DIFFPASS – DIFFERENTIABLE AND SCALABLE PAIRING OF BIOLOGICAL SEQUENCES USING SOFT SCORES

Umberto Lupo[1, 2]
umberto.lupo@epfl.ch

Damiano Sgarbossa[1, 2]
damiano.sgarbossa@epfl.ch

Martina Milighetti[3, 4]
martina.milighetti.19@ucl.ac.uk

Anne-Florence Bitbol[1, 2]
anne-florence.bitbol@epfl.ch

[1]School of Life Sciences, Institute of Bioengineering, EPFL, Lausanne, Switzerland
[2]SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland
[3]Division of Infection and Immunity, University College London, London, United Kingdom
[4]Cancer Institute, University College London, London, United Kingdom

ABSTRACT

Identifying interacting sequences from two sets of potential partners has important applications in computational biology. Several methods have been developed to address this problem, applying different approximate optimization methods to different scores. We introduce DiffPaSS, a framework for flexible, fast, scalable, and hyperparameter-free optimization for pairing interacting biological sequences, which can be applied to a wide variety of scores. DiffPaSS consistently finds strong score optima, outperforming existing algorithms for optimizing the same scores.

1 INTRODUCTION

Identifying which polypeptide molecules interact together, using their sequence data alone, is an important and combinatorially difficult task. Mapping the network of protein-protein interactions, and predicting the three-dimensional structures of individual protein complexes, often requires determining which sequences are functional interaction partners among the paralogous proteins of two families. This problem can be formulated as looking for a permutation of the sequences one family with respect to those of the other. Information-theoretic, coevolution-based, and sequence-based scores have been proposed to tackle this problem; namely, score-maximising permutations are expected to encode several correct interactions (Bitbol et al., 2016; Bitbol, 2018; Gandarilla-Perez et al., 2023). More recently, sophisticated scores coming from language models have also been proposed, see Chen et al. (2023); Lupo et al. (2023).

We present DiffPaSS ("**Diff**erentiable **Pa**iring using **S**oft **S**cores"), a family of flexible, fast, scalable, and hyperparameter-free algorithms for pairing interacting sequences by optimizing smooth extensions of these scores to "soft" permutations of the input sequences, using gradient methods. Strong optima are reached thanks to a novel bootstrap technique, motivated by heuristic insights into this smooth optimization process. When using inter-chain mutual information between two multiple sequence alignments (MSAs) as the score to be maximised, DiffPaSS outperforms existing coevolution- and phylogeny-based pairing methods on difficult benchmarks extracted from ubiquitous interacting prokaryotic systems. Thanks to its scalability, DiffPaSS can easily produce paired alignments that can be used by AlphaFold-Multimer (AFM, (Evans et al., 2021)) to predict the three-dimensional structure of protein complexes. We show promising results in this direction, on a selection of eukaryotic complexes. Finally, we show that DiffPaSS can also be used when reliable MSAs are not available. Namely, using data for T-cell receptor (TCR) CDR3α and CDR3β loops, we use DiffPaSS to perform graph alignment based on pairwise Levenshtein distances, outperforming existing simulated annealing techniques. A `PyTorch` implementation and installable Python package are available at https://github.com/Bitbol-Lab/DiffPaSS.

2 METHODS

2.1 PRELIMINARIES

Let $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{B}}$ be ordered collections of amino-acid sequences that are partitioned into K groups, each of size N_k where $k = 1, \ldots, K$. In the important special case where $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{B}}$ are collections of proteins from two interacting protein families, the "groups" will be species, with species k assumed to contain N_k paralogous proteins in both families.

Let $\mathscr{S}$ be a score function of the two ordered collections. We would like to find a permutation π of the entries in $\mathcal{M}_{\mathrm{A}}$ which maximises $\mathscr{S}(\pi(\mathcal{M}_{\mathrm{A}}), \mathcal{M}_{\mathrm{B}})$, under the constraint that π does not send sequences from any one group into a different group. A priori, there are $\prod_{k=1}^{K} N_k!$ permutations satisfying this constraint. Note that we can equivalently describe this as the problem of finding $\mathscr{S}$-maximising one-to-one matchings between the sequences in $\mathcal{M}_{\mathrm{A}}$ and those in $\mathcal{M}_{\mathrm{B}}$. Since $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{B}}$ will remain fixed, by abusing notation we denote $\mathscr{S}(\pi(\mathcal{M}_{\mathrm{A}}), \mathcal{M}_{\mathrm{B}})$ simply by $\mathscr{S}(\pi)$.

Let $N > 0$, and denote the set of permutation matrices of N elements by $\mathcal{P}_N$. For any integer $\ell > 0$ and arbitrary $N \times N$ square matrix X, define the *ℓ-truncated Sinkhorn operator*

$$S^{\ell}(X) = (\mathcal{C} \circ \mathcal{R})^{\ell}(\exp(X)) \tag{1}$$

consisting of first applying the componentwise exponential function $\exp$ to X, and then iteratively normalizing rows ($\mathcal{R}$) and columns ($\mathcal{C}$) ℓ times. It can be shown (Mena et al., 2018) that $S(X) := \lim_{\ell \to \infty} S^{\ell}(X)$ defines a smooth operator mapping to bistochastic matrices[1] and that, for almost all X,

$$\lim_{\tau \to 0^{+}} S(X/\tau) = M(X) := \arg\max_{P \in \mathcal{P}_N} [\mathrm{trace}(P^{\mathrm{T}} X)]. \tag{2}$$

The operator M defined in Eq. (2), which maps onto permutation matrices, can be computed using standard discrete algorithms for linear assignment problems (Kuhn, 1955). Eq. (2) implies that, using the "parameterization matrices" X, one can smoothly navigate the space $\mathcal{B}_N$ of all $N \times N$ bistochastic (resp. near-bistochastic) matrices using S (resp. S^{ℓ}), while keeping track of the "nearest" permutation matrices using M. In what follows, we will refer to (near-)bistochastic matrices as "soft permutations" and to true permutations as "hard permutations".

We may therefore hope to find optimal hard permutations for the original score $\mathscr{S}$ by optimizing a suitable smooth extension $\hat{\mathscr{S}}$ of $\mathscr{S}$ to soft permutations, since this can be done efficiently using gradient methods. In practice, there are typically infinitely many choices for such an extension, and whether this strategy can work with e.g. a few steps of gradient descent starting from a typical random initialization (e.g. Gaussian i.i.d. entries for X) depends strongly on these choices. For some choices of $\hat{\mathscr{S}}$, the optimal soft permutations may be too distant from hard permutations to approximate good solutions of the original problem. For other choices, every hard permutation may be a local minimum. In general, optimization of $\hat{\mathscr{S}}$ can be very sensitive to hyperparameters such as the "temperature" $\tau > 0$ in Eq. (2), the standard deviation of the entries of X at initialization, the optimizer learning rate, and the strength of regularization.

2.2 PAIRED MULTIPLE SEQUENCE ALIGNMENTS, MUTUAL-INFORMATION–BASED SCORES

In several applications, the ordered collections $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{B}}$ in Section 2.1 consist of full-length protein sequences or of protein domains, and are given in the form of MSAs, assumed here to have the same depth. Denoting the i-th column of $\mathcal{M}_{\mathrm{A},i}$ (and analogously for $\mathcal{M}_{\mathrm{B}}$), we define the *total inter-chain mutual information* score $\mathscr{S}_{\mathrm{MI}}(\mathcal{M}_{\mathrm{A}}, \mathcal{M}_{\mathrm{B}})$ as

$$\mathscr{S}_{\mathrm{MI}}(\mathcal{M}_{\mathrm{A}}, \mathcal{M}_{\mathrm{B}}) := \sum_{i,j} \mathrm{I}(\mathcal{M}_{\mathrm{A},i}; \mathcal{M}_{\mathrm{B},j}) = \sum_{i,j} [\mathrm{H}(\mathcal{M}_{\mathrm{A},i}) + \mathrm{H}(\mathcal{M}_{\mathrm{B},j}) - \mathrm{H}(\mathcal{M}_{\mathrm{A},i}, \mathcal{M}_{\mathrm{B},j})], \tag{3}$$

where $\mathrm{I}(\cdot;\cdot)$ denotes the mutual information between two random variables, and H denotes Shannon entropy. Note that, in Eq. (3), these information quantities are in fact replaced by their plug-in estimates. For a random variable Z, this means using $\mathrm{H}(Z) = -\sum_z f_z \log_2 f_z$, with f_z being the observed frequency of state z, which converges to its probability in the limit of infinite sample size.

[1] A bistochastic matrix is a matrix with non-negative entries whose all rows and columns sum to 1.

Since, for any i and permutation π, $\mathrm{H}(\mathcal{M}_{\mathrm{A},i}) = \mathrm{H}(\pi(\mathcal{M}_{\mathrm{A}})_i)$ (and similarly for $\mathcal{M}_{\mathrm{B}}$ and any j), maximising $\mathscr{S}_{\mathrm{MI}}(\pi)$ is equivalent to minimizing the *total inter-chain two-body entropy loss*

$$\mathscr{L}_{\mathrm{2BE}}(\pi) := \sum_{i,j} \mathrm{H}(\pi(\mathcal{M}_{\mathrm{A}})_i, \mathcal{M}_{\mathrm{B},j}). \tag{4}$$

A simple smooth extension $\hat{\mathscr{L}}_{\mathrm{2BE}}$ of $\mathscr{L}_{\mathrm{2BE}}$ to soft permutations can be defined as follows. First, we represent the amino acids/gap symbols in the MSAs as one-hot vectors. Namely, let $\boldsymbol{m}_{\mathrm{A},i,n}$ denote the one-hot vector corresponding to row (i.e. sequence) n and column (i.e. site) i in $\mathcal{M}_{\mathrm{A}}$ – and similarly for $\mathcal{M}_{\mathrm{B}}$. The matrix of observed counts of all joint amino-acid states at the column pair (i, j) can then be computed as $\sum_n \boldsymbol{m}_{\mathrm{A},i,n} \otimes \boldsymbol{m}_{\mathrm{B},j,n}$, where $\otimes$ denotes vector outer product. This expression is actually well-defined and smooth for pairs of arbitrary vectors. Provided these vectors have non-negative entries, we can thus define a smooth extension $\hat{\mathrm{H}}(\cdot, \cdot)$ of the two-body entropy $\mathrm{H}(\cdot, \cdot)$. For a soft permutation $\hat{\pi}$ represented as a matrix $\hat{P}$, we can compute a "soft MSA" by matrix multiplication as $\hat{\pi}(\mathcal{M}_{\mathrm{A}}) = \hat{P}\boldsymbol{M}_{\mathrm{A}}$, where $\boldsymbol{M}_{\mathrm{A}}$ is the representation of $\mathcal{M}_{\mathrm{A}}$ as a tensor with an additional "one-hot" dimension. Finally, we can extend $\mathscr{L}_{\mathrm{2BE}}$ as

$$\hat{\mathscr{L}}_{\mathrm{2BE}}(\hat{\pi}) := \sum_{i,j} \hat{\mathrm{H}}((\hat{P}\boldsymbol{M}_{\mathrm{A}})_i, \boldsymbol{M}_{\mathrm{B},j}). \tag{5}$$

2.3 Initialization and bootstrapped optimization

We experimentally found that solving $X^* = \arg\min_X \hat{\mathscr{L}}_{\mathrm{2BE}}(S^\ell(X/\tau))$, for some choice of ℓ and τ, using several steps of gradient descent, generally yields sub-optimal hard permutations $M(X^*)$ for the original loss $\mathscr{L}_{\mathrm{2BE}}$, see Eq. (2). Informally, this is because hard permutations are local minima for $\hat{\mathscr{L}}$, a situation reminiscent of how the entropy of a single Bernoulli random variable with parameter p has minima at the "sharp" cases $p = 0$ and $p = 1$. Nevertheless, we found that outcomes improved when entries of X are initialized with zero standard deviation. Indeed, as shown in Figs. A1 and A2 for the benchmark HK-RR prokaryotic dataset described in Appendix A2, the *first* gradient step alone, when $X \equiv 0$ at initialization, is often competitive with a full-blown discrete algorithm for approximate maximization of $\mathscr{S}_{\mathrm{MI}}$, called MI-IPA (Bitbol, 2018). Finally, as previously observed using several methods (Bitbol et al., 2016; Bitbol, 2018; Gandarilla-Perez et al., 2023; Lupo et al., 2023), pairing performance can be expected to increase if correct pairings are used as fixed context, biasing the computation of the two-body entropies. Together, these considerations led us to DiffPaSS, our proposed bootstrapped approach to differentiable pairing. Given a loss $\mathscr{L}$ and its smooth extension $\hat{\mathscr{L}}$, two ordered collections $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{B}}$ containing D sequences, and (optionally) a pre-existing set $\mathcal{F}_{\mathrm{AB}} = \{(a_i, b_i)\}_{i=1}^{D_{\mathrm{fix}}}$ containing D_{fix} matched pairs of sequences, we proceed as follows.[2] Initialize $\mathcal{F}' = \mathcal{F}_{\mathrm{AB}}$; then, for every $n = 1, \ldots, D - D_{\mathrm{fix}}$:

1. define P' as the $(D_{\mathrm{fix}} + n - 1) \times (D_{\mathrm{fix}} + n - 1)$ permutation matrix corresponding to the matchings in $\mathcal{F}'$;
2. initialize two zero $D \times D$ matrices P and $\hat{P}$, and copy P' into the row-column pairs belonging to $\mathcal{F}'$ in both cases;
3. initialize a $(D - D_{\mathrm{fix}} - n + 1) \times (D - D_{\mathrm{fix}} - n + 1)$ parameterization matrix $X \equiv 0$, to be used for sequences not involved in pairs in $\mathcal{F}'$;
4. update $X \leftarrow -\nabla(\hat{\mathscr{L}} \circ \tilde{S}^\ell)(X = 0)$, where $\tilde{S}^\ell(X)$ denotes copying $S^\ell(X)$ into the submatrix of $\hat{P}$ obtained by removing rows and columns involved in pairs in $\mathcal{F}'$;
5. compute the hard permutation $\pi = \tilde{M}(X)$, where $\tilde{M}(X)$ denotes copying $M(X)$ into the submatrix of P obtained by removing rows and columns involved in pairs in $\mathcal{F}'$;
6. pick n pairs of sequences matched by π, but not in $\mathcal{F}_{\mathrm{AB}}$, uniformly at random;
7. update $\mathcal{F}' \leftarrow \mathcal{F}_{\mathrm{AB}} \cup \{\text{pairs selected in step 6}\}$.

[2] An animation illustrating this algorithm in the special case $D_{\mathrm{fix}} = 0$ is available at the following URL: https://www.youtube.com/watch?v=G2rV4ldgTIY.

For every $n = 1, \ldots, D - D_{\text{fix}}$, we record the loss $\mathscr{L}(\pi)$ at step 5, and the final output of DiffPaSS is the hard permutation π^* corresponding to the lowest recorded loss.

Note that the gradient of $S^\ell(X/\tau)$, evaluated at $X = 0$, only changes by a global scale factor if τ is changed, and the matching operator M is scale-invariant. Hence, we can set $\tau = 1$ as the obtained hard permutations are independent of it. Similarly, all hard permutations obtained are independent of the choice of learning rate and regularization strength. Perhaps more surprisingly, for any $\ell > 1$, the gradient of S^ℓ evaluated at $X = 0$ is equal to corresponding gradient of $S^{\ell=1}$. We prove this in Appendix A1. Hence, we can set $\ell = 1$ throughout, leading to significant runtime gains.

Robust pairs and DiffPaSS-IPA. We noticed that, even when the starting set $\mathcal{F}_{\text{AB}}$ of fixed pairs is empty, some pairs are matched by all the $D - D_{\text{fix}}$ hard permutations π explored by DiffPaSS (see step 5 in Section 2.3). We call these *robust pairs*, and notice experimentally that they tend to have high precision, see Fig. A4. This suggests that they can be used as the starting set $\mathcal{F}_{\text{AB}}$ of fixed pairs in a further run of DiffPaSS. We call this procedure DiffPaSS-IPA ("Iterative Pairing Algorithm", following Bitbol et al. (2016); Bitbol (2018)). It can be iterated an arbitrary number N_{IPA} of times, enlarging the set of robust pairs after each iteration and stopping if no further robust pairs are found. The final output of DiffPaSS-IPA is the hard permutation with lowest observed loss across all IPA runs. We use $N_{\text{IPA}} = 3$ throughout.

2.4 Other scores

We have motivated the DiffPaSS bootstrap framework using the data structure of MSAs and the mutual information between MSA columns. However, DiffPaSS can in principle be applied to a variety of other scores, including scores based on sequence similarities, orthology, and phylogeny Izarzugaza et al. (2008); Bradde et al. (2010). In the case of coevolving families of interacting proteins for which alignments of good quality are available, we could not match the performance obtained using mutual information using these alternative scores alone. This is in agreement with previous findings (Gandarilla-Perez et al., 2023).

However, for some polypeptide sequences of interest, alignments are not always easy to construct or even meaningful. This is arguably the case for collections of sequence from hypervariable CDR3α and CDR3β loops in TCRs binding to a fixed epitope.[3] In these cases, one may instead opt for computing matrices of pairwise scores between unaligned sequences, and performing a graph alignment between the two collections of interest $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{B}}$, using these matrices as edge weights. We propose a simple variant of DiffPaSS for performing hyperparameter-free graph alignment (GA) starting from two weighted adjacency matrices W_{A} and W_{B} computed from $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{B}}$ (Petric Maretic et al., 2019; Maretic et al., 2022b;a; Gandarilla-Perez et al., 2023). We assume that the score $\mathscr{S}$ has the form $\mathscr{S}(\mathcal{M}_{\text{A}}, \mathcal{M}_{\text{B}}) = \mu(W_{\text{A}}, W_{\text{B}})$ for a differentiable function μ, so that $\mathscr{S}(\pi) = \mu(PW_{\text{A}}P^{\text{T}}, W_{\text{B}})$. Then, we simply perform a DiffPaSS bootstrap procedure as in Section 2.3, using $-\mu(\hat{P}W_{\text{A}}\hat{P}^{\text{T}}, W_{\text{B}})$, for soft permutations $\hat{P}$, as the loss.

3 Results and discussion

DiffPaSS significantly outperforms other coevolution or phylogeny methods based on simple MSA scores. We test DiffPaSS and DiffPaSS-IPA, using the total inter-chain mutual information as a score (see Section 2.2), on a dataset of cognate pairs of histidine kinases (HK) and response regulators (RR), paired using genome proximity and described in Appendix A2. Fig. 1 shows that they significantly outperform existing methods based on simple MSA scores, particularly the MI-IPA algorithm in Bitbol (2018) – which performs an approximate maximization of the same MI score – and the current in-class state of the art GA–MI-IPA (Gandarilla-Perez et al., 2023) – which improves on MI-IPA by combining it with a computationally costly simulated annealing GA algorithm, based on sequence similarity and used to produce "phylogenetically-robust pairs". With all methods, a full one-to-one within-species pairing is produced, and performance is measured as "precision-100", which is the fraction of correct pairs among all predicted pairs. Importantly, these results are obtained in all cases without giving any paired sequences (other than the self-identified robust pairs in the case of GA–MI-IPA and DiffPaSS-IPA) as inputs.

[3] Here, the "groups" are not species but individuals, e.g. human patients.

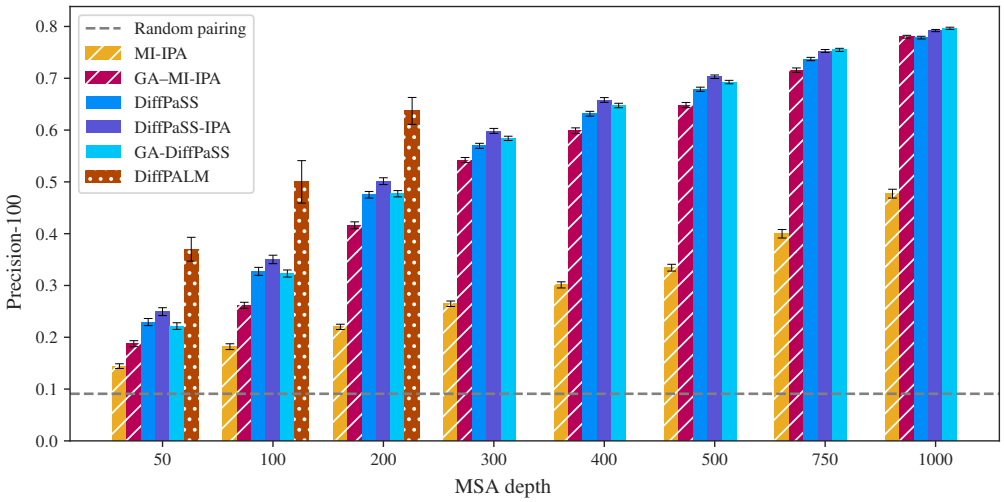

Figure 1: **Performance of pairing** on the HK-RR dataset described in Appendix A2. For GA–MI-IPA, we used the same settings as Gandarilla-Perez et al. (2023). Results for DiffPALM are taken from Lupo et al. (2023).

While DiffPaSS does *not* outperform the MSA-Transformer–based model DiffPALM (Rao et al., 2021; Lupo et al., 2023) on this dataset, it is several orders of magnitude faster (see below), and easily scales to much deeper MSA depths for which DiffPALM cannot be run due to memory limitations. Fig. A3 shows that, on average, DiffPaSS(-IPA) final pairings have almost indistinguishable scores from the ground truth pairings. Hence, motivated by the question of whether additional signal can be extracted from simple phylogeny measures, we test a variant, called GA-DiffPaSS, in which the same robust pairs obtained by the GA step in the GA–MI-IPA algorithm are used as fixed context pairs for a run of DiffPaSS. While we obtain some improvements for deeper MSAs, relative to a single DiffPaSS run, these can be matched or surpassed by DiffPaSS-IPA, which has no direct access to phylogenetic signal. Fig. A7 shows that DiffPaSS-IPA typically identifies fewer correct pairs as robust than GA–MI-IPA, but its robust pairs are correct more often.

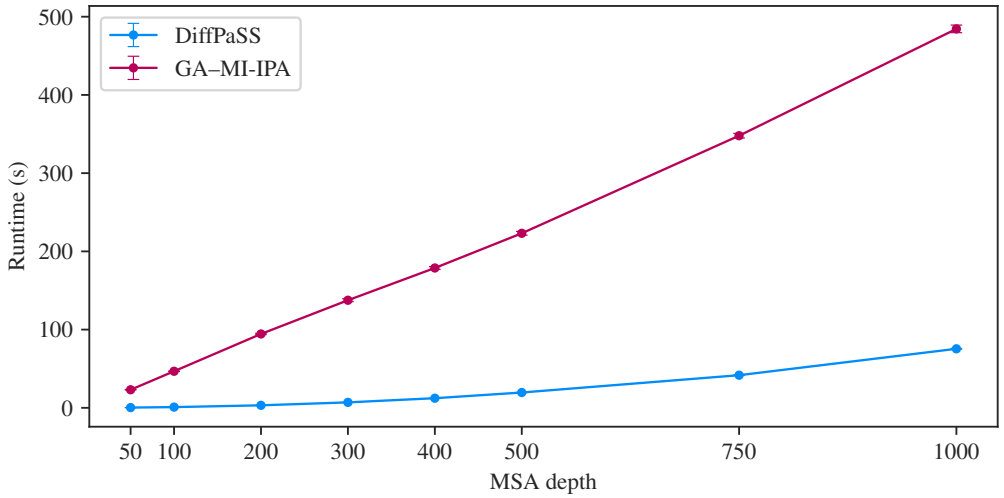

Figure 2: **Runtime comparison** between DiffPaSS and GA–MI-IPA (Gandarilla-Perez et al., 2023), on the HK-RR dataset described in Appendix A2. DiffPaSS was implemented in `PyTorch` v2.2.1, and run on an NVIDIA® GeForce RTX™ 3090 GPU with 24 GB memory. We used the original implementation of GA–MI-IPA in the Julia programming language (Julia v1.10.0). All 20 GA replicates were run in parallel on separate Intel® Xeon® Platinum 8360Y CPUs running at 2.4 GHz.

DiffPaSS can easily be run on a modern GPU, while GA–MI-IPA is a CPU-only algorithm. Fig. 2 demonstrates that DiffPaSS is considerably faster than GA–MI-IPA across all the MSA depths we analyzed. DiffPALM (Lupo et al., 2023) – not shown in Fig. 2 – has much longer runtimes than both methods, taking e.g. over three orders of magnitudes longer than DiffPaSS to pair HK-RR MSAs of depth ~ 50.

Using DiffPaSS for eukaryotic complex structure prediction by AFM. We ask the question whether DiffPaSS can improve complex structure prediction by AFM (Evans et al., 2021) in eukaryotic complexes, for which pairing correct interaction partners is notoriously difficult. To this end, we consider the same 15 eukaryotic structures as in Lupo et al. (2023), where improvements over the default AFM pairing methods were reported by pairing using the MSA-Transformer–based method DiffPALM (Rao et al., 2021). More information on these structures and on the AFM setup can be found in Appendices A2 and A3. Fig. 3 compares the performance of AFM on these complexes, using three different pairing methods (default AFM, DiffPALM, and DiffPaSS based on mutual information) on the same initial unpaired MSAs. We use the DockQ score – a widely used measure of quality for protein-protein docking (Basu & Wallner, 2016) – as a performance metric for this task. Multimer confidence scores for these predictions are shown in Fig. A5. These preliminary results show that DiffPaSS can improve complex structure prediction in some cases. We leave a larger-scale exploration on more eukaryotic complexes – made possible by the scalability of our method – to a future version of this work.

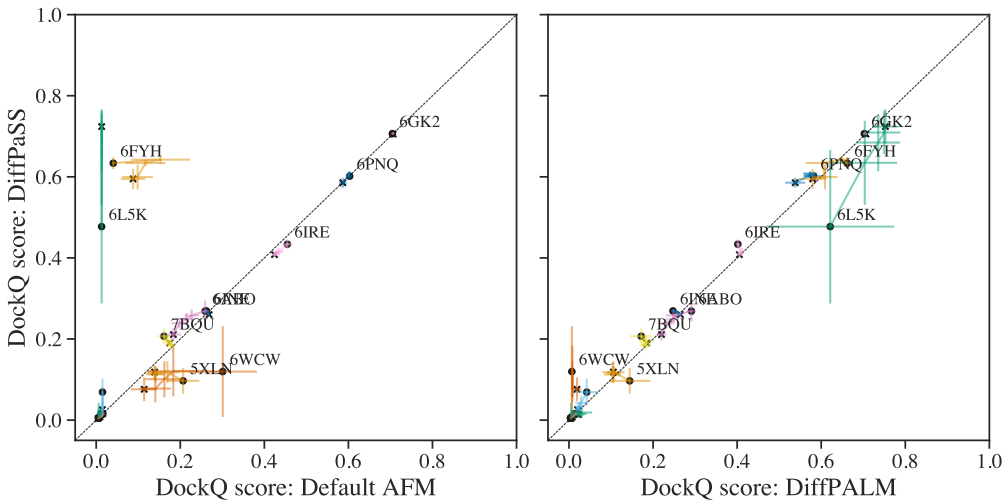

Figure 3: **Performance of structure prediction by AFM using different MSA pairing methods.** "Trajectory visualizations" and results for Default AFM and DiffPALM-based pairing are as in Lupo et al. (2023, Fig. S5).

Graph alignment for TCR data using DiffPaSS. Our dataset of paired CDR3α-CDR3β loop sequences from TCRs is described in Appendix A2. Using DiffPaSS as described in Section 2.4, we maximise the same graph alignment score as the stochastic algorithm based on simulated annealing in Gandarilla-Perez et al. (2023), using (exponentially weighted) Levenshtein distances between pairs of CDR3 sequences to define edge weights. Fig. A6 shows that, on average, DiffPaSS obtains better scores than simulated annealing. Fig. A7 shows that the fraction of pairs correctly predicted by DiffPaSS is higher than the chance expectation in most cases. These results show promise for the use of DiffPaSS to optimize scores that cannot be constructed from MSAs.

ACKNOWLEDGMENTS

U. L., D. S., and A.-F. B. acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 851173, to A.-F. B.). M. M. acknowledges funding by Cancer Research UK through a doctoral studentship.

REFERENCES

M. Barakat, P. Ortet, C. Jourlin-Castelli, M. Ansaldi, V. Mejean, and D. E. Whitworth. P2CS: a two-component system resource for prokaryotic signal transduction research. *BMC Genomics*, 10:315, 2009. doi: 10.1186/1471-2164-10-315.

M. Barakat, P. Ortet, and D. E. Whitworth. P2CS: a database of prokaryotic two-component systems. *Nucleic Acids Research*, 39(Database issue):D771–776, 2011. doi: 10.1093/nar/gkq1023.

Sankar Basu and Björn Wallner. DockQ: A quality measure for protein-protein docking models. *PLoS ONE*, 11(8):1–9, 2016. doi: 10.1371/journal.pone.0161879.

Anne-Florence Bitbol. Inferring interaction partners from protein sequences using mutual information. *PLoS Comput. Biol.*, 14(11):e1006401, 2018. doi: 10.1371/journal.pcbi.1006401.

Anne-Florence Bitbol, Robert S Dwyer, Lucy J Colwell, and Ned S Wingreen. Inferring interaction partners from protein sequences. *Proc. Natl. Acad. Sci. U.S.A.*, 113(43):12180–12185, 2016. doi: 10.1073/pnas.1606762113.

S. Bradde, A. Braunstein, H. Mahmoudi, F. Tria, M. Weigt, and R. Zecchina. Aligning graphs and finding substructures by a cavity approach. *EPL*, 89(3), 2010. doi: 10.1209/0295-5075/89/37009.

Bo Chen, Ziwei Xie, Jiezhong Qiu, Zhaofeng Ye, Jinbo Xu, and Jie Tang. Improved the heterodimer protein complex prediction with protein language models. *Briefings in Bioinformatics*, 24(4): bbad221, 2023. doi: 10.1093/bib/bbad221.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, 2021. doi: 10.1101/2021.10.04.463034.

Carlos A. Gandarilla-Perez, Sergio Pinilla, Anne-Florence Bitbol, and Martin Weigt. Combining phylogeny and coevolution improves the inference of interaction partners among paralogous proteins. *PLoS Comput. Biol.*, 19(3):e1011010, 2023. doi: 10.1371/journal.pcbi.1011010.

Mikhail Goncharov, Dmitry Bagaev, Dmitrii Shcherbinin, Ivan Zvyagin, Dmitry Bolotin, Paul G. Thomas, Anastasia A. Minervina, Mikhail V. Pogorelyy, Kristin Ladell, James E. McLaren, David A. Price, Thi H.O. Nguyen, Louise C. Rowntree, E. Bridie Clemens, Katherine Kedzierska, Garry Dolton, Cristina Rafael Rius, Andrew Sewell, Jerome Samir, Fabio Luciani, Ksenia V. Zornikova, Alexandra A. Khmelevskaya, Saveliy A. Sheetikov, Grigory A. Efimov, Dmitry Chudakov, and Mikhail Shugay. VDJdb in the pandemic era: a compendium of T cell receptors specific for SARS-CoV-2. *Nature Methods 2022 19:9*, 19(9):1017–1019, August 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01578-0. URL https://www.nature.com/articles/s41592-022-01578-0. Publisher: Nature Publishing Group.

J. M. Izarzugaza, D. Juan, C. Pons, F. Pazos, and A. Valencia. Enhancing the prediction of protein pairings between interacting families using orthology information. *BMC Bioinformatics*, 9:35, 2008. doi: 10.1186/1471-2105-9-35.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955. doi: 10.1002/nav.3800020109.

Umberto Lupo, Damiano Sgarbossa, and Anne-Florence Bitbol. Pairing interacting protein sequences using masked language modeling. *bioRxiv*, 2023. doi: 10.1101/2023.08.14.553209. URL https://www.biorxiv.org/content/early/2024/01/05/2023.08.14.553209.

Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. FGOT: Graph distances based on filters and optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7710–7718, 2022a.

Hermina Petric Maretic, Mireille El Gheche, Matthias Minder, Giovanni Chierchia, and Pascal Frossard. Wasserstein-based graph alignment. *IEEE Transactions on Signal and Information Processing over Networks*, 8:353–363, 2022b.

Gonzalo E. Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–22, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Hermina Petric Maretic, Mireille El Gheche, Giovanni Chierchia, and Pascal Frossard. Got: an optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32, 2019.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA Transformer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8844–8856. PMLR, 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

A1 INDEPENDENCE OF NUMBER OF SINKHORN NORMALIZATIONS

Let S^ℓ (for an integer $\ell > 0$) be the ℓ-truncated Sinkhorn operator defined in Eq. (1). In this section we prove that, for any $\ell > 1$, all first-order derivatives of S^ℓ, when evaluated at the zero matrix, are equal to the corresponding derivatives of $S^{\ell=1}$.

Let $\mathcal{R}$ (resp. $\mathcal{C}$) be the row-wise (resp. column-wise) matrix normalization operator on $D \times D$ matrices. Denote the partial derivative operator with respect to the (i, j)-th matrix entry by ∂_{ij}, and the (k, l)-th matrix component of a matrix-valued operator $\mathcal{O}$ by $\mathcal{O}_{kl}$. Furthermore, let $\mathcal{T}^* = \mathcal{C} \circ \mathcal{R}$. Let $\mathbf{1}_{\text{mat}}$ denote the $D \times D$ matrix whose entries are all equal to 1. Given the definition of the Sinkhorn operator in Eq. (1), and since the componentwise exponential of the zero matrix is a matrix of ones, it suffices, for our purposes, to show that

$$[\partial_{ij}(\mathcal{T}^* \circ \mathcal{T}^*)_{kl}](\mathbf{1}_{\text{mat}}) = [\partial_{ij}\mathcal{T}^*_{kl}](\mathbf{1}_{\text{mat}}) \tag{6}$$

for all $i, j, k, l = 1, \dots, D$. Indeed, we will prove that this actually holds when both sides of Eq. (6) are evaluated at $\mu\mathbf{1}_{\text{mat}}$, for any real number $\mu > 0$.

Proof of Eq. (6). Let $\boldsymbol{X}$ denote a $D \times D$ matrix with positive entries. We begin by noting that

$$[\partial_{ij}\mathcal{R}_{kl}](\boldsymbol{X}) = \delta_{ik}\,[\partial_j\mathcal{T}_l](\boldsymbol{X}_{i\cdot}) \quad \text{and} \quad [\partial_{ij}\mathcal{C}_{kl}](\boldsymbol{X}) = \delta_{jl}\,[\partial_i\mathcal{T}_k](\boldsymbol{X}_{\cdot j})\,, \tag{7}$$

where δ denotes the Kronecker delta, $\mathcal{T}$ the normalization operator for D-dimensional vectors, and $\boldsymbol{X}_{i\cdot}$ (resp. $\boldsymbol{X}_{\cdot j}$) the i-th row (resp. j-th column) of $\boldsymbol{X}$.

Let $\boldsymbol{x}$ denote a D-dimensional vector. The partial derivatives of the components of $\mathcal{T}$, evaluated at $\boldsymbol{x}$, are given by

$$\partial_\alpha\mathcal{T}_\beta(\boldsymbol{x}) = \frac{\partial}{\partial\alpha}\frac{x_\beta}{\sum_\gamma x_\gamma} = \frac{\delta_{\alpha\beta}}{\sum_\gamma x_\gamma} - \frac{x_\beta}{(\sum_\gamma x_\gamma)^2}. \tag{8}$$

Let $\mathbf{1}$ denote the D-dimensional vector whose entries are all equal to 1. Applying Eq. (8) to $\boldsymbol{x} = \mu\mathbf{1}$ yields, for any real number $\mu > 0$,

$$\partial_\alpha\mathcal{T}_\beta(\mu\mathbf{1}) = \frac{1}{D\mu}(\delta_{\alpha\beta} - 1/D) = \frac{1}{D\mu}\boldsymbol{\Delta}_{\alpha\beta}, \tag{9}$$

where we have defined the square matrix $\boldsymbol{\Delta} = \text{Id} - \mathbf{1}_{\text{mat}}/D$, with Id denoting the $D \times D$ identity matrix. We remark for later that $\mathbf{1}_{\text{mat}}/D$ is idempotent (i.e., its matrix square is itself), and that therefore so is $\boldsymbol{\Delta}$.

Using the chain rule and the fact that $\mathcal{R}(\mu\mathbf{1}_{\text{mat}}) = \mathbf{1}_{\text{mat}}/D$ for any $\mu > 0$, we can write

$$\begin{aligned}
[\partial_{ij}\mathcal{T}^*_{kl}](\mu\mathbf{1}_{\text{mat}}) &= \sum_{m,n}[\partial_{mn}\mathcal{C}_{kl}](\mathbf{1}_{\text{mat}}/D)\,[\partial_{ij}\mathcal{R}_{mn}](\mu\mathbf{1}_{\text{mat}}) \\
&= \sum_{m,n}\delta_{nl}\boldsymbol{\Delta}_{mk}\,\frac{\delta_{im}}{D\mu}\boldsymbol{\Delta}_{jn} \\
&= \frac{1}{D\mu}\boldsymbol{\Delta}_{ik}\boldsymbol{\Delta}_{jl}\,,
\end{aligned} \tag{10}$$

where we used Eq. (7) and Eq. (9) to obtain the second line.

We now compute

$$\begin{aligned}
[\partial_{ij}(\mathcal{T}^* \circ \mathcal{T}^*)_{kl}](\mu\mathbf{1}_{\text{mat}}) &= \sum_{mn}[\partial_{mn}\mathcal{T}^*_{kl}](\mathbf{1}_{\text{mat}}/D)\,[\partial_{ij}\mathcal{T}^*_{mn}](\mu\mathbf{1}_{\text{mat}}) \\
&= \frac{1}{D\mu}\sum_{mn}\boldsymbol{\Delta}_{mk}\boldsymbol{\Delta}_{nl}\boldsymbol{\Delta}_{im}\boldsymbol{\Delta}_{jn} \\
&= \frac{1}{D\mu}(\boldsymbol{\Delta}^2)_{ik}(\boldsymbol{\Delta}^2)_{jl} \\
&= [\partial_{ij}\mathcal{T}^*_{kl}](\mu\mathbf{1}_{\text{mat}}),
\end{aligned}$$

where the last equality follows from the idempotency of $\boldsymbol{\Delta}$ and from Eq. (10). When $\mu = 1$, this proves Eq. (6). □

A2 DATASETS

Benchmark prokaryotic datasets. We developed and tested DiffPaSS using joint MSAs extracted from a dataset composed of 23,632 cognate pairs of histidine kinases (HK) and response regulators (RR) from the P2CS database (Barakat et al., 2009; 2011), paired using genome proximity, and previously described in Bitbol et al. (2016); Bitbol (2018). Our focus is on pairing interaction partners among paralogs within each species. Pairing is trivial for a small number of species comprising only one pair of sequences. Hence, these species were discarded from the dataset. The average number of pairs per species in the resulting dataset is 11.0.

From this benchmark dataset of known interacting pairs, we extract paired MSAs of average depth 50, 100, 200, 300, 400, 500, 750 or 1000, constructed by selecting all the sequences of randomly sampled species from the full dataset. Each depth bin contains at least 200 MSAs. More precisely, for a target MSA depth $\overline{D} = 50$, 100, 200, 300, 400, 500, 750 or 1000, we add randomly sampled complete species one by one; if the first m species (but no fewer) give an MSA depth $D \geq 0.9\overline{D}$, and the first $n \geq m$ species (but no more) give $D \leq 1.1\overline{D}$, then we select the first k species in our final MSA, with k picked uniformly at random between m and n.

Eukaryotic complexes. We consider 15 heteromeric eukaryotic targets whose structures are not in the training set of AFM with v2 weights, already considered in Lupo et al. (2023).

T-cell receptor paired CDR3α-CDR3β data. We downloaded the full VDJDb database (Goncharov et al., 2022) and removed all entries where only a single TCR chain is available. For each epitope, we removed duplicate TCRs (defined at the level of α/β) and retained only epitopes for which at least 100 and no more than 10,000 sequences were available. Patient metadata from the database is used to define groups that permutations are to be restricted to. Our final dataset contains 19 pairs of CDR3α-CDR3β sequence collections of highly variable total size (ranging from 103 to 704 sequence pairs) and mean group size (see Fig. A7 for the expected fraction of correct pairs under random pairing), for which the ground truth matchings are known.

A3 GENERAL POINTS ON AFM

For all structure prediction tasks, we use the five pre-trained AFM models with v2 weights (Evans et al., 2021). We use full genomic databases and code from release v2.3.1 of the official implementation in `https://github.com/deepmind/alphafold`. We use no structural templates, and perform 3 recycles for each structure, without early stopping. We relax all models using AMBER.

We pair the same subset of pairable sequence retrieved by AFM as Lupo et al. (2023), and refer to Lupo et al. (2023, Table S1) for details on the pairable MSAs. For all structures, we use the query protein pair as fixed context for DiffPaSS.

The AFM confidence score is defined as $0.8 \cdot \mathrm{iptm} + 0.2 \cdot \mathrm{ptm}$, where iptm is the predicted TM-score in the interface, and ptm the predicted TM-score of the entire complex (Evans et al., 2021).

A4 DIFFPASS PAIRING QUALITY AFTER THE FIRST GRADIENT STEP

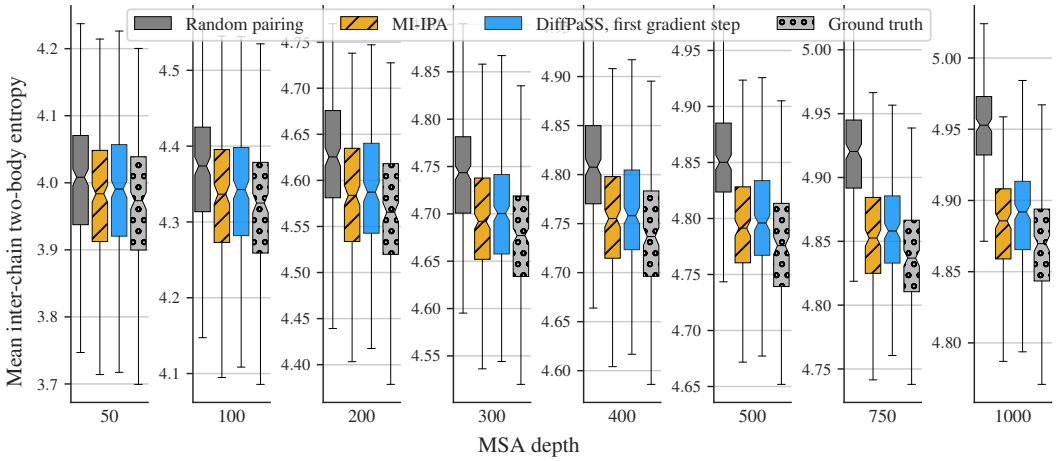

Figure A1: Comparing the distributions of inter-chain two-body entropy losses between MI-IPA and DiffPaSS after the first gradient step.

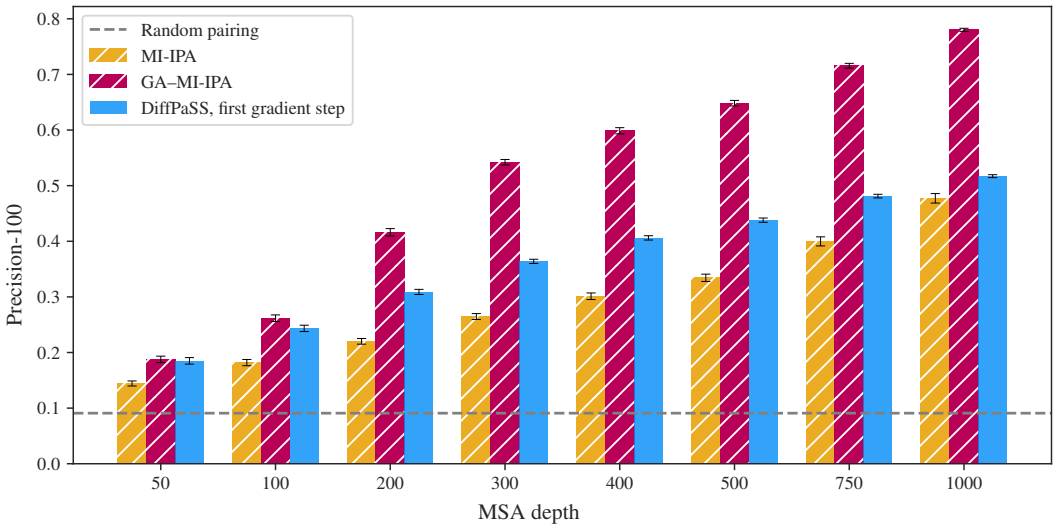

Figure A2: Pairing performance by DiffPaSS after the first gradient step.

A5 QUALITY OF SIGNAL EXTRACTION BY DIFFPASS

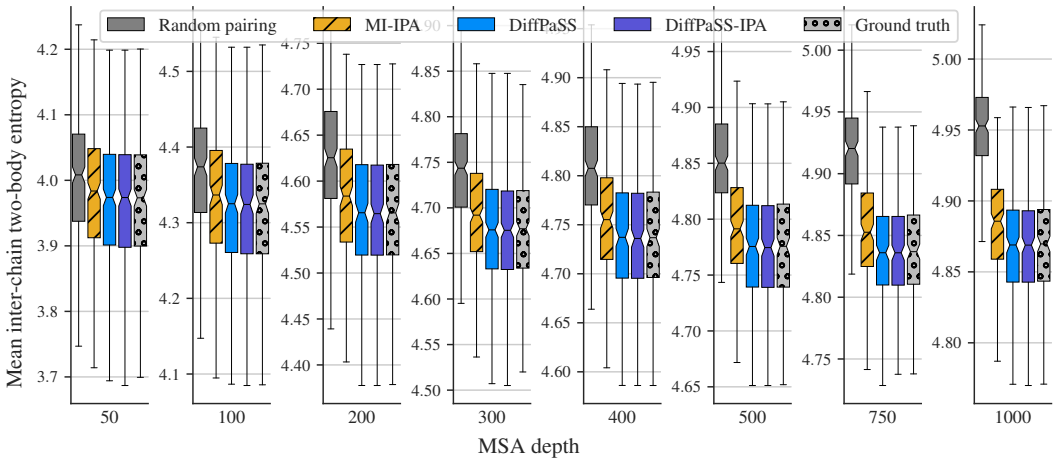

Figure A3: Distributions of the inter-chain two-body entropy losses for several methods, on the HK-RR dataset.

A6 ROBUST PAIRS

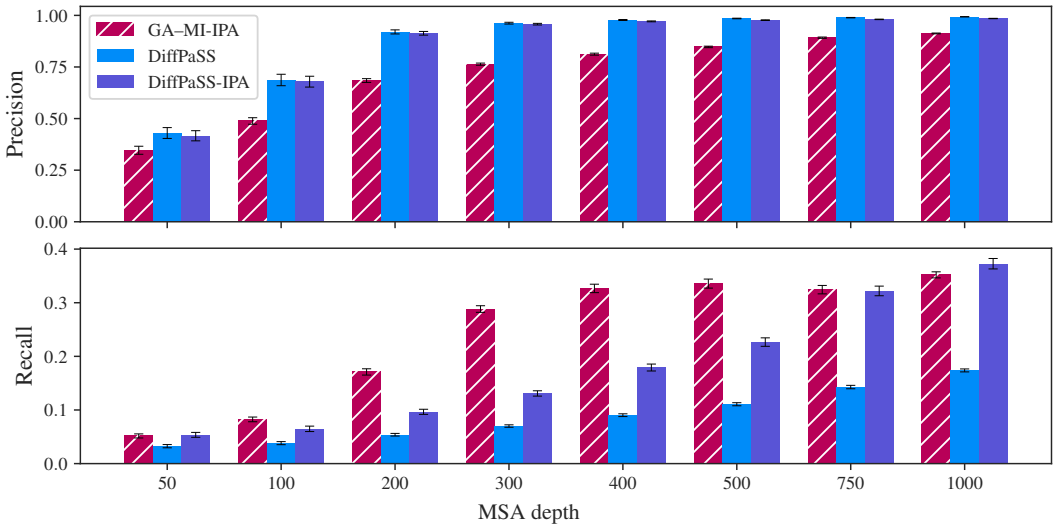

Figure A4: Precision and recall for the robust pairs found by GA–MI-IPA, DiffPaSS, and DiffPaSS-IPA.

A7 ALPHAFOLD-MULTIMER CONFIDENCE SCORES ON OUR EUKARYOTIC TARGETS

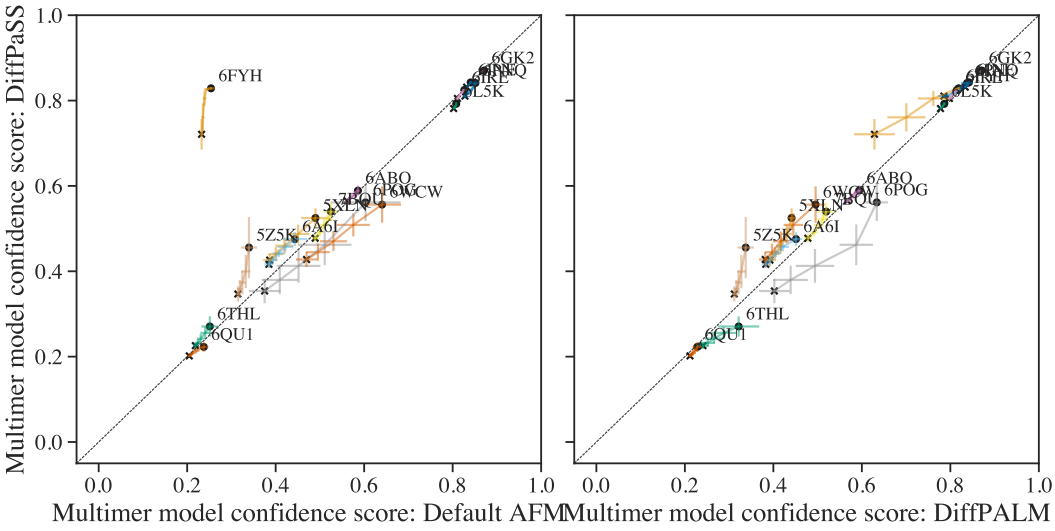

Figure A5: Multimer model confidence for structure prediction by AFM using different MSA pairing methods. See Appendix A3 for a definition of this score. "Trajectory visualizations" and results for Default AFM and DiffPALM-based pairing are as in Lupo et al. (2023, Fig. S5).

A8 DIFFPASS OPTIMIZATION AND PAIRING QUALITY ON THE CDR3α-CDR3β DATASET

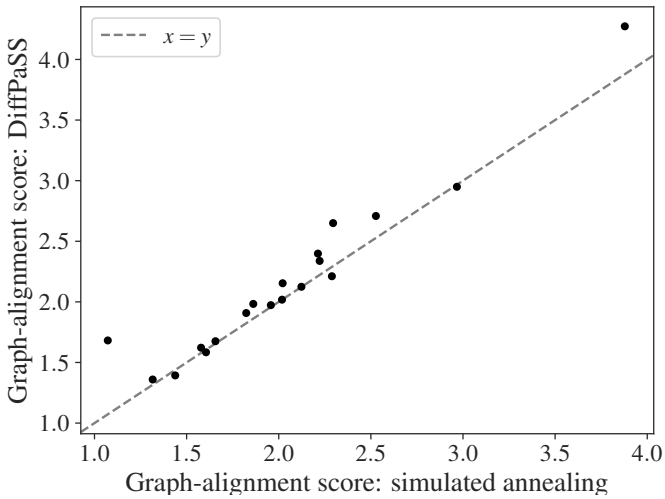

Figure A6: Mean normalized graph alignment scores per epitope in the paired CDR3α-CDR3β dataset, across 100 runs of both DiffPaSS and the simulated annealing graph alignment algorithm in Gandarilla-Perez et al. (2023). Each point represents one of the epitopes in the dataset, and higher scores indicate more successful optimization. GA scores are as defined in (Gandarilla-Perez et al., 2023), but normalized by the number of sequences to pair in each case.

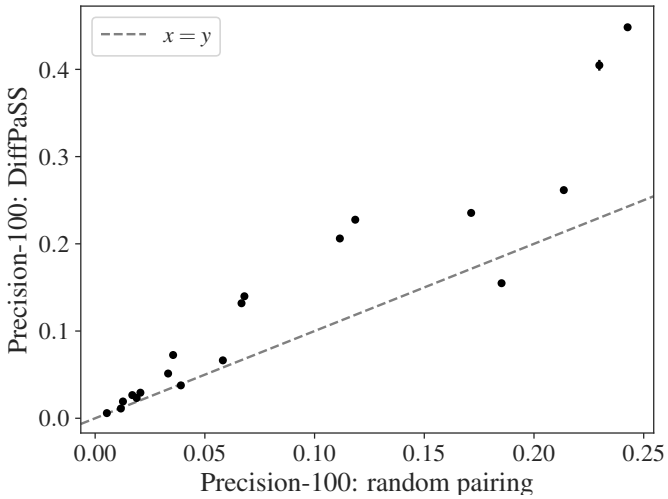

Figure A7: Mean pairing performance per epitope in the paired CDR3α-CDR3β dataset, across 100 runs of DiffPaSS. Each point represents one of the epitopes in the dataset.