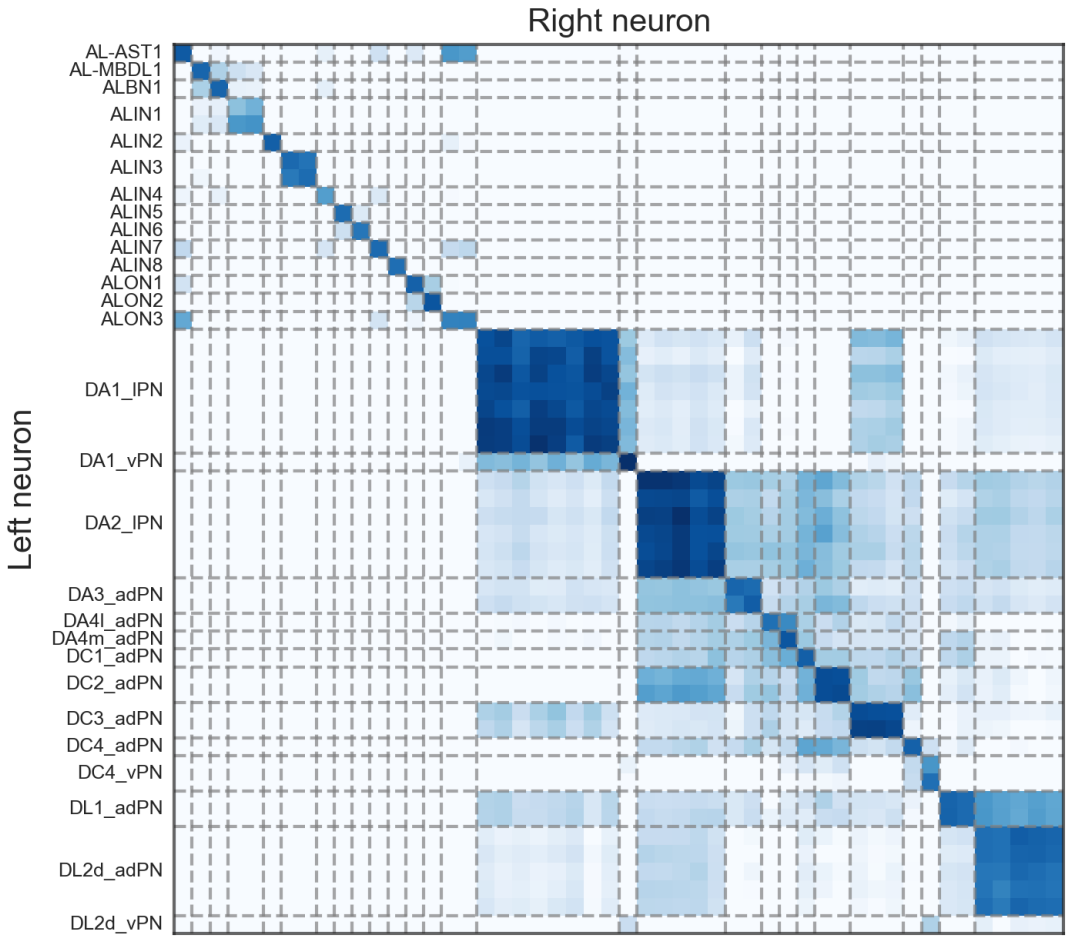
- P is a permutation matrix.
- P_{ij} is 1 if $i \leftrightarrow j$, 0 otherwise.

Intuition:

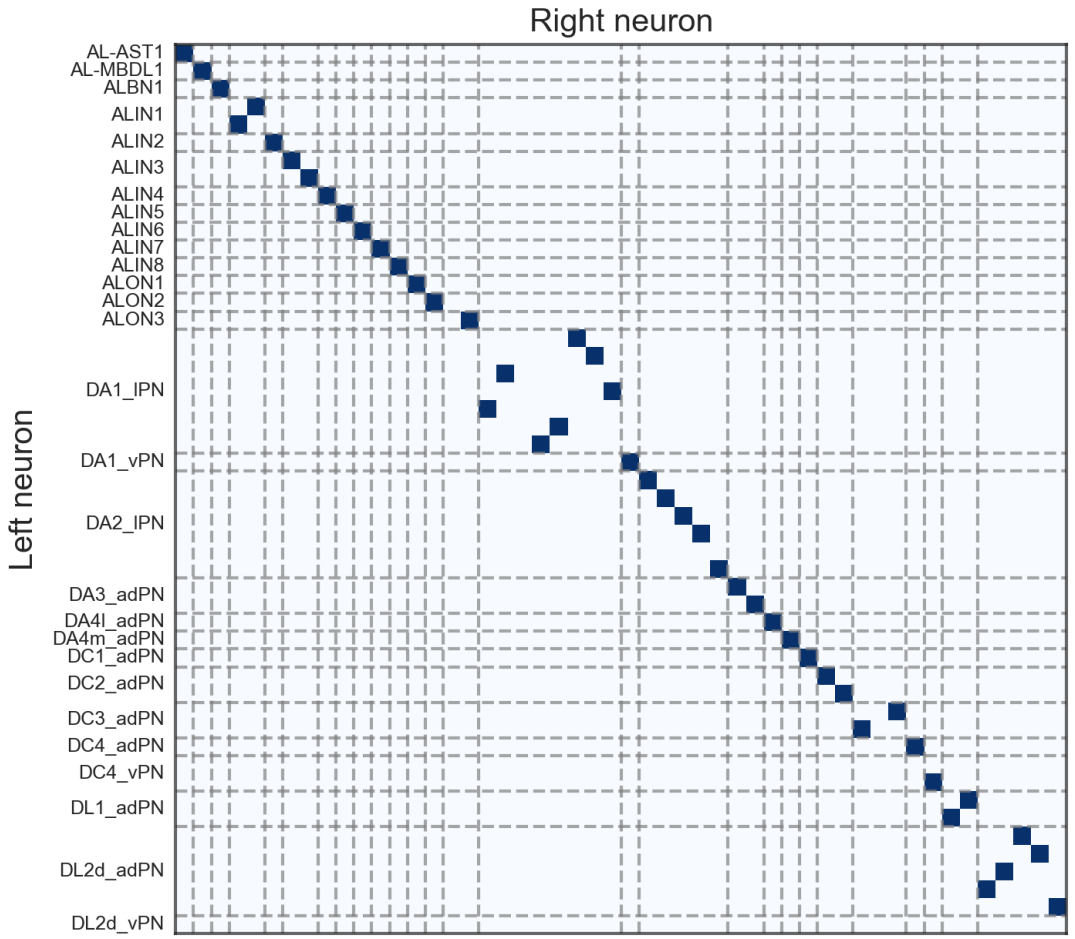
Maximize the total NBLAST scores of neurons which are matched, over the set of all matchings.

P^T reshuffles the columns of S to make the diagonal big

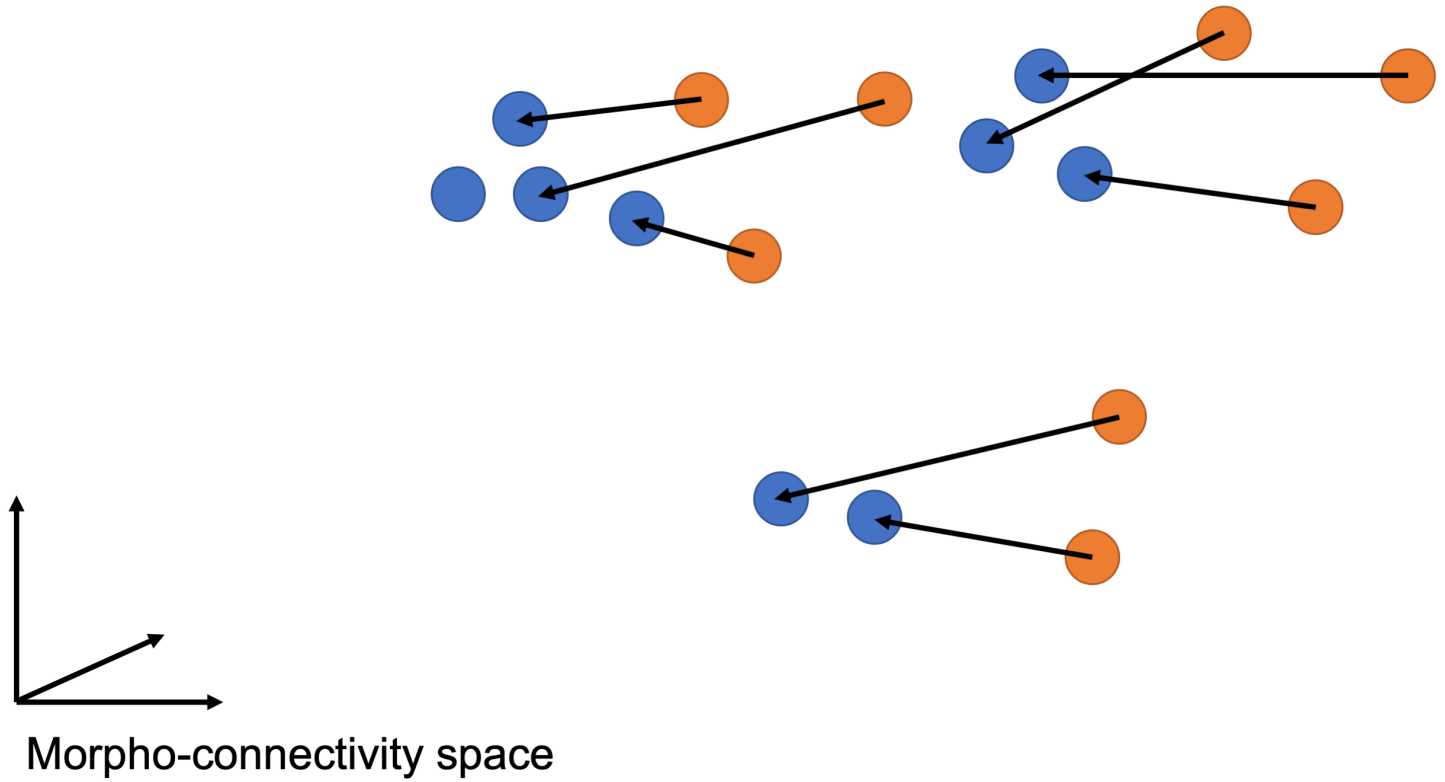
NBLAST



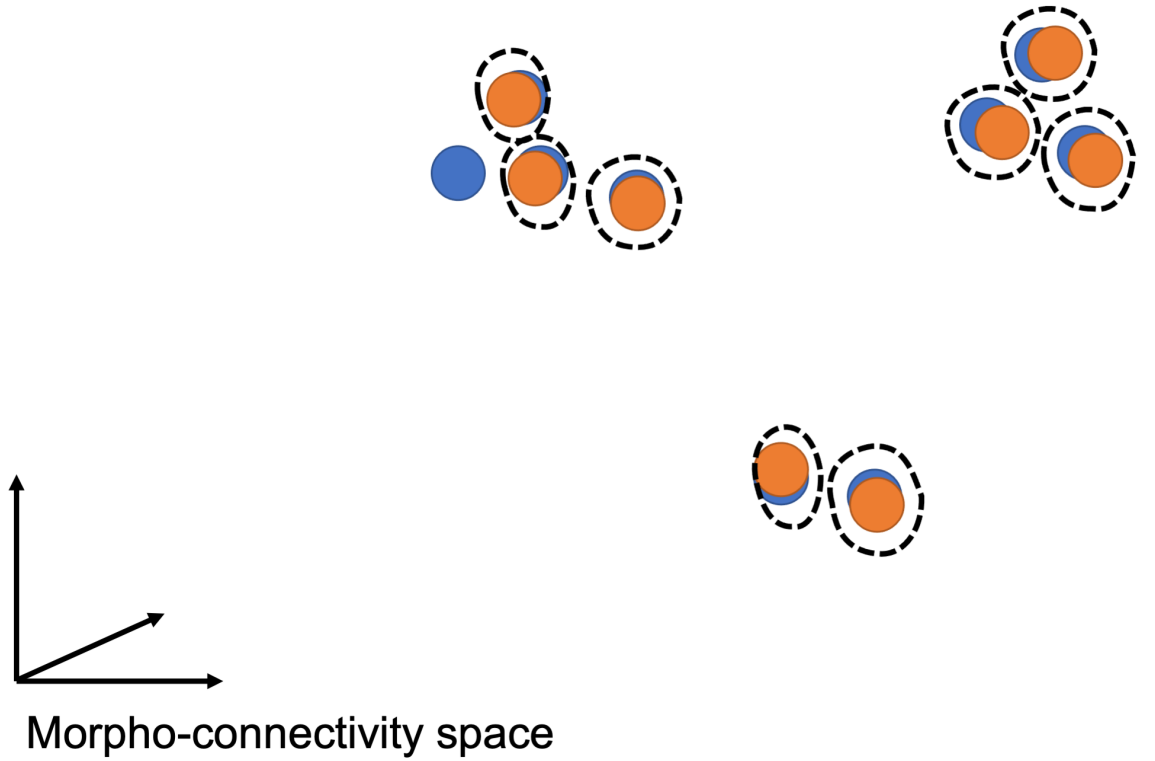
LAP solution



Issue with a "hard" matching?



Issue with a "hard" matching?

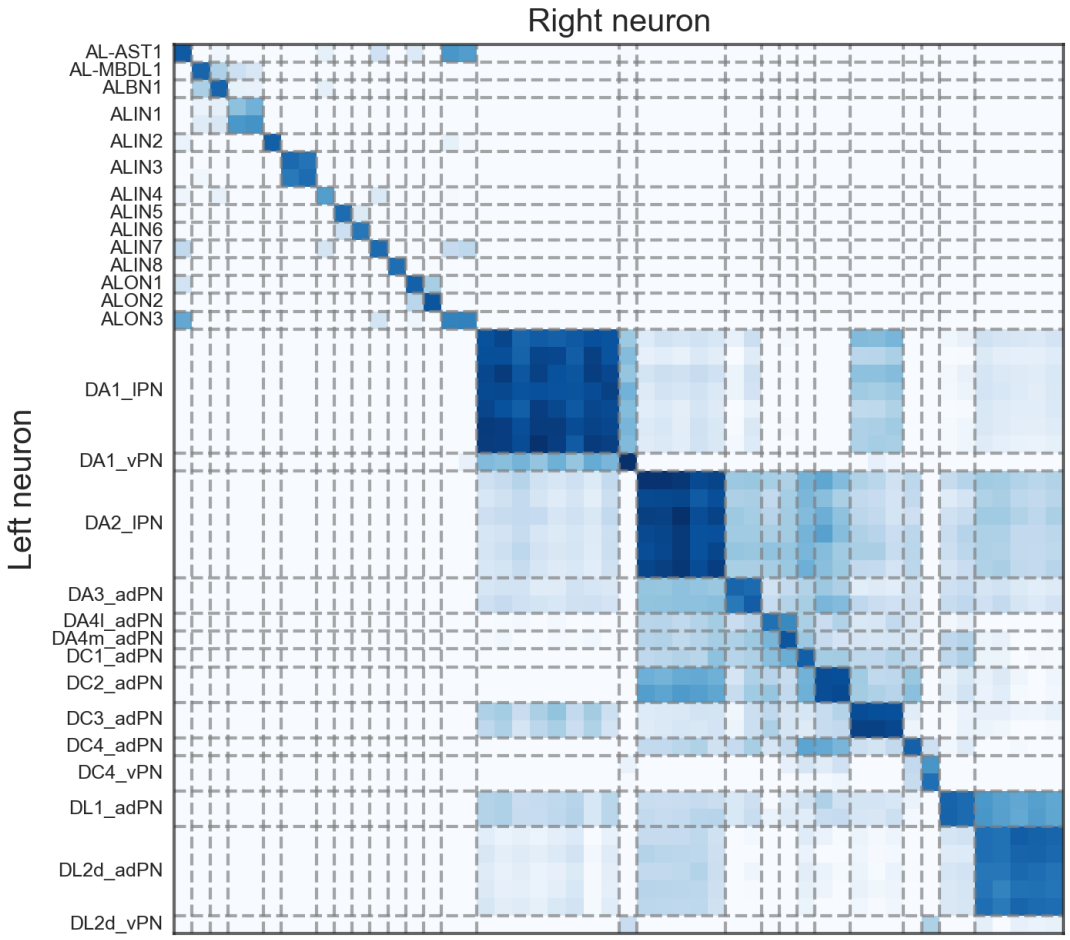


Smoother matchings

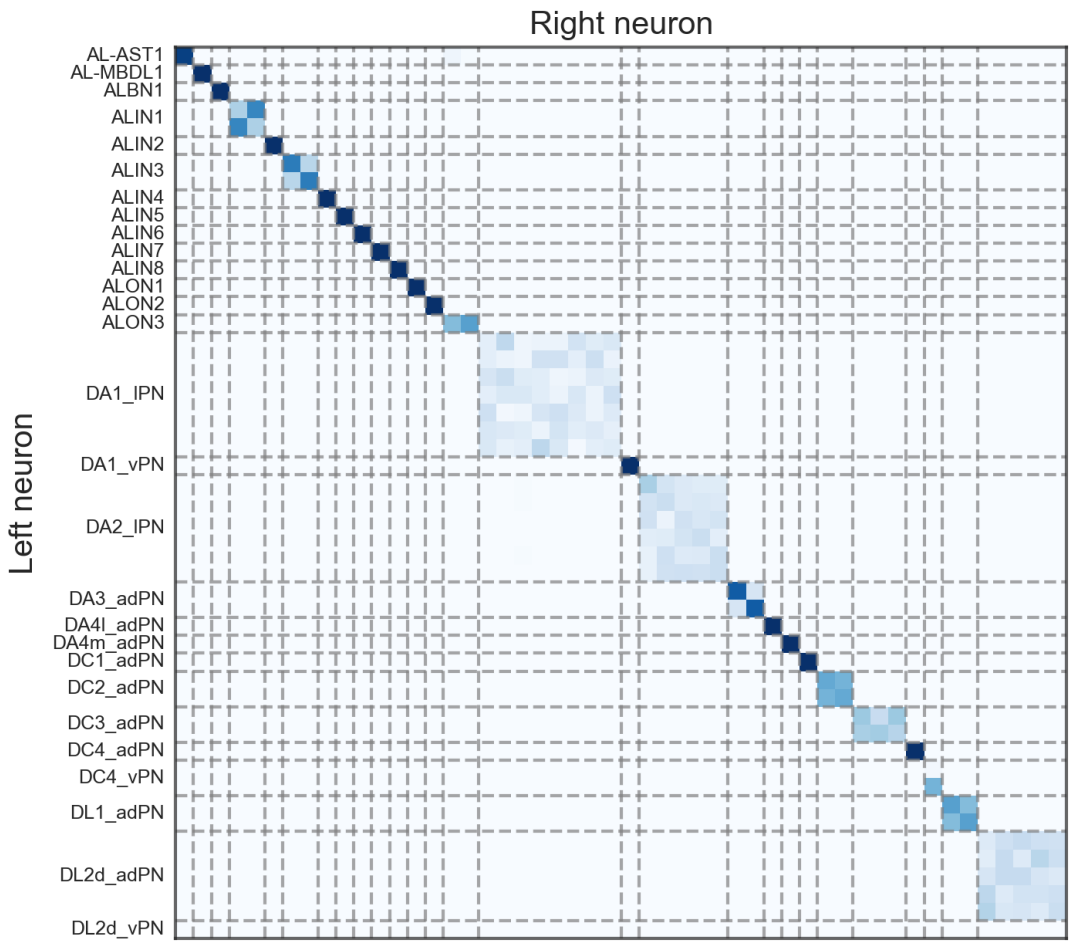
Or, how stable are these matchings?

- Add i.i.d. Gaussian noise to NBLAST matrix, $\tilde{S}_{ij} = S_{ij} + Normal(0, 0.05)$
- Run linear assignment problem, get P^*
- Take the average of 100 runs of the above.

NBLAST



Noisy EMD solution ($\sigma = 0.05$)



Regularized optimal transport

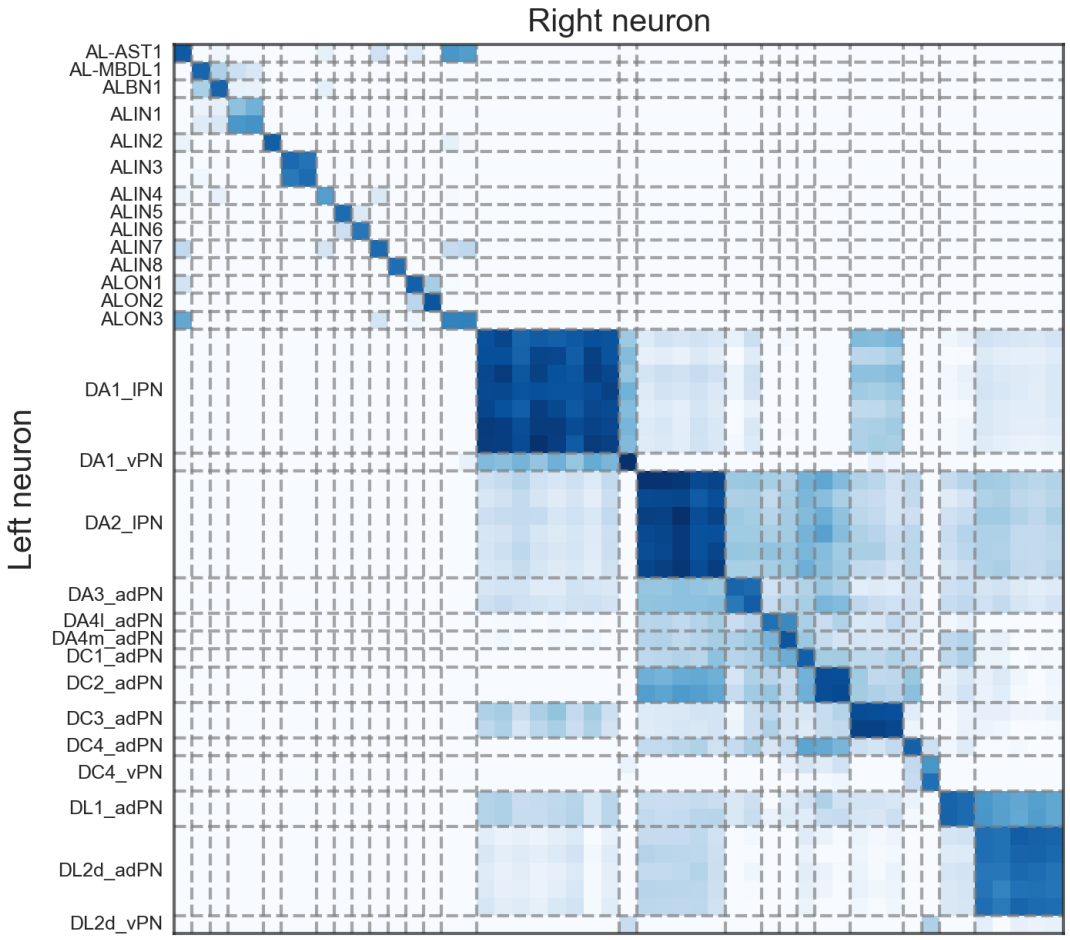
$$\max_{D} trace(SD^T) + \lambda\Omega(D)$$

- D is a transportation matrix, i.e. rows/columns sum to 1, and D_{ij} represents the amount of "flow" or matching weight from i to j.
- $\Omega(D)$ is a regularizer which promotes "smooth" solutions (i.e. not 0-1),
- and λ is a weight on the regularization.

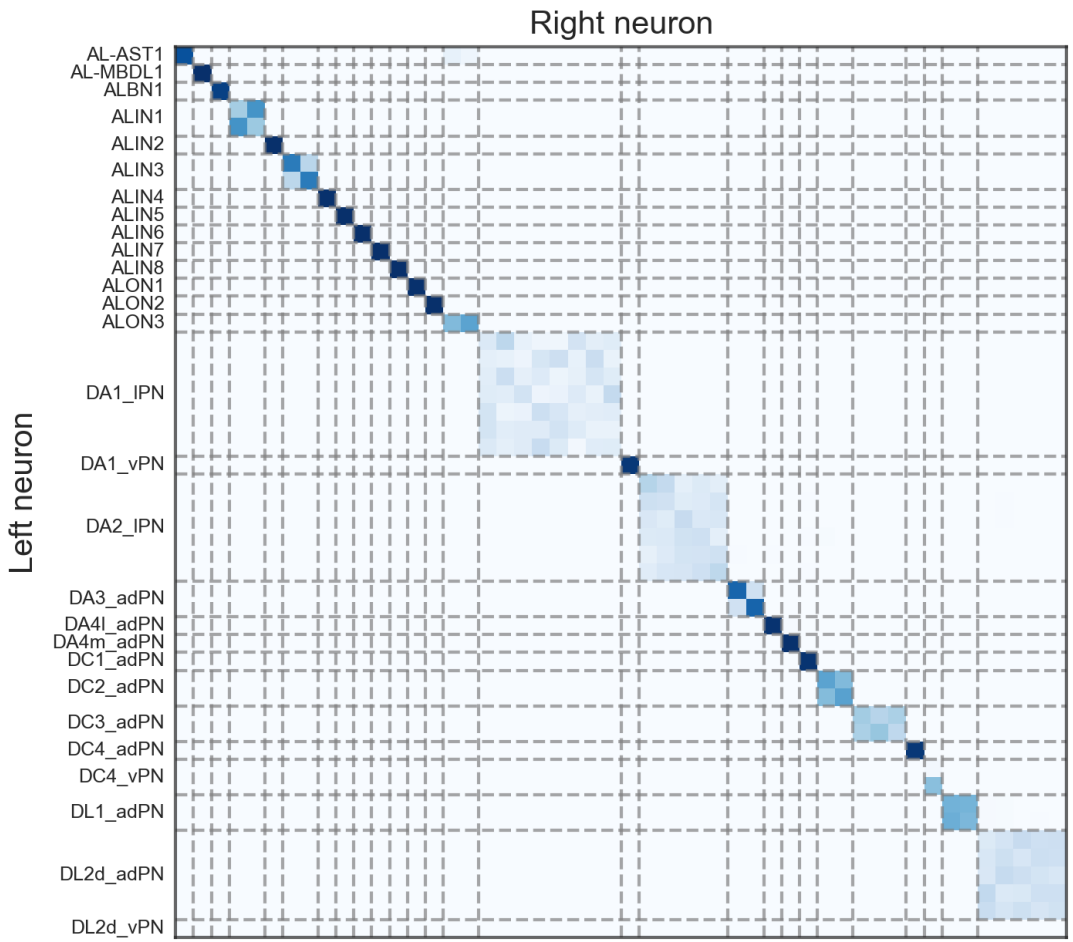
Intuition

Maximize the NBLAST scores of soft-matched neurons, weighted by how strongly those neurons are matched, over the set of (somewhat smooth) soft matchings

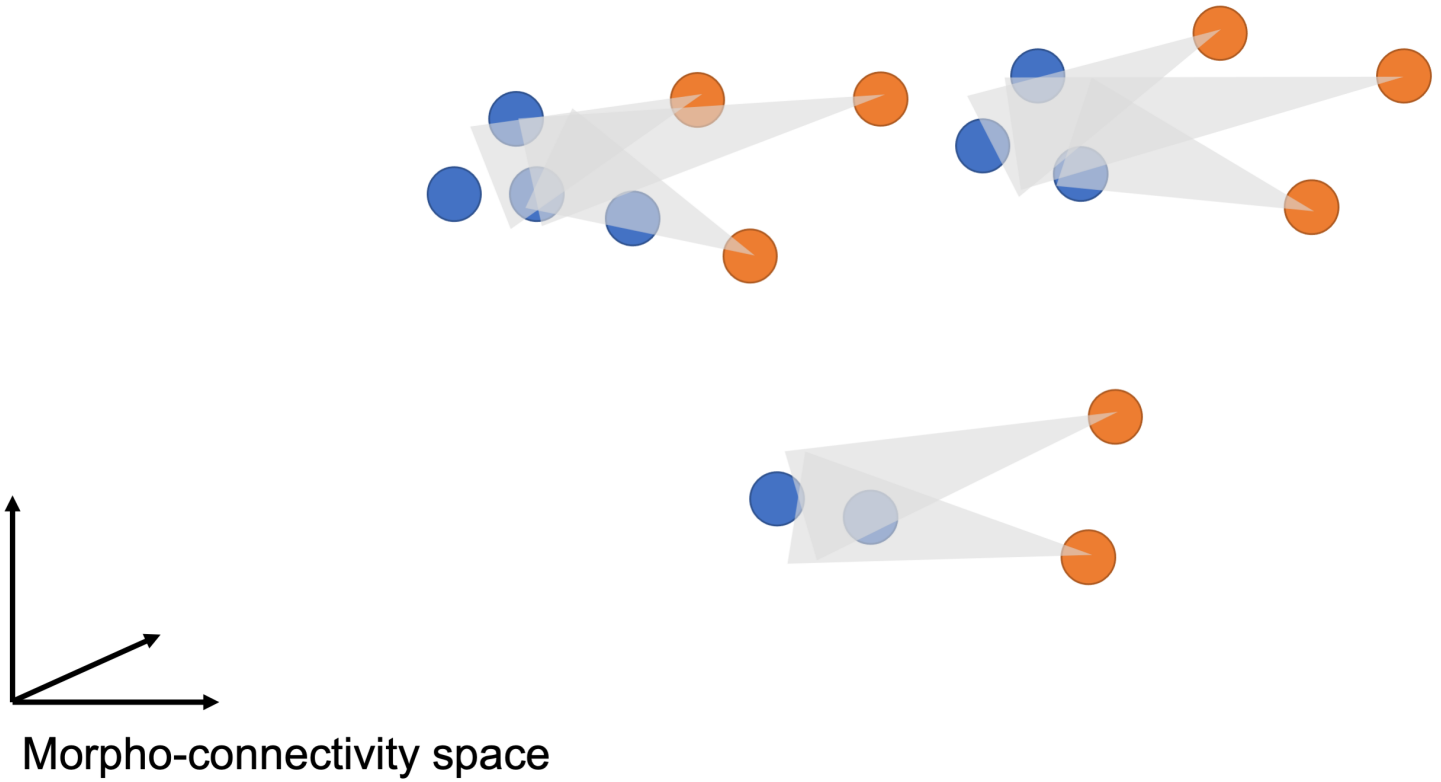
NBLAST



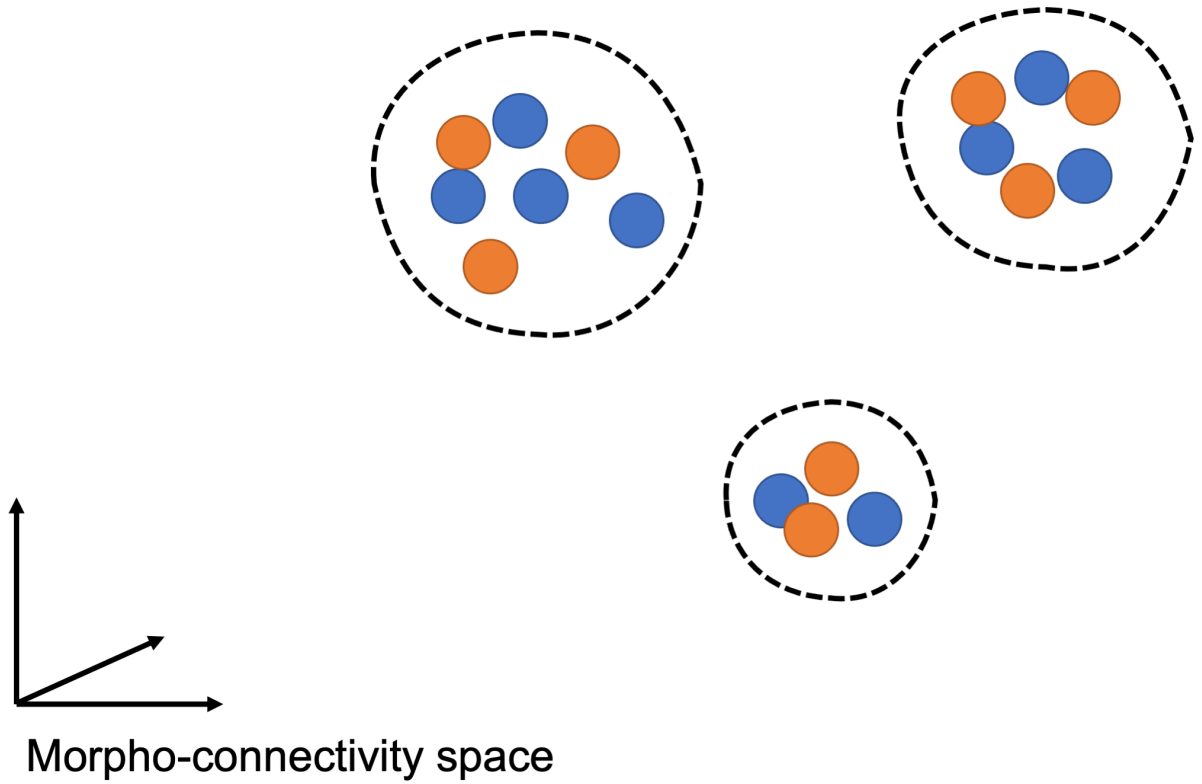
Sinkhorn solution ($\lambda = 0.03$)



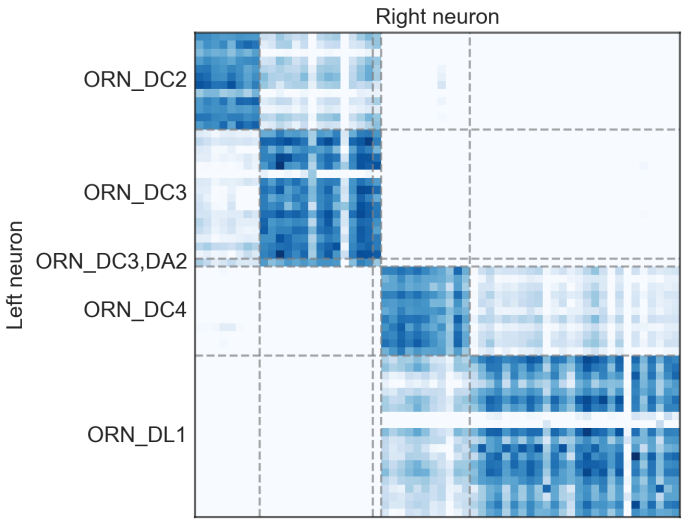
Benefit (hopefully) of a smoother matching



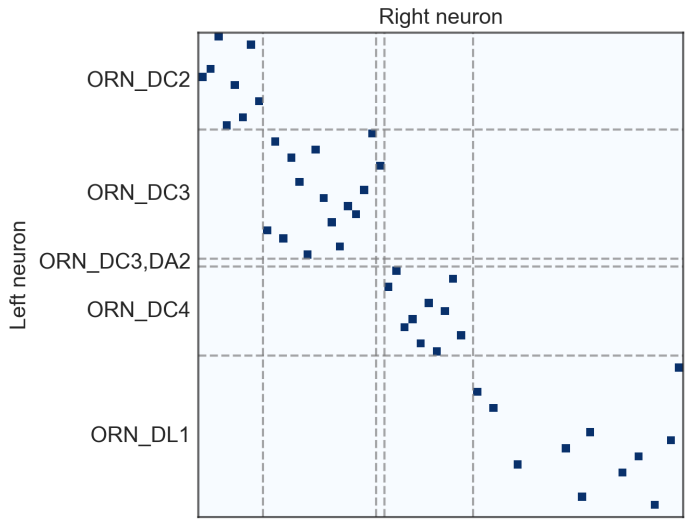
Benefit (hopefully) of a smoother matching



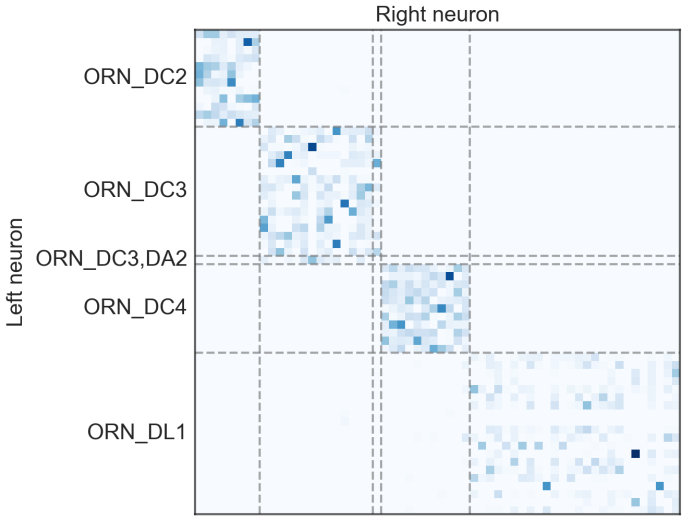
NBLAST



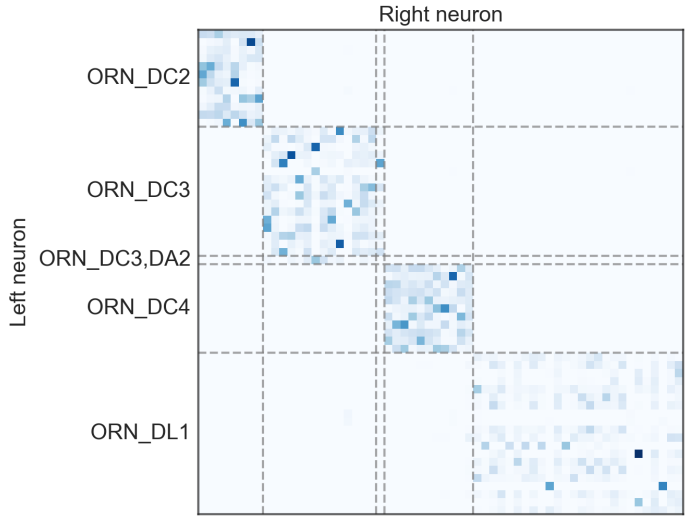
LAP solution

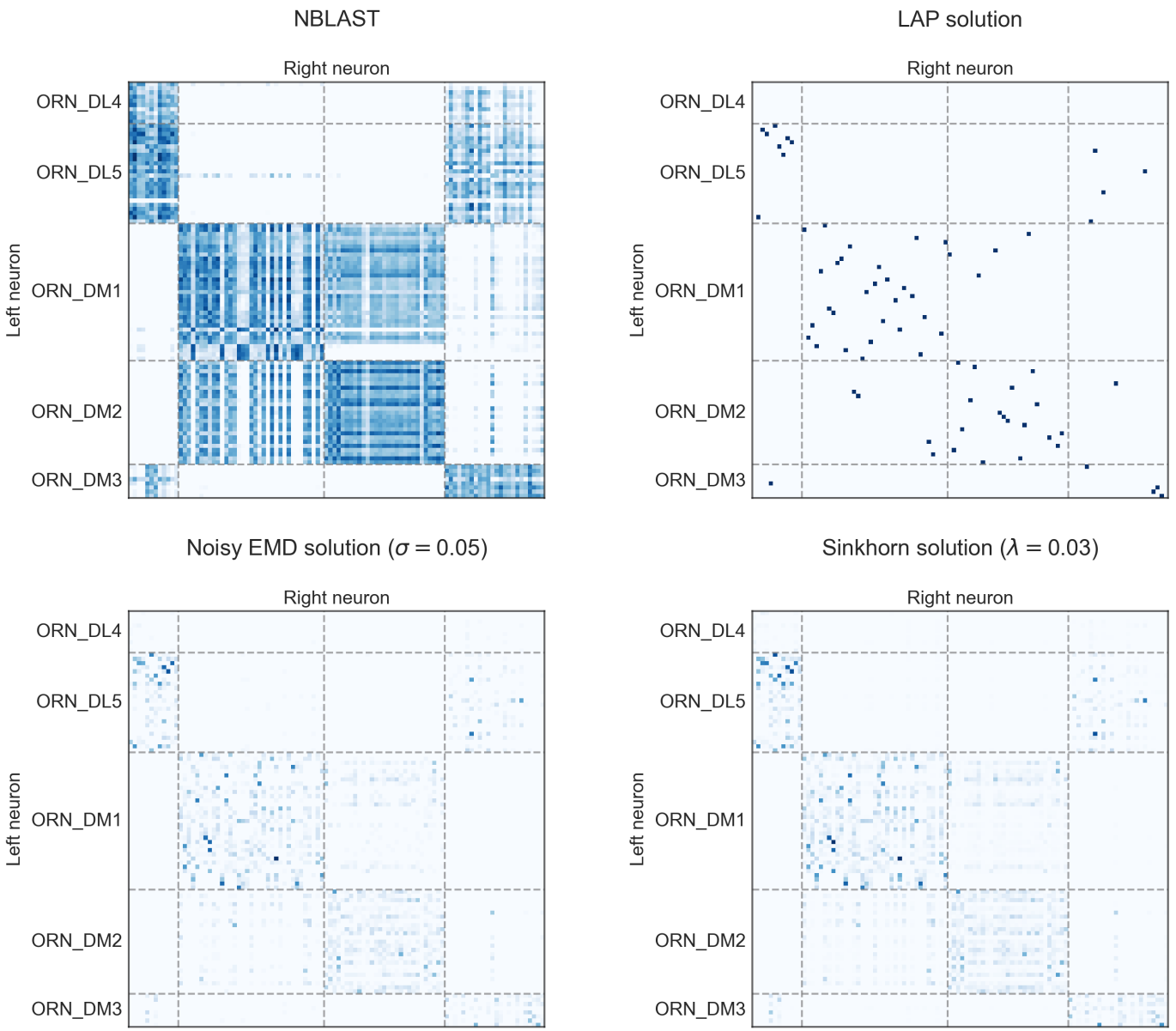


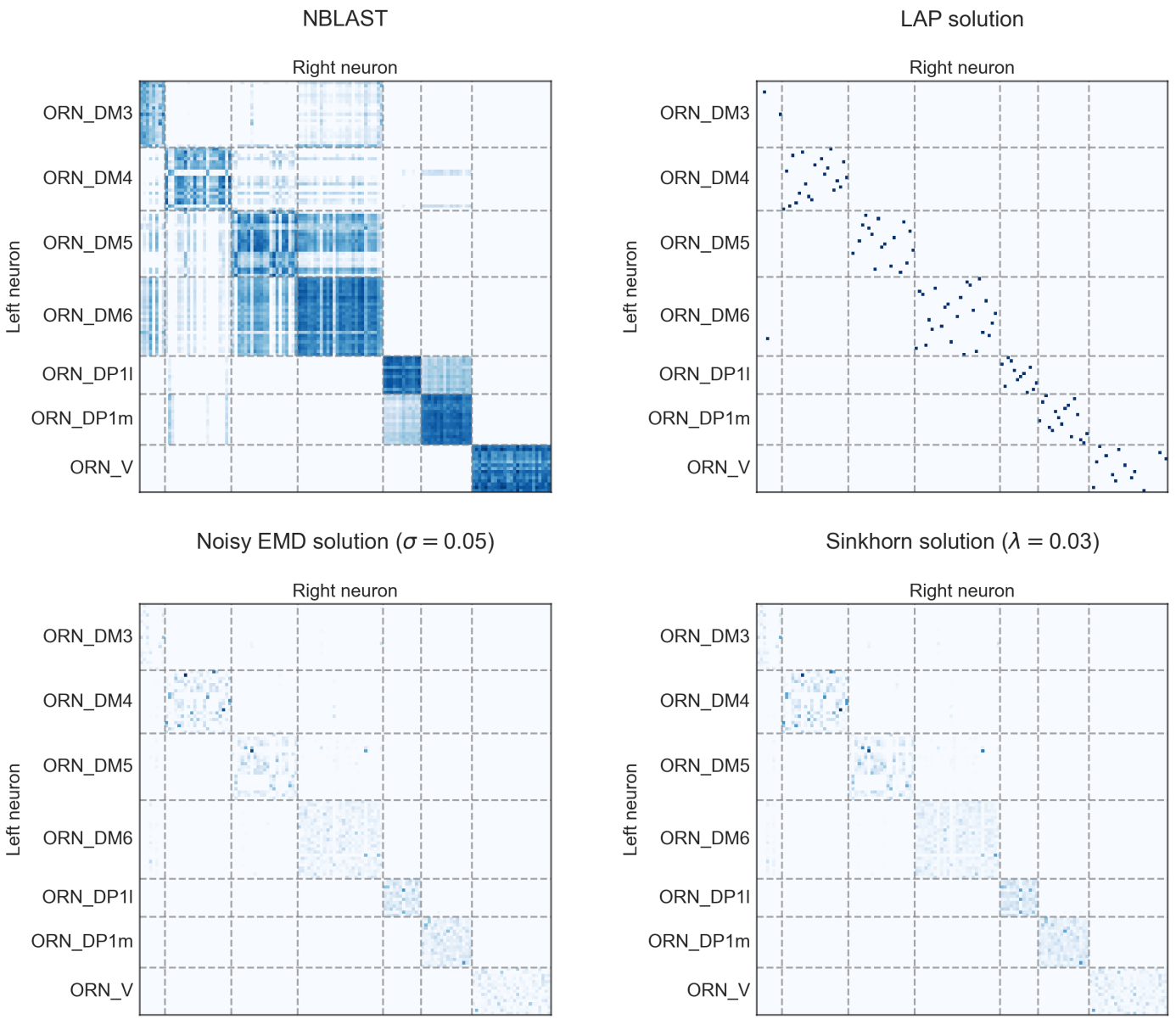
Noisy EMD solution ($\sigma = 0.05$)



Sinkhorn solution ($\lambda = 0.03$)




NBLAST
Right neuron
Left neuron
ORN_DL4
ORN_DL5
ORN_DM1
ORN_DM2
ORN_DM3
LAP solution
Right neuron
Left neuron
ORN_DL4
ORN_DL5
ORN_DM1
ORN_DM2
ORN_DM3
Noisy EMD solution ($\sigma = 0.05$)
Right neuron
Left neuron
ORN_DL4
ORN_DL5
ORN_DM1
ORN_DM2
ORN_DM3
Sinkhorn solution ($\lambda = 0.03$)
Right neuron
Left neuron
ORN_DL4
ORN_DL5
ORN_DM1
ORN_DM2
ORN_DM3


NBLAST
Right neuron
Left neuron
ORN_DM3
ORN_DM4
ORN_DM5
ORN_DM6
ORN_DP1l
ORN_DP1m
ORN_V
LAP solution
Right neuron
Left neuron
ORN_DM3
ORN_DM4
ORN_DM5
ORN_DM6
ORN_DP1l
ORN_DP1m
ORN_V
Noisy EMD solution ($\sigma = 0.05$)
Right neuron
Left neuron
ORN_DM3
ORN_DM4
ORN_DM5
ORN_DM6
ORN_DP1l
ORN_DP1m
ORN_V
Sinkhorn solution ($\lambda = 0.03$)
Right neuron
Left neuron
ORN_DM3
ORN_DM4
ORN_DM5
ORN_DM6
ORN_DP1l
ORN_DP1m
ORN_V

What if we want to include connectivity?

Cosine similarity:

for two vectors, a and b...

$$cos(a,b) = \frac{< a, b >}{\|a\|\|b\|} = \frac{a^T b}{\|a\|\|b\|}$$

if we have a bunch of vectors stored in the matrices A and B, then

$$C = A^T B$$

since $C_{ij} = \sum_k a_{ki} b_{kj}$

Graph matching

$$C = A^T I B$$

where I is the identity matrix - this represents a belief about the permutation of rows of B with respect to A... so more generally, could write:

$$C = A^T P B$$

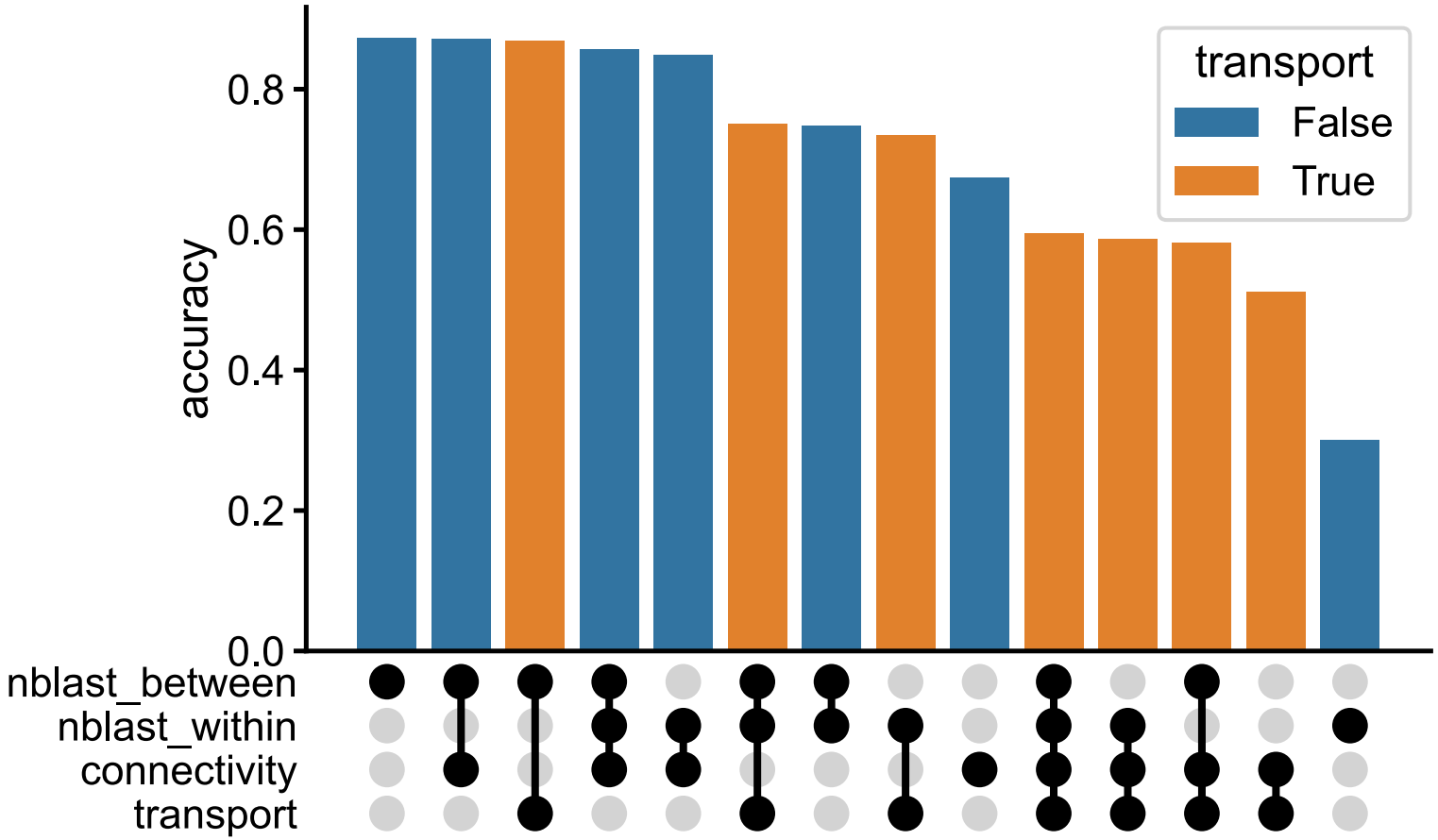
Much like before, if we want to measure the "matchedness"

$$\max_P trace(CP^T) = trace(A^T P B P^T)$$

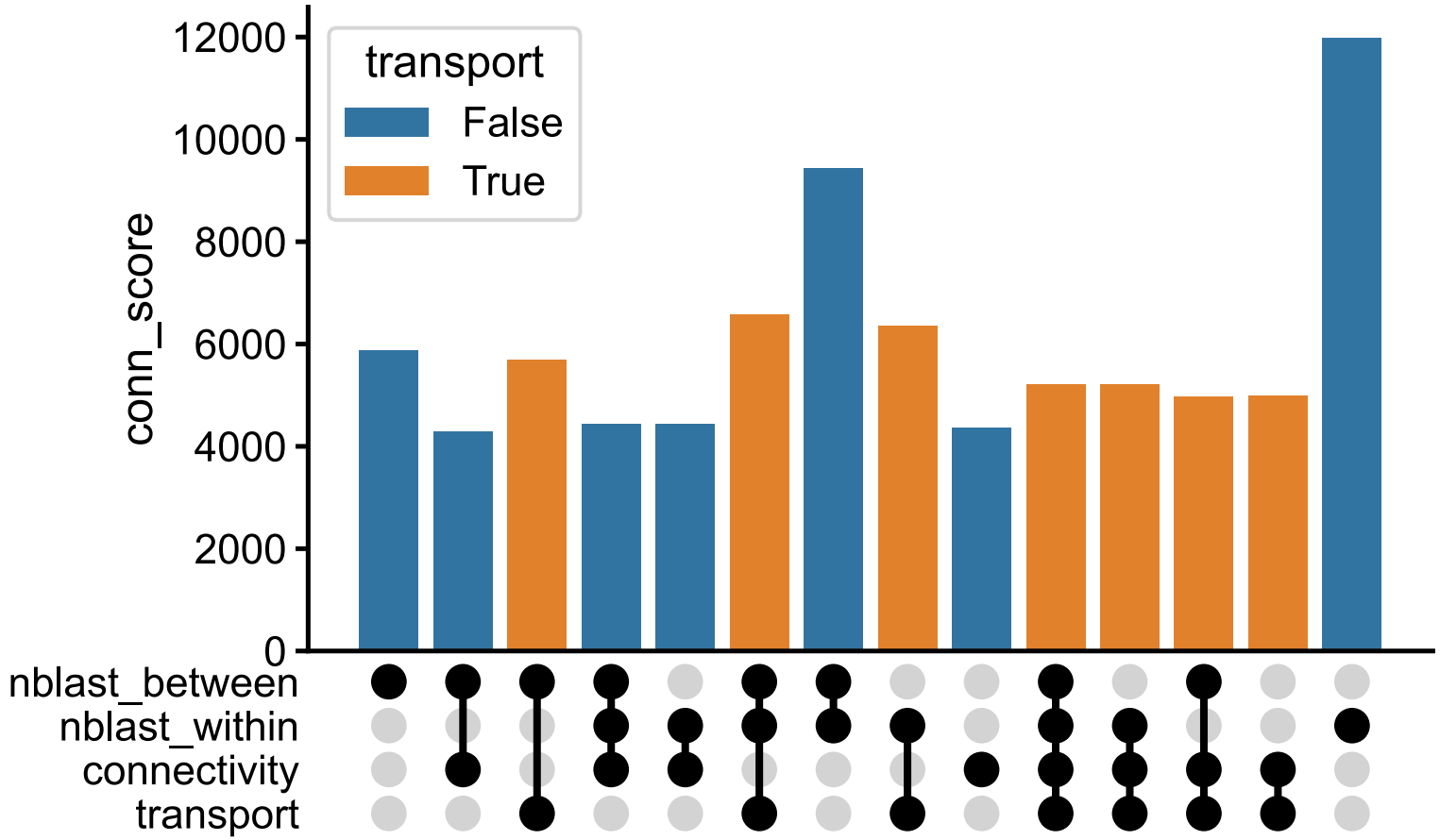
Can show this is equivalent to:

$$\min_P \|A - PBP^T\|_F$$

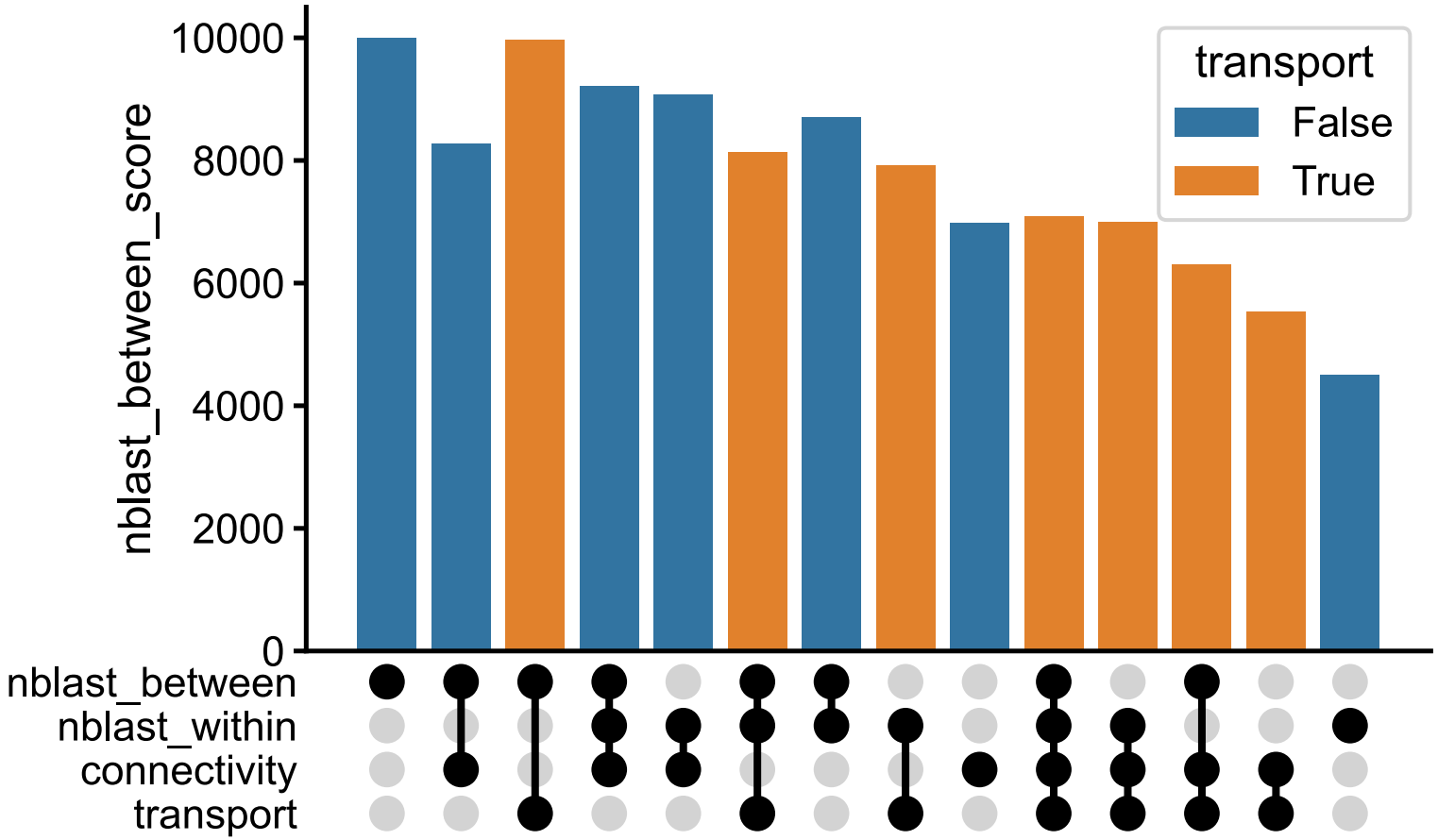
Accuracy on known labels



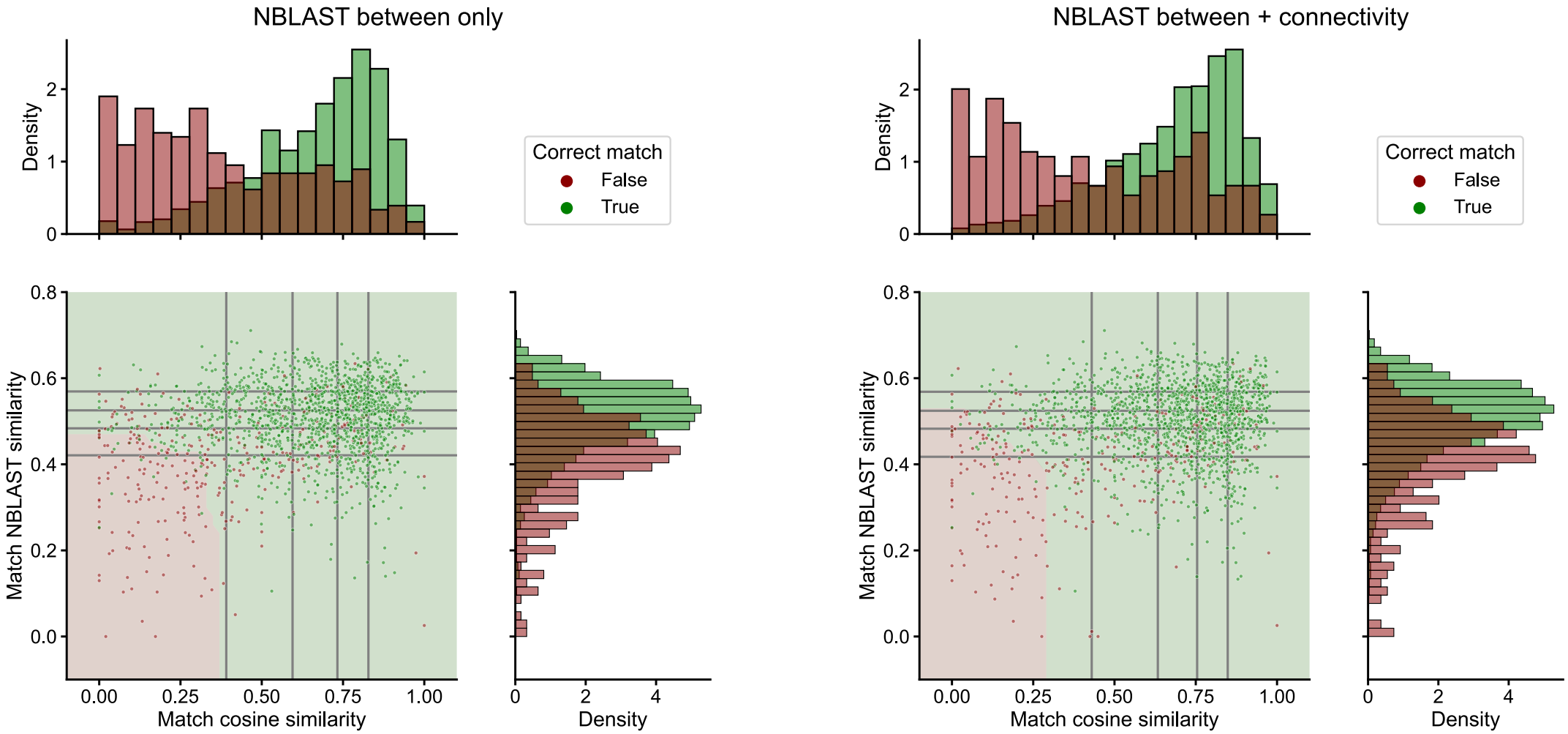
Connectivity score (low is good)



NBLAST score (high is good)

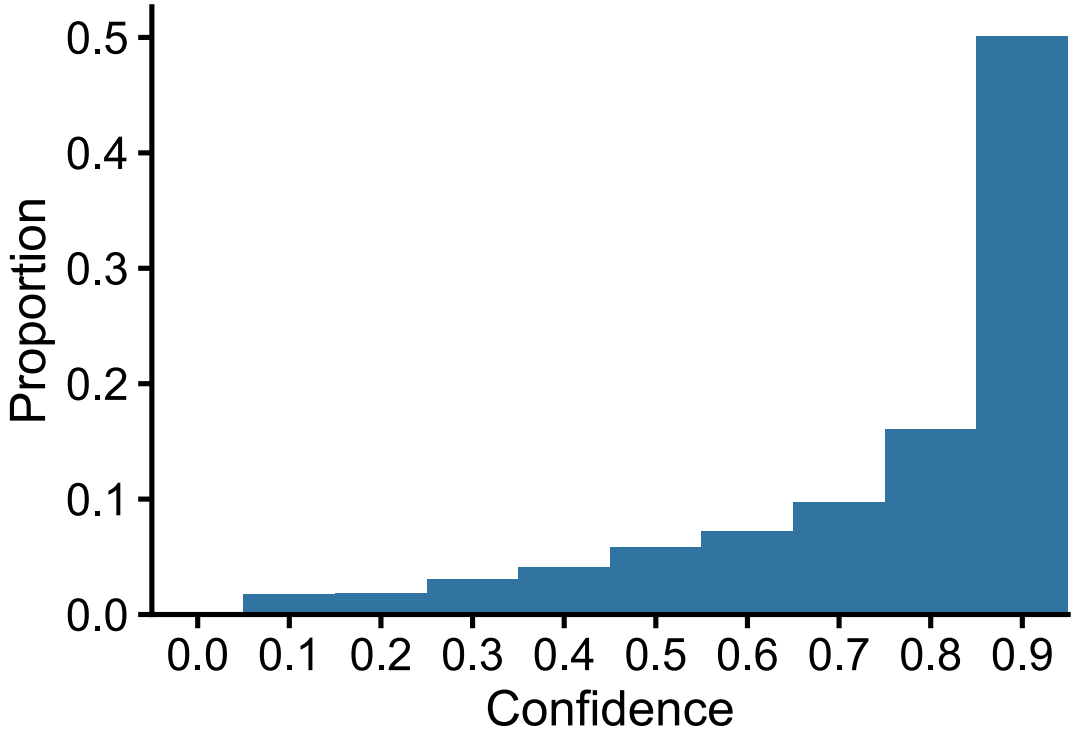


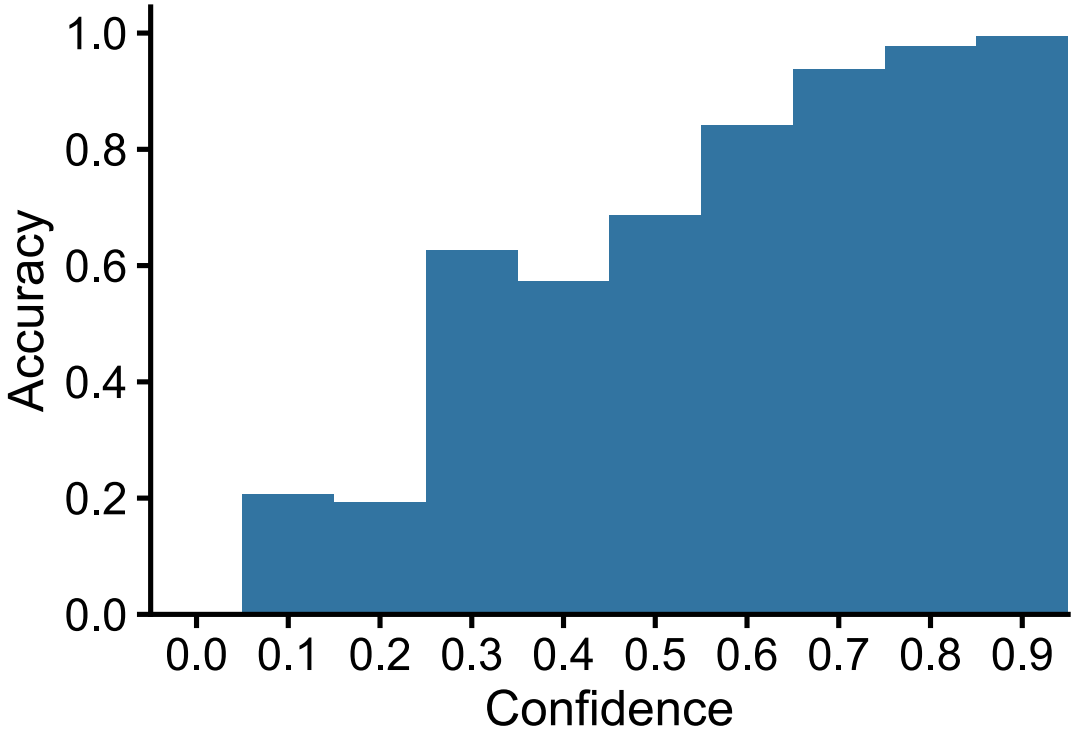
Can we tell when we're wrong?



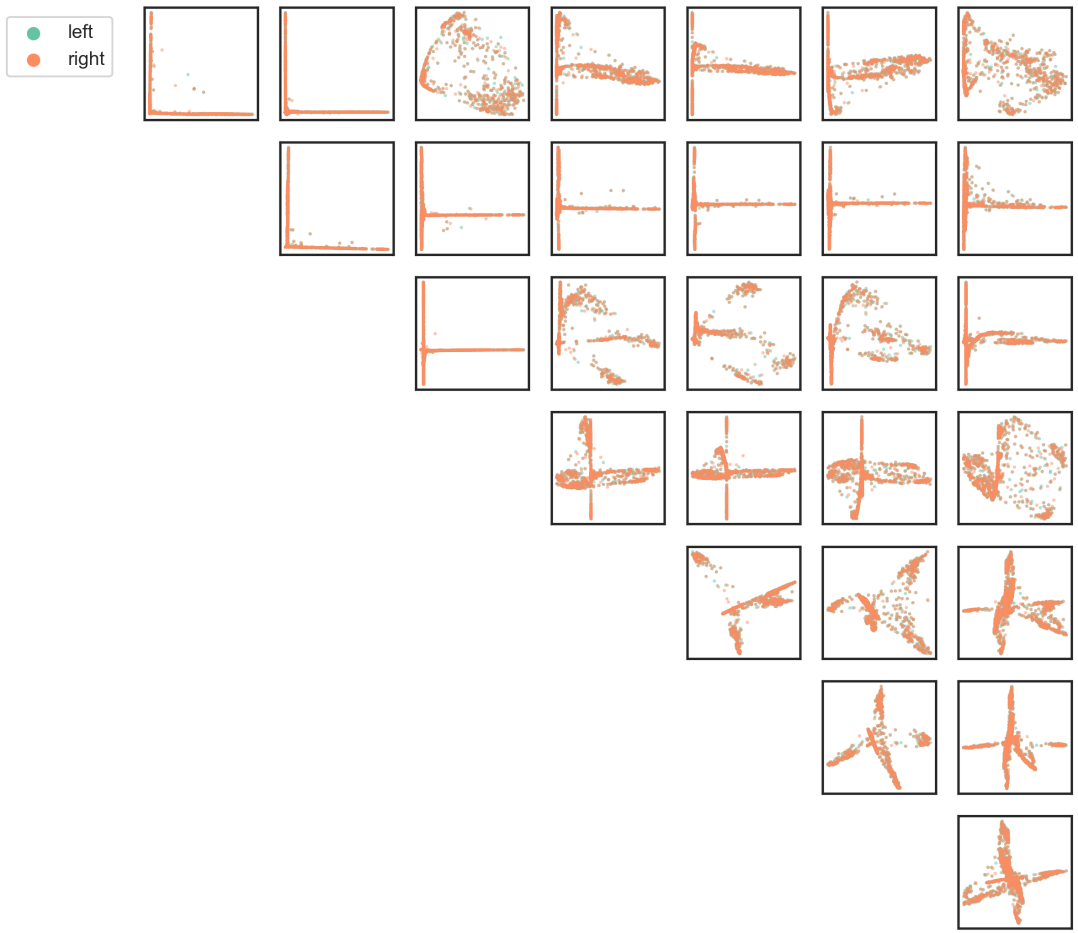
Can we tell when we're wrong?

For optimal transport on NBLAST, and grouping by label on the left side:

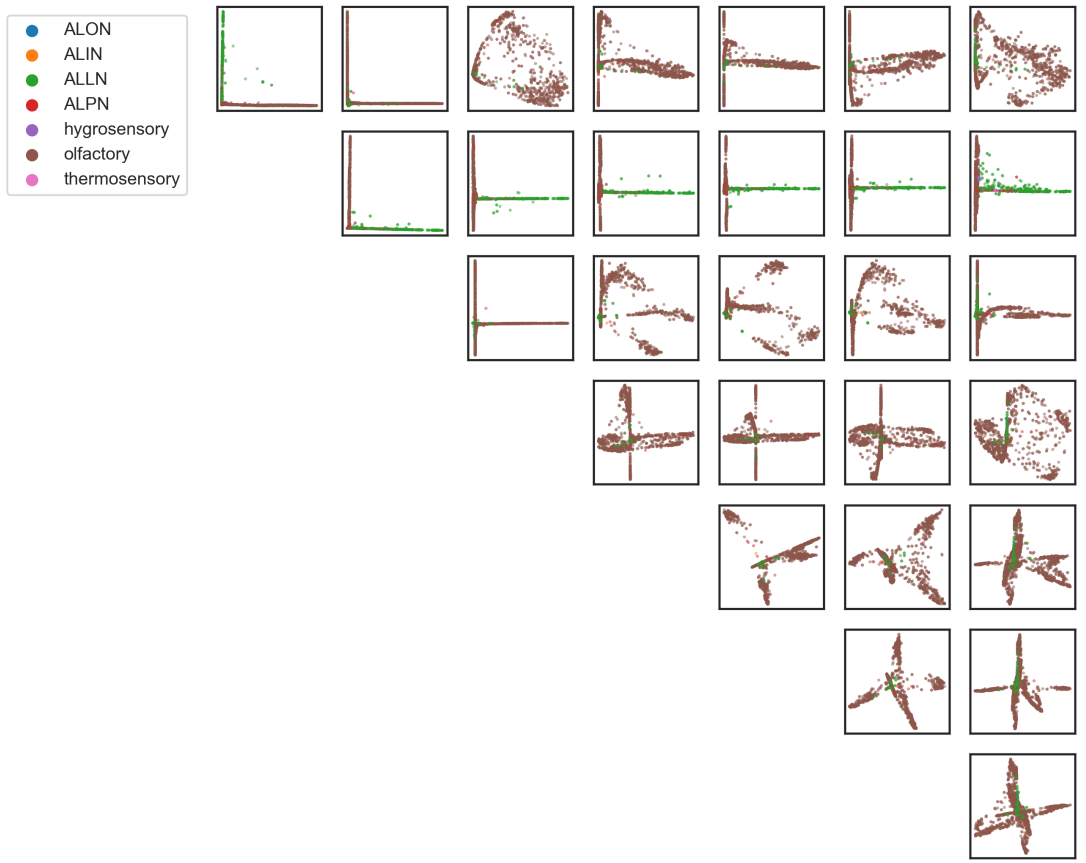


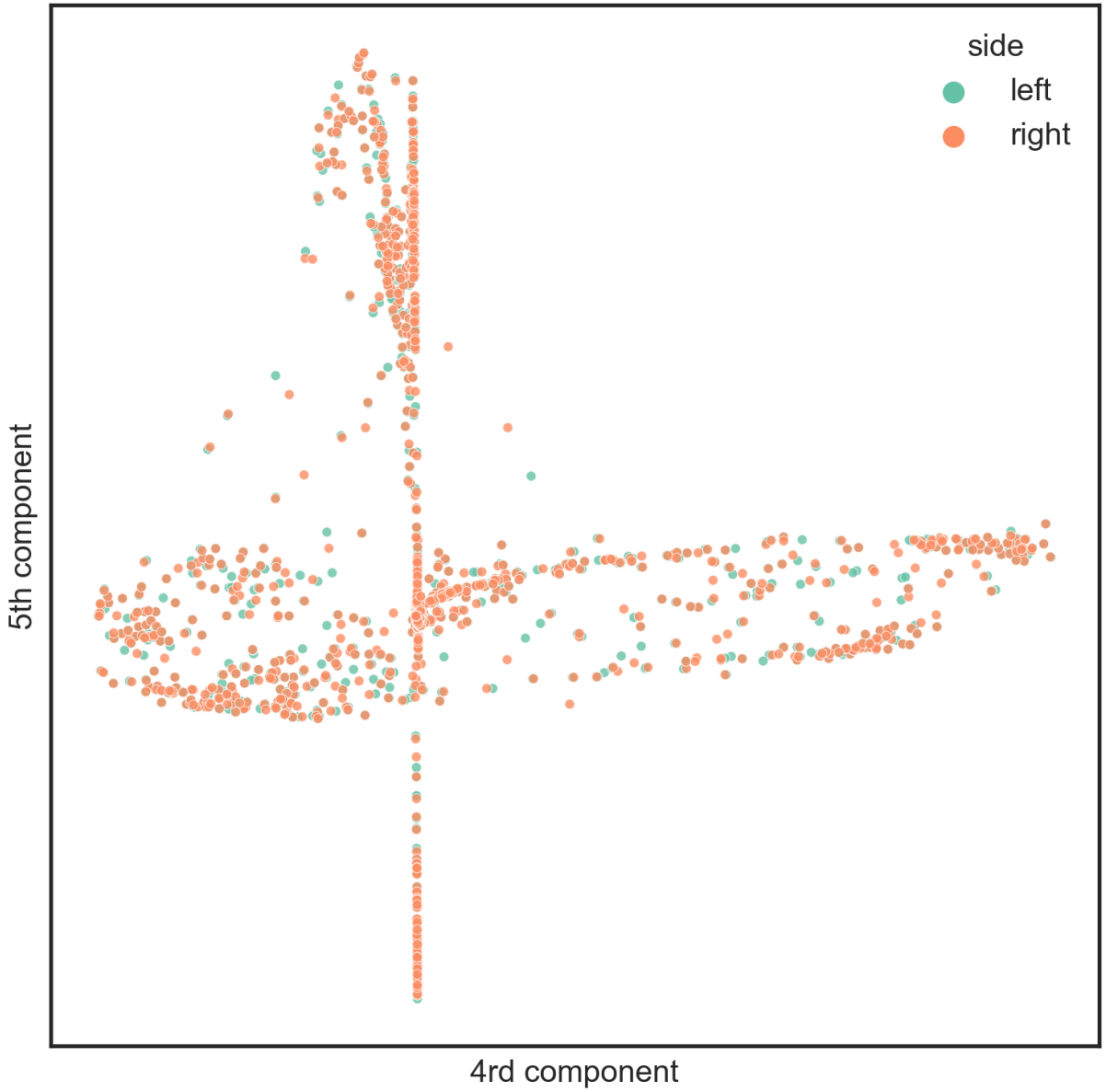


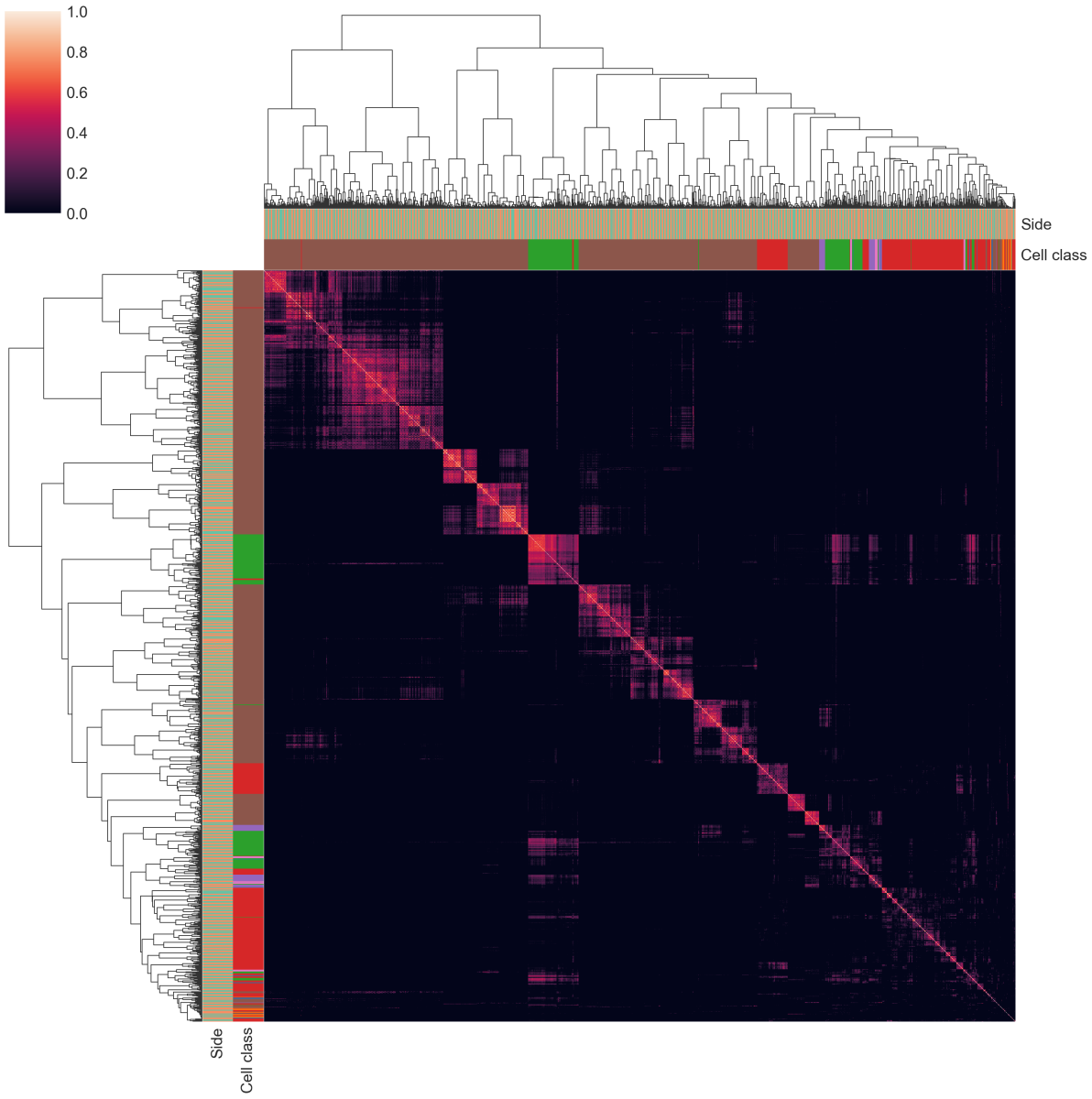
EMD


left
right

EMD


ALON
ALIN
ALLN
ALPN
hygrosensory
olfactory
thermosensory


side
left
right
5th component
4rd component


1.0
0.8
0.6
0.4
0.2
0.0
Side
Cell class
Side
Cell class

TODOs

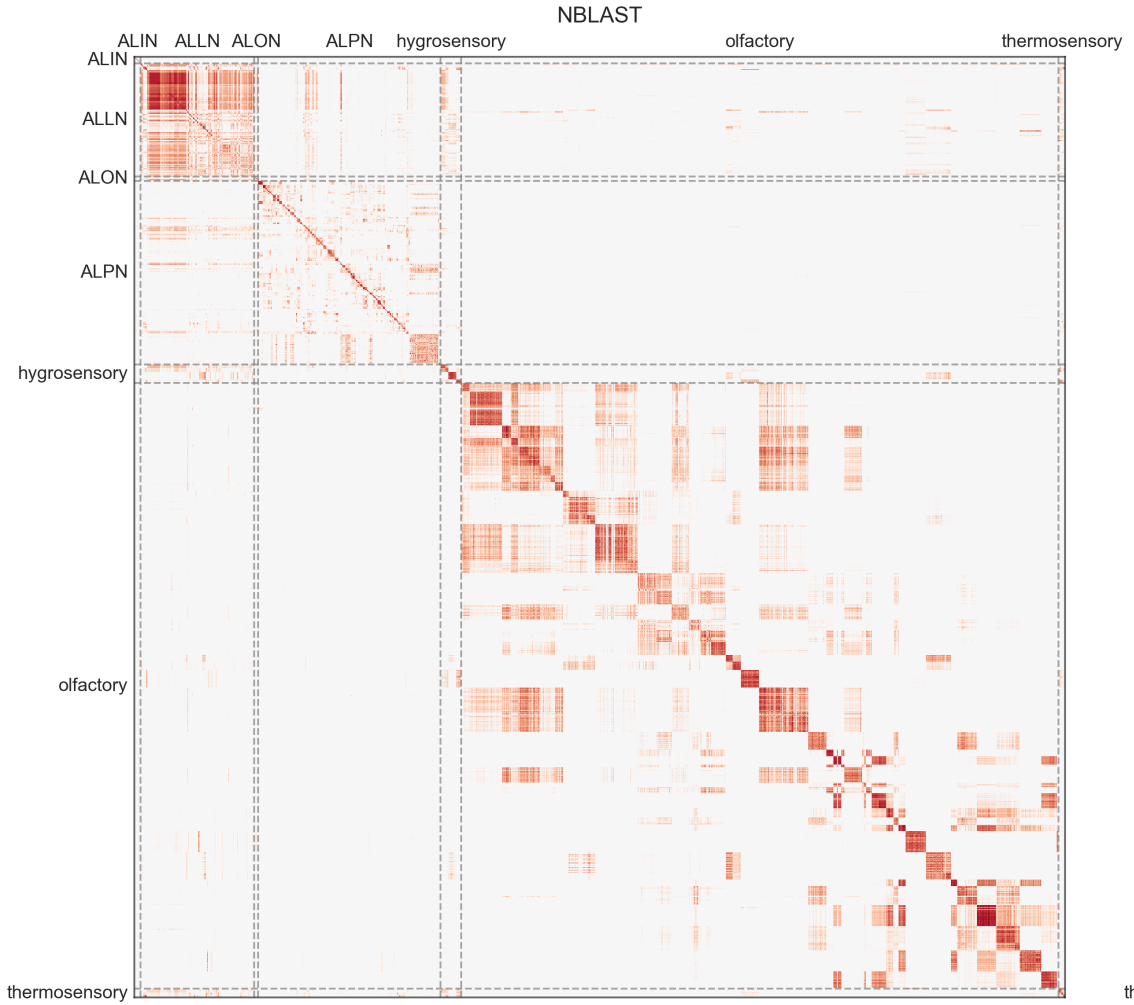
- Generalizing to more than 2 datasets at a time
- Scaling experiments
  - Optimal transport runs on central brain in ~minutes (on laptop)
  - Need to see whether graph matching can scale to that size
- Seeds/soft seeds?
  - Using pre-known matchings in the optimization

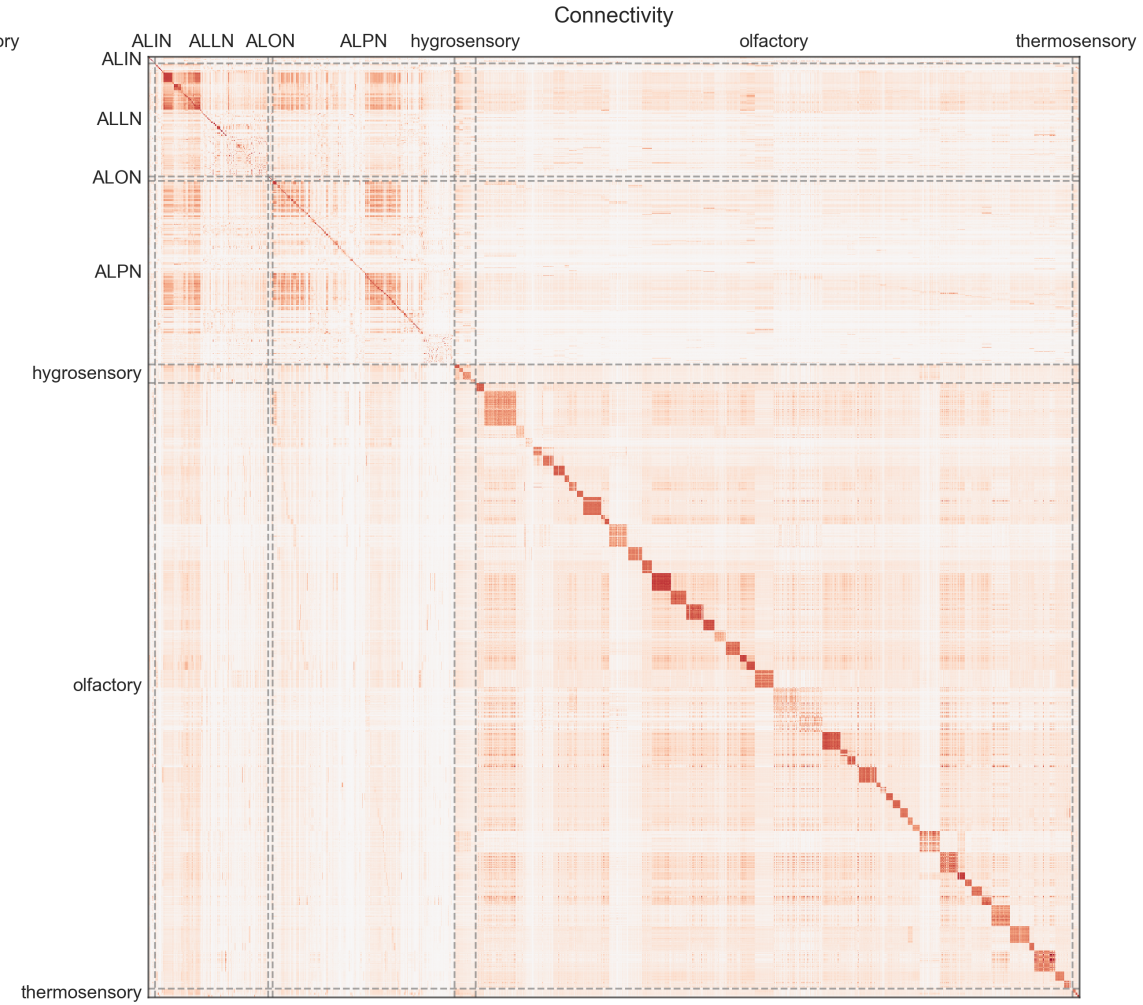
Appendix

Using this rough assignment to induce a matching for connectivity

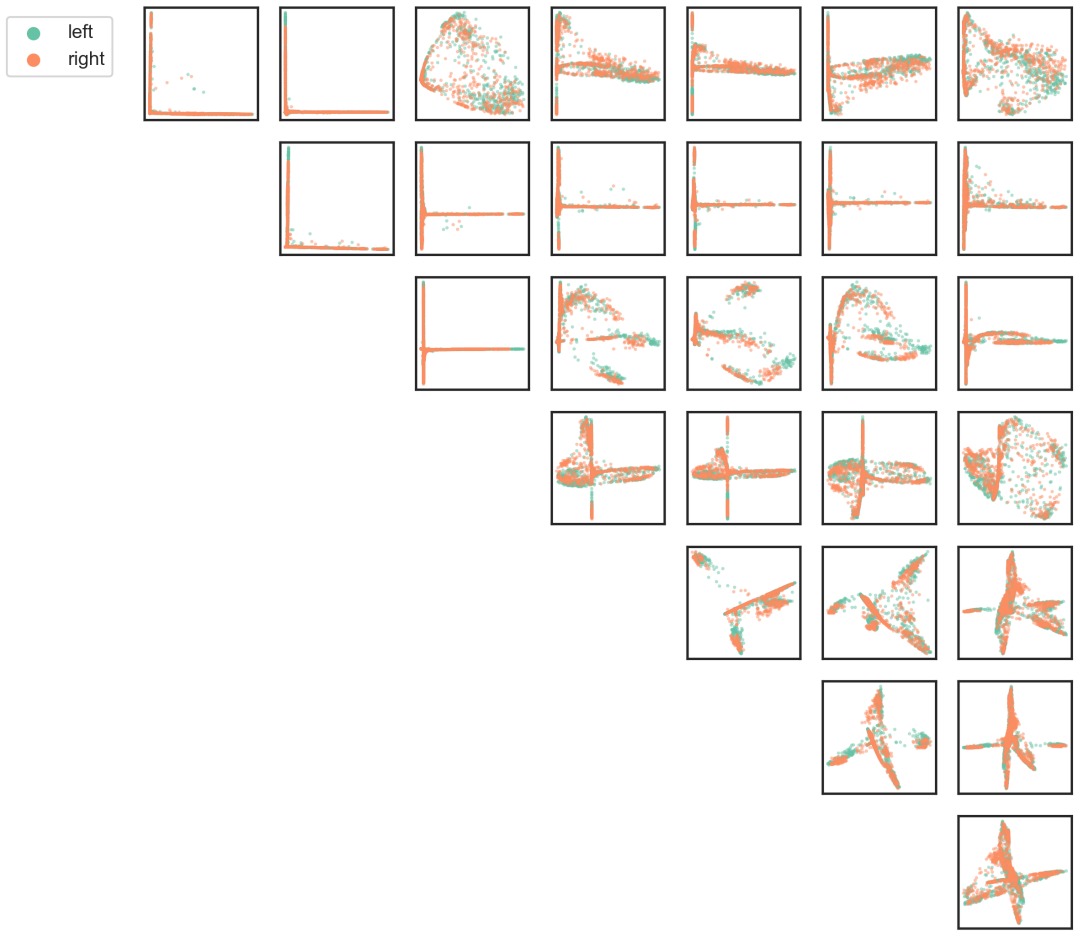
$$A^T F B$$

where F is the transportation solution we found above, which roughly maps neurons together based on their NBLAST similarity

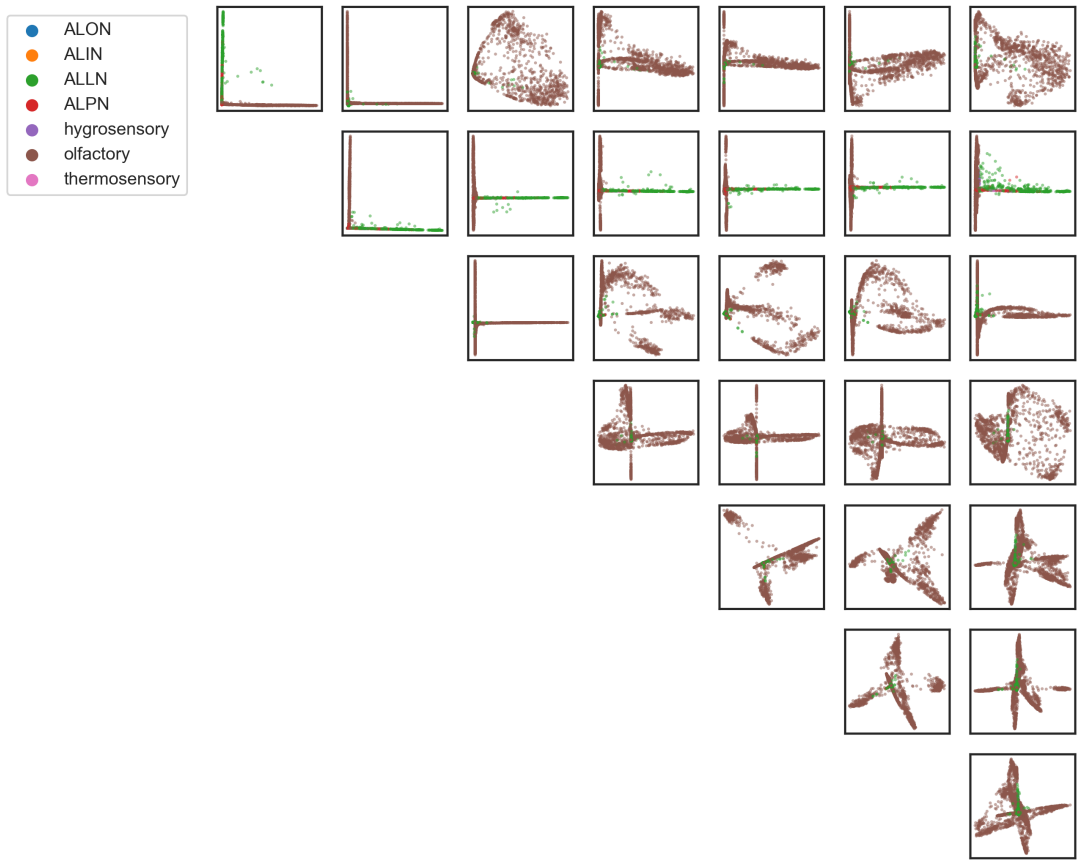

NBLAST
ALIN
ALLN
ALON
ALPN
hygrosensory
olfactory
thermosensory

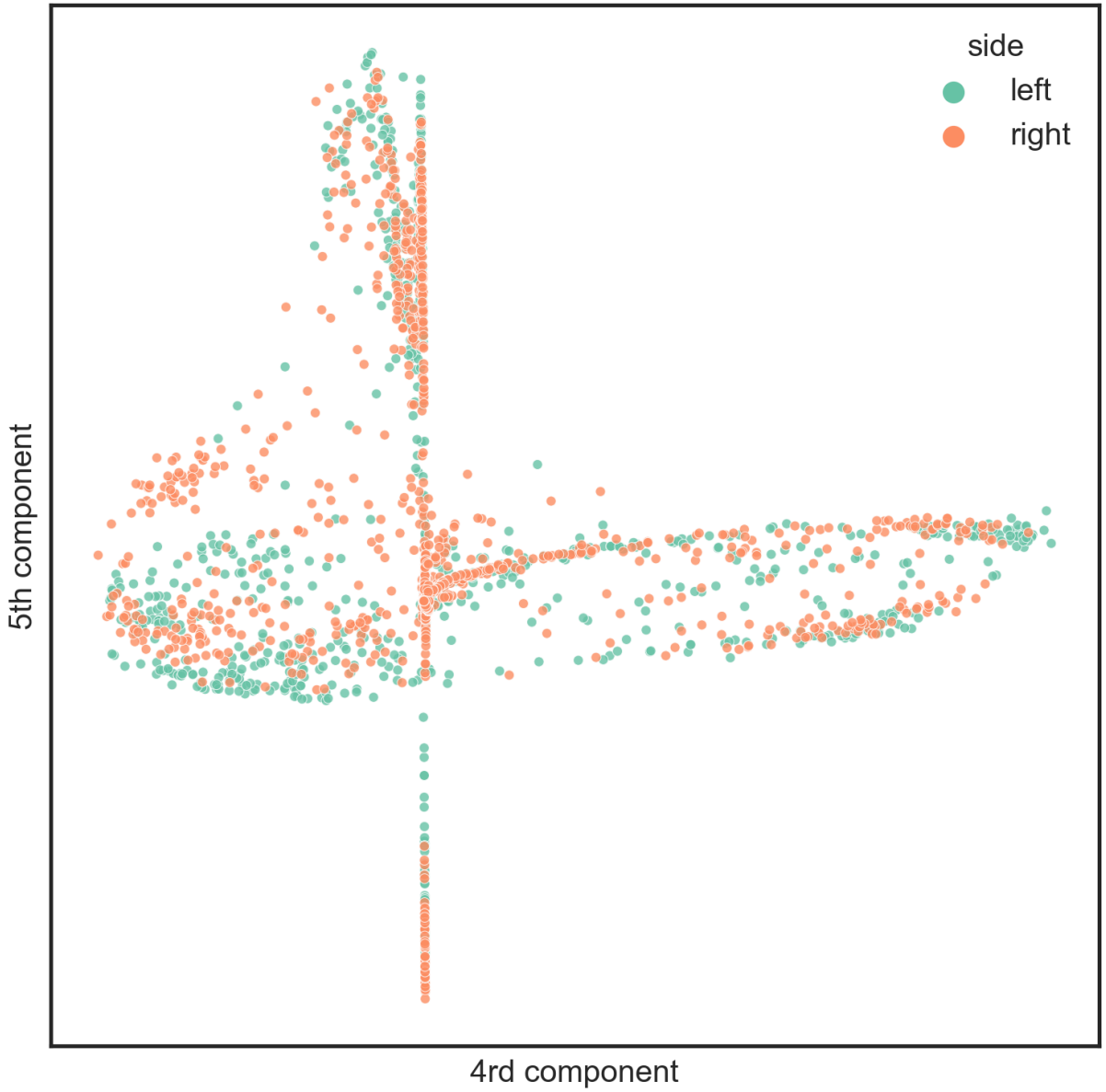

Connectivity
ALIN
ALLN
ALON
ALPN
hygrosensory
olfactory
thermosensory

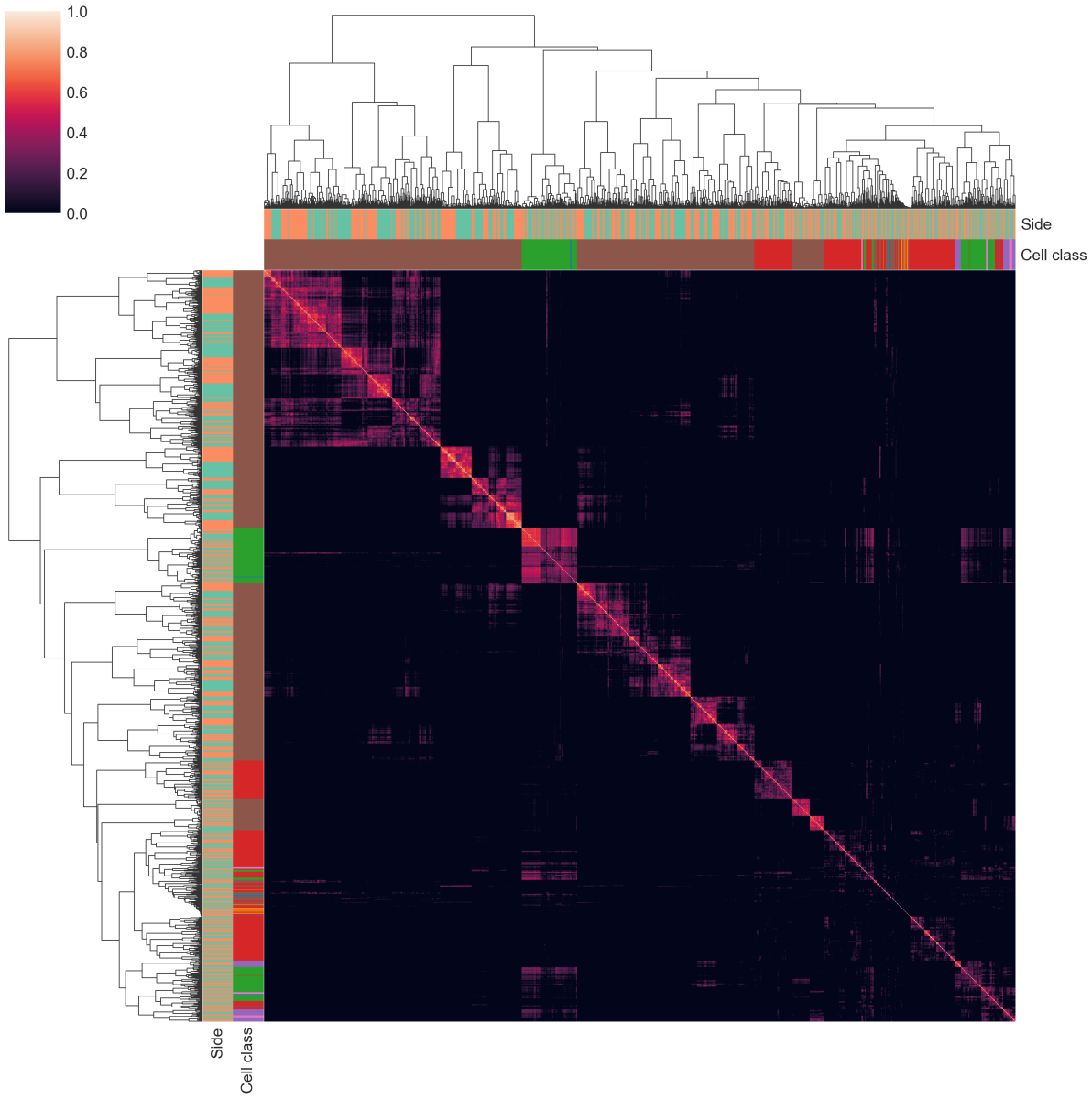
Raw


left
right

Raw


ALON
ALIN
ALLN
ALPN
hygrosensory
olfactory
thermosensory


side
left
right
5th component
4rd component


1.0
0.8
0.6
0.4
0.2
0.0
Side
Cell class
Side
Cell class