POLITECNICO DI TORINO
Repository ISTITUZIONALE

Data driven topology optimization of AM parts accounting for process-affected fatigue performance: Application to automotive and aerospace components

Original
Data driven topology optimization of AM parts accounting for process-affected fatigue performance: Application to automotive and aerospace components / Centola, A., Ciampaglia, A., Boursier Niutta, C., Berto, F., Paolino, D.S., Tridello, A.. - In: INTERNATIONAL JOURNAL OF FATIGUE. - ISSN 0142-1123. - 206:(2026). [10.1016/j.ijfatigue.2026.109481]

Availability:
This version is available at: 11583/3009771 since: 2026-04-10T08:08:48Z

Publisher:
Elsevier

Published
DOI:10.1016/j.ijfatigue.2026.109481

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

07 October 2026

Bioinformatics, 2023, **39(9)**, btad537
https://doi.org/10.1093/bioinformatics/btad537
Advance Access Publication Date: 30 August 2023
Applications Note

OXFORD

Sequence analysis

DCAlign v1.0: aligning biological sequences using co-evolution models and informed priors

Anna Paola Muntoni [1,2,*] **and Andrea Pagnani** [1,2,3]

[1]Italian Institute for Genomic Medicine, IRCCS Candiolo, I-10060 Candiolo (TO), Italy
[2]Politecnico di Torino, I-10129 Torino, Italy
[3]INFN, Sezione di Torino, Torino, Via Pietro Giuria 1, I-10125 Torino, Italy

*Corresponding author. Italian Institute for Genomic Medicine [IIGM], Torre della Ricerca dell'IRCCS di Candiolo, SP142 Km. 3.95, Candiolo I-10060, Italy. E-mail: anna.muntoni@polito.it (A.P.M.)

Associate Editor: Christina Kendziorski

Abstract

Summary: DCAlign is a new alignment method able to cope with the conservation and the co-evolution signals that characterize the columns of multiple sequence alignments of homologous sequences. However, the pre-processing steps required to align a candidate sequence are computationally demanding. We show in v1.0 how to dramatically reduce the overall computing time by including an empirical prior over an informative set of variables mirroring the presence of insertions and deletions.

Availability and implementation: DCAlign v1.0 is implemented in Julia and it is fully available at https://github.com/infernet-h2020/DCAlign.

1 Introduction

A common task in *Bioinformatics* is to cast evolutionary-related biological sequences into a multiple sequence alignment (MSA). The objective of this task is to identify and align conserved regions of the sequences by maximizing the similarity among the columns of the MSA. State-of-the-art alignment methods, like `HMMER` for proteins (Eddy 2011) and `Infernal` (Nawrocki and Eddy 2013) for RNAs, use hand-curated MSAs of small representative subsets of sequences to be aligned (the so-called *seed* alignments). Whereas for proteins, `HMMER` builds the Hidden Markov Model (HMM) by using only the seed alignment, `Infernal` needs also secondary structure information to generate a Covariance Model (CM). In both cases, HMM (for proteins) or CM (for RNAs) are used to align query sequences. However, homologous sequences show signals of correlated mutations (epistasis) undetected by profile models.

Conservation and co-evolution signals are at the basis of Direct Coupling Analysis (DCA)-based statistical models (Morcos *et al.* 2011, Cocco *et al.* 2018). Recently, these models have been used to align biological sequences (Muntoni *et al.* 2020) and perform remote homology search (Wilburn and Eddy 2020) by alignment of the sequences to a seed model, or by pairwise alignments of seed models (Talibart and Coste 2021). The method in Muntoni *et al.* (2020), viz. `DCAlign`, returns the ordered sub-sequence of a query unaligned sequence which maximizes an objective function related to the DCA model of the seed. In this latter case, standard DCA models fail to adequately describe the statistics of insertions and gaps. To alleviate this limitation, we added to the objective function gap and insertion penalties learned from the seed alignment. While for the insertions, the computational complexity is negligible, inferring gap penalties is a time-consuming problem [see (Muntoni *et al.* 2020) and Supplementary text]. Here, we treat penalties in terms of informed priors computed from the seed sequences. The parameters for gaps and insertions, extracted from the seed alignment, are determined in an unsupervised manner. Finally, to further speed up the learning of the seed-based objective function, we obtain the parameters of the DCA model using pseudo-likelihood maximization (Ekeberg *et al.* 2013) instead of Boltzmann Machine Learning (Figliuzzi *et al.* 2018, Muntoni *et al.* 2021). `DCAlign v1.0`, is a computational pipeline that allows for the computation of the seed-model parameters in a few minutes, contrary to its original implementation which required at least a day of computation in the best scenario. The alignment problem is then solved approximately through a message-passing algorithm (see Supplementary text).

2 Methods

Our alignment algorithm estimates the optimal ordered sub-sequence compatible with a DCA model and empirical knowledge of insertions and gaps of the seed. Let $\boldsymbol{A}$ be an unaligned sequence of length N, and $\boldsymbol{S}$ be its aligned counterpart of length L (which is the length of the seed MSA). We only consider the $L \leq N$ case. At each $i = 1, \ldots, L$, we define a Boolean variable $x_i \in \{0, 1\}$ and a pointer $n_i \in \{0, \ldots, N+1\}$. The variable x_i indicates whether the position i is a *gap* '-' ($x_i = 0$) or a *match*, *i.e.* a symbol in $\boldsymbol{A}$. When i is a match, n_i identifies where S_i matches $\boldsymbol{A}$, i.e. $S_i = A_{n_i}$; instead, for $x_i = 0$, the value of n_i is used for keeping track of the last matched symbol in $\boldsymbol{A}$. Let us define a pointer-difference variable as $\Delta n_{i,j} = n_j - n_i$ for $i = 1, \ldots, L$ and $j > i$. Each auxiliary variable $\Delta n_{i,j}$ quantifies how

Received: 20 May 2022; **Revised:** 14 June 2023; **Editorial Decision:** 13 August 2023; **Accepted:** 29 August 2023
© The Author(s) 2023. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btad537/7255914 by guest on 09 September 2023

many symbols of the unaligned sequence $\boldsymbol{A}$ are present between two i, j positions of the aligned counterpart $\boldsymbol{S}$. If a configuration of the $\boldsymbol{n}$ is given, the full set of the pointer differences reveal the presence of insertions and gaps between any columns i and j of the alignment (see Supplementary text).

2.1 Seed modeling

Together with a DCA model of the aligned seed (see Fig. 1, central panel), for every site i (in red), we compute the $\Delta n_{i,j}$ for $j > i$ for all the seed sequences, and we learn an empirical probability $P_{i,j}(\Delta n_{i,j})$ as shown in the bottom central panel of Fig. 1 (this procedure is computationally very fast). The color gradient is associated with the value of j, the lighter the color, the larger is j. In Fig. 1 (bottom central panel), we consider as an example three sequences differing in the nature of the $\Delta n_{i,i+1}$.

2.2 Alignment procedure

We can express the alignment problem in terms of the following optimization problem: $\boldsymbol{x}, \boldsymbol{n} = \text{argmax}_{\overline{\boldsymbol{x}},\overline{\boldsymbol{n}}} \frac{e^{-\beta\mathcal{H}(\overline{\boldsymbol{x}},\overline{\boldsymbol{n}})}}{Z(\beta)} \prod_{i,j} P_{ij}^{\beta}(\overline{\boldsymbol{n}})$, where $\mathcal{H}$ is the DCA model describing the seed (see Fig. 1, top central panel), Z is a normalization factor, and β is a free parameter whose relevance will be discussed below. The maximization only runs over the feasible assignment of the variables, i.e. we impose that $n_{i+1} > n_i$ for every column i. The informed prior will guide the optimization process toward solutions that, among those that maximize the Boltzmann distribution associated with $\mathcal{H}$, reproduce the statistics of the seed pointer differences. Unfortunately, the problem thus stated is unfeasible as the normalization function Z cannot be efficiently computed. Similarly to the first `DCAlign` version, we use an approximate message-passing algorithm coupled with an annealing scheme over β (i.e. we iteratively increase β) to get the best alignment for the query sequence $\boldsymbol{A}$ (see Supplementary text and Supplementary Fig. S2).

3 Results

We can classify the type of tests performed to assess the performance of our computational strategy into three different categories:

- *Comparison with the previous implementation:* As in Muntoni *et al.* (2020), we compared our results against HMMER, Infernal (the last algorithm only for RNA sequences) on four Pfam (PF00035, PF00677, PF00684, PF00763), and Rfam (RF00059, RF00162, RF00167, RF01734) families. A detailed description of the dataset is contained in Supplementary Tables S2 and S3. We utilized the following comparison metrics: (i) the positive predictive value (PPV) of the DCA-based contact prediction (Morcos *et al.* 2011, Cocco *et al.* 2018), (ii) the proximity measures between the generated and the seed MSAs. As far as the contact map prediction is concerned, we observe either a mild improvement or a similar performance. With respect to the proximity measures, we notice a negligible increase in the average distance between seed sequences and generated alignments (see Supplementary Figs S3–S6 and Supplementary Tables S7–S10).
- *Leave-one-out experiment:* As a stress test for DCAlign v1.0 we also compared our results to 25 ground-truth MSAs either extracted from benchmark sets (Bahr *et al.* 2001, Thompson *et al.* 2005, Freyhult *et al.* 2007) or built from structural alignments (Akdel *et al.*, 2020) (see Supplementary Tables S2, S4, and S5). The numerical experiments consist of iteratively excluding one of the sequences of the reference alignment and training HMM, CM, or DCAlign using the remaining sequences. The excluded sequence is then aligned and quantitatively compared to the ground truth (viz. the structural alignment, or the benchmark sets). The emerging picture depends on the data type considered: for benchmark sets all computational strategies seem to perform reasonably well. In particular, HMMER (resp. Infernal) and our algorithm provide similar outcomes for protein (resp. RNA) domains (see Supplementary Figs S7–S10 and Supplementary Tables S11 and S12). However, when we consider structural alignments as our reference ground truth, our method significantly outperforms HMMER as shown in Supplementary Figs S11 and S12 and Supplementary Tables S13 and S14.
- *Divergent sequence alignment:* Finally, to assess our algorithm's remote homology detection performance, we considered three RNA benchmark sets (the seed of Rfam

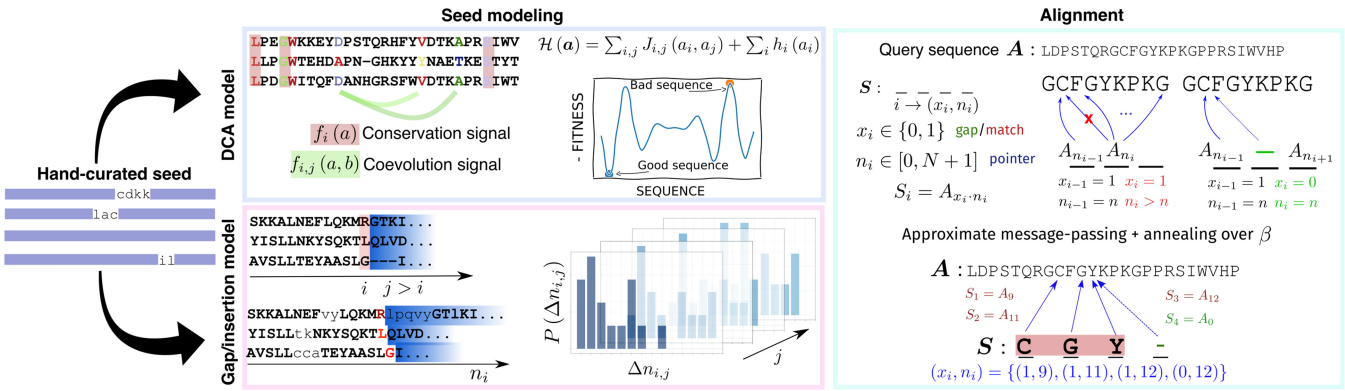

Figure 1. Schematic representation of the DCAlign v1.0 pipeline. From a (given) hand-curated alignment (the seed, shown in the left panel), our algorithm learns (i) a DCA model $\mathcal{H}$ exploiting the one-site and two-site statistics of the seed (upper central box), and (ii) the gap and insertion penalties by means of the empirical distribution of the pointer differences $P(\Delta n_{ij})$ for $i = 1, \ldots, L$, and $j > i$ (bottom central box). The three sequences represent the three scenarios that can occur between position i and $j = i + 1$: some insertion can appear, no insertion and no gap is present, or $i + 1$ contain a gap, so $\Delta n_{i,i+1} = 0$. For $j > i + 1$ (gradient shaded region on the left end of the sequence), both insertions and matched symbols contribute to the computation of the $\Delta n_{i,j}$, while gaps do not carry any contribution (see Supplementary Fig. S1 for a more detailed example). The alignment problem is then mapped into a constrained optimization problem over the $(\boldsymbol{x}, \boldsymbol{n})$ variables. The constraints on the variables and an example of alignment are shown in the right panel.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btad537/7255914 by guest on 09 September 2023

RF00162 (Kalvari *et al.* 2020), Twister type P1 (Roth *et al.* 2014), tRNA (Sprinzl *et al.* 1998), see Supplementary Table S6) from (Wilburn and Eddy 2020). Results suggest that Infernal is the best-performing method on two of the three datasets, while our method achieves the best alignment for the tRNA case. Note that Infernal is trained using secondary structure information that our algorithm does not use. All results are presented in Supplementary Fig. S13 and Supplementary Table S15.

From a computational efficiency point of view, the time needed to train the algorithm is significantly smaller than both our old implementation and CM-Infernal (see Supplementary text and Supplementary Fig. S14). However, the time necessary to align a sequence is equivalent compared to `DCAlign`, and probably to other computational strategies taking into account epistasis (Wilburn and Eddy 2020, Talibart and Coste 2021).

4 Conclusion

`DCAlign v1.0` is a new implementation of the DCA-based alignment technique, `DCAlign`, which conversely to the first implementation, allows for a fast parametrization of the seed alignment. The new modeling significantly drops the pre-processing time and guarantees a qualitatively equivalent alignment of a set of target sequences.

Acknowledgements

We warmly thank Indaco Biazzo, Alfredo Braunstein, Louise Budzynski, and Luca Dall'Asta for interesting discussions.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

A.P.M. and A.P. acknowledge financial support from Marie Skłodowska-Curie, grant agreement no. 734439 (INFERNET).

Data availability

We created a Zenodo link associated with the GitHub repository https://zenodo.org/badge/latestdoi/269696171.

References

Akdel M, Durairaj J, de Ridder D *et al.* Caretta—a multiple protein structure alignment and feature extraction suite. *Comput Struct Biotechnol J* 2020;**18**:981–92.

Bahr A, Thompson JD, Thierry J-C *et al.* BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res* 2001;**29**:323–6.

Cocco S, Feinauer C, Figliuzzi M *et al.* Inverse statistical physics of protein sequences: a key issues review. *Rep Prog Phys* 2018;**81**:032601.

Eddy SR. Accelerated profile HMM searches. *PLoS Comput Biol* 2011;**7**:e1002195.

Ekeberg M, Lövkvist C, Lan Y *et al.* Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. *Phys Rev E Stat Nonlin Soft Matter Phys* 2013;**87**:012707.

Figliuzzi M, Barrat-Charlaix P, Weigt M. How pairwise coevolutionary models capture the collective residue variability in proteins? *Mol Biol Evol* 2018;**35**:1018–27.

Freyhult EK, Bollback JP, Gardner PP. Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. *Genome Res* 2007;**17**:117–25.

Kalvari I, Nawrocki EP, Ontiveros-Palacios N *et al.* Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Res* 2020;**49**:D192–200.

Morcos F, Pagnani A, Lunt B *et al.* Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc Natl Acad Sci USA* 2011;**108**:E1293–301.

Muntoni AP, Pagnani A, Weigt M *et al.* Aligning biological sequences by exploiting residue conservation and coevolution. *Phys Rev E* 2020;**102**:062409.

Muntoni AP, Pagnani A, Weigt M *et al.* adabmDCA: adaptive Boltzmann machine learning for biological sequences. *BMC Bioinformatics* 2021;**22**:528.

Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* 2013;**29**:2933–5.

Roth A, Weinberg Z, Chen AGY *et al.* A widespread self-cleaving ribozyme class is revealed by bioinformatics. *Nat Chem Biol* 2014;**10**:56–60.

Sprinzl M, Horn C, Brown M *et al.* Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res* 1998;**26**:148–53.

Talibart H, Coste F. PPalign: optimal alignment of Potts models representing proteins with direct coupling information. *BMC Bioinformatics* 2021;**22**:317.

Thompson JD, Koehl P, Ripp R *et al.* BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins* 2005;**61**:127–36.

Wilburn GW, Eddy SR. Remote homology search with hidden Potts models. *PLoS Comput Biol* 2020;**16**:e1008085.

Downloaded from https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btad537/7255914 by guest on 09 September 2023