BMC Bioinformatics

SOFTWARE Open Access

hoDCA: higher order direct-coupling analysis

Michael Schmidt[1*] and Kay Hamacher[1,2,3]

Abstract

Background: Direct-coupling analysis (DCA) is a method for protein contact prediction from sequence information alone. Its underlying principle is parameter estimation for a Hamiltonian interaction function stemming from a maximum entropy model with one- and two-point interactions. Vastly growing sequence databases enable the construction of large multiple sequence alignments (MSA). Thus, enough data exists to include higher order terms, such as three-body correlations.

Results: We present an implementation of hoDCA, which is an extension of DCA by including three-body interactions into the inverse Ising problem posed by parameter estimation. In a previous study, these three-body-interactions improved contact prediction accuracy for the PSICOV benchmark dataset. Our implementation can be executed in parallel, which results in fast runtimes and makes it suitable for large-scale application.

Conclusion: Our hoDCA software allows improved contact prediction using the `Julia` language, leveraging power of multi-core machines in an automated fashion.

Keywords: Contact prediction, Proteins, DCA

Background

Thanks to rapidly growing sequence databases, the prediction of protein contacts from sequence information has become an promising route for computational structural biophysics [1–4]. The so called direct-coupling analysis (DCA) uses a multiple sequence alignment (MSA) to predict residue contacts in a maximum entropy approach. Its high accuracy was shown in various studies [5–11] and also made it suitable for protein structure prediction software [12–14].

The DCA approach leads to a Potts model with probability for a sequence $\vec{\sigma} = (\sigma_1, \ldots, \sigma_N)$ given as $P(\vec{\sigma}) = \exp\left[-H(\vec{\sigma})\right]/Z$, with Hamiltonian $H(\vec{\sigma}) = -\sum_i^N h_i(\sigma_i) - \sum_{1\leq i<j\leq N} J_{ij}(\sigma_i,\sigma_j)$ consisting of local fields and two-body interactions and N being the length of the sequences. $Z = \sum_{\vec{\sigma}\in\mathcal{A}^N} P(\vec{\sigma})$ is the partition function as the sum over all sequences where each position is chosen from the alphabet $\mathcal{A}$. After estimation of parameters $\{h_i, J_{ij}\}$ from empirical sequences $\vec{\sigma}^{(b)}$, a contact prediction score for residue i and j can be obtained by taking the l_2-norm $\left\|J_{ij}\right\|_2$. In a recent study [15], an improved prediction accuracy was shown by incorporating three-body interactions $V_{ijk}(\sigma_i,\sigma_j,\sigma_k)$ into H, obtaining a three-body Hamiltonian

$$H^{(3)}(\vec{\sigma}) = -\sum_i^N h_i(\sigma_i) - \sum_{1\leq i<j\leq N} J_{ij}(\sigma_i,\sigma_j) - \sum_{1\leq i<j<k\leq N} V_{ijk}(\sigma_i,\sigma_j,\sigma_k).$$

Here, we present an implementation of this method, which we call hoDCA.

Implementation

hoDCA is implemented in the `julia` language (0.6.2) [16], and depends directly on a) the ArgParse [17] module for command-line arguments and b) on the GaussDCA [18] module for performing preprocessing operations on the MSA and the implicit dependencies for those packages. A typical command-line call is

*Correspondence: schmidt@cbs.tu-darmstadt.de
[1] Department of Physics, TU Darmstadt, Karolinenpl. 5, 64287 Darmstadt, Germany
Full list of author information is available at the end of the article

```
julia hoDCA.jl Example.fasta
Example.csv
-No_A_Map=1 -Path_Map=A_Map.csv
-MaxGap=0.9 -theta=0.2 -Pseudocount=4.0
-No_Threads=2 -Ign_Last=0
```

with input `Example.fasta` and output `Example.csv`. The latter consists of lists of all two-body contact scores J_{ij} separated by at least one residue along the backbone. The meaning of the remaining (optional) parameters will become clear in the following.

General notes. For inference of parameters $\left\{h_i, J_{ij}, V_{ijk}\right\}$, we use the mean-field approximation as described in [15] with a reduced alphabet for three-body couplings. This is accomplished by a mapping

$$\mu : \{l | l \leq q\} \rightarrow \left\{\alpha | \alpha \leq q_{\text{red}}\right\}, \tag{1}$$

with q being the full alphabet of the MSA and $q_{\text{red}} \leq q$. On the one hand, this accounts for the so called curse of dimensionality [19], occuring if the size of the MSA is too small to observe all possible q^3 combinations for each V_{ijk}. On the other hand, this significantly reduces memory usage and allows for a faster computation of contact prediction scores. The mapping μ can be specified by `Path_Map`, which is a csv file with every row representing a mapping. `No_A_Map` tells which row to choose. As the bottleneck is still the calculation of three-body couplings, it can be performed using parallel threads by specifying the `No_Threads` flag.

In traditional DCA, the last amino acid q usually represents the gap character and is not taken into account for score computation within the l_2-norm. In hoDCA, each two-body coupling state $l \leq q$ contains contributions from $\{n \leq q | \mu(n) = \mu(l)\}$ due to the reduced alphabet. We therefore take gap contributions into account by default, which can be changed by the `Ign_Last` flag.

MSA preprocessing. The MSA is read in by the `GaussDCA` module, ignoring sequences with a higher amount of gaps than `MaxGap`, and subsequently converted into an array of integers. However, in contrast to `GaussDCA`, we check for the actual number of amino acids types contained in the MSA given. We, then, reduce the alphabet from $q = 21$ to the number of present characters (amino acid types). Afterwards, the reweighting for every sequence $\vec{\sigma}^{(b)}$ is obtained by the `GaussDCA` module via $w_b = 1/|\{a \in \{1, ..., B\} : \text{difference}\left(\vec{\sigma}^{(a)}, \vec{\sigma}^{(b)}\right) \leq \texttt{theta}\}|$, where the difference is computed by the percentage hamming distance [6]. The aim of reweighting is to reduce potential phylogenetic bias.

Frequency computation. Empirical frequency counts for the full alphabet are computed according to [6]

$$\begin{aligned} f_i(l) &= \frac{1}{\lambda_c + B_{eff}} \left(\frac{\lambda_c}{q} + \sum_{b=1}^{B} w_b \cdot \delta\left(\sigma_i^{(b)}, l\right) \right) \\ f_{ij}(l, m) &= \frac{1}{\lambda_c + B_{eff}} \left(\frac{\lambda_c}{q^2} \right. \\ &\quad \left. + \sum_{b=1}^{B} w_b \cdot \delta\left(\sigma_i^{(b)}, l\right) \delta\left(\sigma_j^{(b)}, l\right) \right), \end{aligned} \tag{2}$$

with δ being the Kronecker delta, B the number of sequences in the MSA, $B_{eff} = \sum_{b=1}^{B} w_b$ and $\lambda_c = \texttt{Pseudocount} \cdot B_{eff}$. The `Pseudocount` parameter shifts empirical data towards a uniform distribution. This is necessary to ensure invertibility of the empirical covariance matrix in the mean-field approach.

Frequency counts for the reduced alphabet are computed via

$$\begin{aligned} f_i^{\text{red}}(\alpha) &= \sum_{\{l|\mu(l)=\alpha\}} f_i(l) \\ f_{ij}^{\text{red}}(\alpha, \beta) &= \sum_{\substack{\{l|\mu(l)=\alpha\} \\ \{m|\mu(m)=\beta\}}} f_{ij}(l, m) \\ f_{ijk}^{\text{red}}(\alpha, \beta, \gamma) &= \sum_{\substack{\{l|\mu(l)=\alpha\} \\ \{m|\mu(m)=\beta\} \\ \{n|\mu(n)=\gamma\}}} f_{ijk}(l, m, n). \end{aligned} \tag{3}$$

The computation of three-point frequencies takes some time and will be executed on `No_Threads` threads. For this, we parallelized their calculation over the sequence size N, meaning that the i-th process computes f_{ijk}^{red} for all $k \geq j \geq i$ and fixed i. Besides the parallelization scheme, three-point frequencies are preprocessed in the same manner as one- and two-point frequencies.

Contact prediction scores. Contact prediction scores follow directly from two-body couplings. Two-body couplings are obtained within the mean-field approximation by

$$\begin{aligned} J_{ij}(l, m) \approx &- g_{ij}(l, m) \\ &+ \sum_{\substack{k=1, \\ k \neq i,j}}^{N} \sum_{n=1}^{q-1} g_{ijk}^{\text{red}}(\mu(l), \mu(m), \mu(n)) \cdot f_k(n), \end{aligned} \tag{4}$$

where $g_{ij}(l, m)$ is the inverse of the empirical two-point covariance matrix $e_{ij}(l, m) = f_{ij}(l, m) - f_i(l) f_j(m)$. $g_{ijk}^{\text{red}}(\alpha, \beta, \gamma)$ is given by a relation to the three-point covariance matrix over the reduced alphabet

$$\begin{aligned} e^{\text{red}}_{ijk}(\alpha,\beta,\gamma) =& f^{\text{red}}_{ijk}(\alpha,\beta,\gamma) + 2f^{\text{red}}_{i}(\alpha)f^{\text{red}}_{j}(\beta)f^{\text{red}}_{k}(\gamma)\\ &- f^{\text{red}}_{ij}(\alpha,\beta)f^{\text{red}}_{k}(\gamma)\\ &- f^{\text{red}}_{ik}(\alpha,\gamma)f^{\text{red}}_{j}(\beta)\\ &- f^{\text{red}}_{jk}(\beta,\gamma)f^{\text{red}}_{i}(\alpha) \end{aligned} \tag{5}$$

via

$$\begin{aligned} g^{\text{red}}_{ijk}(\alpha,\beta,\gamma) =& - \sum_{a_1,b_1,c_1=1}^{N} \sum_{a_2,b_2,c_2=1}^{q_{\text{red}}-1} \Big(e^{\text{red}}_{a_1,b_1,c_1}(a_2,b_2,c_2) \quad ,\\ & \cdot\, g^{\text{red}}_{i,a_1}(\alpha,a_2) \cdot g^{\text{red}}_{j,b_1}(\beta,b_2) \cdot g^{\text{red}}_{k,c_1}(\gamma,c_2) \Big) \end{aligned} \tag{6}$$

where $g_{ij}(\alpha,\beta)$ is the inverse of the two-point covariance matrix over the reduced alphabet (see [15] for more details). For the calculation of scores, $\{J_{ij}\}$ are transformed into so called zero-sum gauge, satisfying $\sum_{l}^{q} \hat{J}_{ij}(l,.) = \sum_{m}^{q} \hat{J}_{ij}(.,m) = 0$, where "." stands for an arbitrary state via

$$\begin{aligned} \hat{J}_{ij}(l,m) =& J_{ij}(l,m) + \frac{1}{q}\sum_{r=1}^{q}\Bigg[- J_{ij}(r,m) - J_{ij}(l,r)\\ &+ \frac{1}{q}\sum_{s=1}^{q} J_{ij}(r,s)\Bigg]\\ &+ \frac{1}{q_{\text{red}}}\sum_{\substack{k=1\\k\neq i,j}}^{N}\sum_{\eta=1}^{q_{\text{red}}}\Bigg[V^{\text{red}}_{ijk}(\mu(l),\mu(m),\eta)\\ &+ \frac{1}{q}\sum_{r=1}^{q}\Bigg[- V^{\text{red}}_{ijk}(\mu(r),m,\eta)\\ &- V^{\text{red}}_{ijk}(\mu(l),\mu(r),\eta)\\ &+ \frac{1}{q}\sum_{s=1}^{q} V^{\text{red}}_{ijk}(\mu(r),\mu(s),\eta)\Bigg]\Bigg] \end{aligned} \tag{7}$$

The purpose of the gauge transformation is to shift local bias from two-body couplings into local fields [8, 20]. Above calculations are the most time consuming parts and run on `No_Threads` threads. The final scores result from average product correction (APC) of l_2 norm [21] via

$$S_{ij} = \left\|\hat{\mathbf{J}}_{ij}\right\|_2 - \frac{\left\|\hat{\mathbf{J}}_{:j}\right\|_2 \left\|\hat{\mathbf{J}}_{i:}\right\|_2}{\left\|\hat{\mathbf{J}}_{::}\right\|_2} \tag{8}$$

and $\left\|\hat{\mathbf{J}}_{ij}\right\|_2 = \sqrt{\sum_{l,m=1}^{q} \hat{J}_{ij}(l,m)^2}$.

Discussion

A performance benchmark on the PSICOV-dataset [10], consisting of 150 proteins, is presented in [15]. For evaluating the performance of a single protein, the so called area under precision curve

$$A := \frac{1}{C}\sum_{i=1}^{C}\frac{p_i}{i} \tag{9}$$

was used, where C is the total amount of contacts and p_i is the number of true positives of the first i predictions. Figure 1 shows the predicted contact map of the protein data bank entry 1fx2A as an exemplary case. For this particular protein, the classical two-body DCA has an A-value of $A \approx 0.5$ while hoDCA shows a superior $A \approx 0.72$.

Interestingly, the majority of hoDCA's false positives are located in the lower and upper right corner of the contact map. We hypothesize that this finding is due to correlated gap regions in the corresponding MSA: For this particular pdb entry, many sequences were too short and had to be extended by gaps on both termini. This, in turn, leads to intra and inter correlations between the left and right termini. Figure 2 shows the two-point gap-gap frequencies of the non-preprocessed MSA (i.e. without sequence reweighting, pseudocount modification or deletion of sequences). As can be seen, there is indeed an accumulation of gap regions at the beginning and ending of the protein, thus possibly leading to false correlations.

Results

Figure 3 shows the runtime behavior of hoDCA when `No_Threads` are used for calculation of three-body

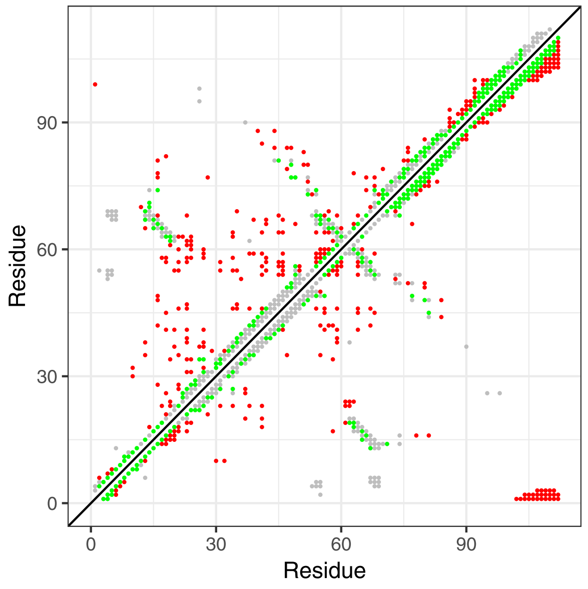

Fig. 1 Contact map of pdb entry 1fx2A (gray) with true positives (green) and false positives (red) for a distance threshold of 7.5 Å. Upper left: classical mean-field DCA. Lower right: hoDCA with a mapping classification according to polarity [24]

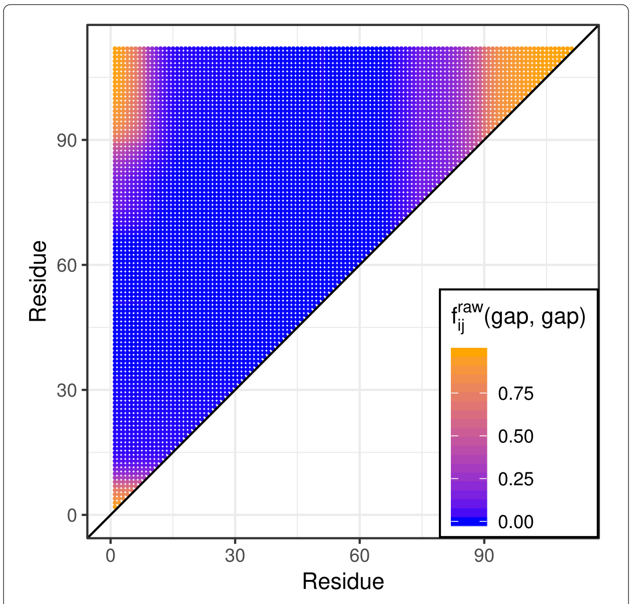

Fig. 2 Raw gap-gap frequencies for pdb entry 1fx2A

terms. We used entry 1tqhA for the benchmark, which has one of the largest MSAs in the PSICOV dataset ($N = 242$, $B = 18,170$) and parameters as in Eq. (2). The overall speedup is about five-fold when executed on $n \geq 12$ threads in comparison to a single CPU core. A fit of Amdahl's law $T = T_0 \cdot (1 - p \cdot (1 - 1/n))$, with T_0 being the single-threaded runtime and $n =$ `No_Threads`, reveals the proportion of parallelized routines as $p \approx 0.86$. The serial runtime proportion of ≈ 0.14 comes mainly due to computation of two-body terms. Also note that we did not modify the standard `julia` parameters, meaning, e.g., a parallel computation of the matrix inverse by default.

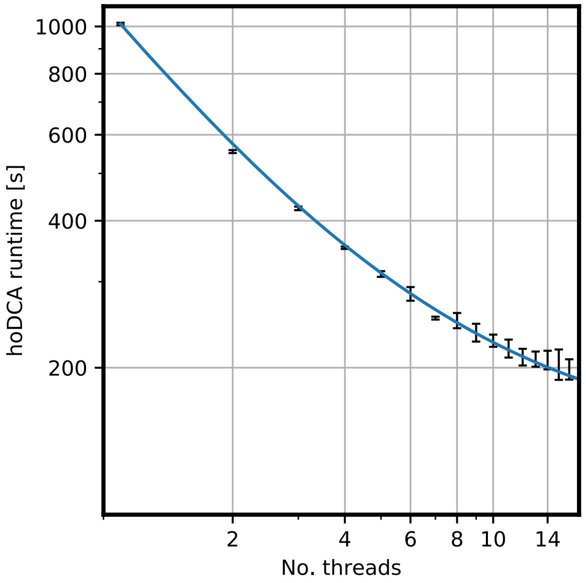

Fig. 3 Runtime behaviour of hoDCA for PSICOV entry 1tqhA. The benchmark system was a Debian-operating server with two `Intel(R) Xeon(R) CPU E5-2687W v2 @ 3.40GHz`. Runtimes were taken for `julia`-compiled code, thus potential initalization overhead is omitted. The solid line shows a fit of Amdahl's law

Conclusions

Higher-order interactions have been shown to have a strong influence on contact prediction in certain proteins [15, 22, 23]. Here, we implemented hoDCA, an extension of DCA by incorporating three-body couplings into the Hamiltonian. The accessible command-line user interface and the significant speedup within parallel execution make hoDCA suitable for contact prediction in a variety of proteins, using biochemical inspired alphabet reduction schemes. We hope to have made this method easily accessible for other researchers by this software release.

Availability and requirements

Project name: hoDCA

Project home page: http://www.cbs.tu-darmstadt.de/hoDCA/

Operating systems: Linux, Windows, macOS

Programming language: julia (0.6.2)

Other requirements: julia packages Argparse, GaussDCA

License: GNU General Public License v3, http://www.gnu.org/licenses/gpl-3.0.html

Any restrictions to use by non-academics: Any commercial use is subject to a contractual agreement between involved parties.

Abbreviations

APC: Average product correction; DCA: Direct-coupling analysis; MSA: Multiple sequence alignment

Acknowledgements

N/A

Funding

The authors thank the LOEWE project iNAPO funded by the Ministry of Higher Education, Research and the Arts (HMWK) of the Hessen state. The authors acknowledge financial support by the Deutsche Forschungsgemeinschaft via the Graduate School (GRK 1657) "Radiation Biology", project 1A. The funding bodies had no role in the design of the study or the collection/analysis/interpretation of data or the writing of the manuscript.

Availability of data and materials

The source code of our package and its documentation are available from the url http://www.cbs.tu-darmstadt.de/hoDCA.

Authors' contributions

KH conceived the study, MS wrote the software and documentation, analyzed data and prepared packaging; both authors wrote the paper. Both authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Department of Physics, TU Darmstadt, Karolinenpl. 5, 64287 Darmstadt, Germany. [2]Department of Biology, TU Darmstadt, Schnittspahnstr. 10, 64287 Darmstadt, Germany. [3]Department of Computer Science, TU Darmstadt, Karolinenpl. 5, 64287 Darmstadt, Germany.

Received: 18 March 2018 Accepted: 12 December 2018
Published online: 29 December 2018

References

1. Rost B, Sander C. Bridging the protein sequence-structure gap by structure predictions. Ann Rev Biophys Biomol Struct. 1996;25:113–36.
2. Yang J, Zhang Y. Protein Structure and Function Prediction Using I-TASSER. Curr Protoc Bioinforma. 2015;52:5.8.1–15. https://doi.org/10.1002/0471250953.bi0508s52.
3. Kaufmann KW, Lemmon GH, DeLuca SL, Sheehan JH, Meiler J. Practically useful: What the rosetta protein modeling suite can do for you. Biochemistry. 2010;49:2987–98.
4. Krieger E, Joo K, Lee J, Lee J, Raman S, Thompson J, Tyka M, Baker D, Karplus K. Improving physical realism, stereochemistry and side-chain accuracy in homology modeling: four approaches that performed well in CASP8. Proteins. 2009;77:114–22. https://doi.org/10.1002/prot.22570.
5. Weigt M, White RA, Szurmant H, Hoch JA, Hwa T. Identification of direct residue contacts in protein–protein interaction by message passing. Proc Natl Acad Sci. 2009;106(1):67–72. https://doi.org/10.1073/pnas.0805923106. http://www.pnas.org/content/106/1/67.full.pdf.
6. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, Zecchina R, Onuchic JN, Hwa T, Weigt M. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proc Natl Acad Sci. 2011;108(49):1293–301. https://doi.org/10.1073/pnas.1111471108. http://www.pnas.org/content/108/49/E1293.full.pdf.
7. Ekeberg M, Lövkvist C, Lan Y, Weigt M, Aurell E. Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. Phys Rev E. 2013;87:012707. https://doi.org/10.1103/PhysRevE.87.012707.
8. Ekeberg M, Hartonen T, Aurell E. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. J Comput Phys. 2014;276:341–56. https://doi.org/10.1016/j.jcp.2014.07.024.
9. Baldassi C, Zamparo M, Feinauer C, Procaccini A, Zecchina R, Weigt M, Pagnani A. Fast and accurate multivariate gaussian modeling of protein families: Predicting residue contacts and protein-interaction partners. PLoS ONE. 2014;9(3):1–12. https://doi.org/10.1371/journal.pone.0092721.
10. Jones DT, Buchan DWA, Cozzetto D, Pontil M. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics. 2012;28(2):184–90. https://doi.org/10.1093/bioinformatics/btr638. http://bioinformatics.oxfordjournals.org/content/28/2/184.full.pdf+html.
11. Stein RR, Marks DS, Sander C. Inferring pairwise interactions from biological data using maximum-entropy probability models. PLoS Comput Biol. 2015;11(7):1–22. https://doi.org/10.1371/journal.pcbi.1004182.
12. Jones DT, Singh T, Kosciolek T, Tetchner S. Metapsicov: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics. 2015;31(7):999. https://doi.org/10.1093/bioinformatics/btu791.
13. Sheridan R, Fieldhouse RJ, Hayat S, Sun Y, Antipin Y, Yang L, Hopf T, Marks DS, Sander C. Evfold.org: Evolutionary couplings and protein 3d structure prediction. bioRxiv. 2015. https://doi.org/10.1101/021022. http://www.biorxiv.org/content/early/2015/07/02/021022.full.pdf.
14. Skwark MJ, Raimondi D, Michel M, Elofsson A. Improved contact predictions using the recognition of protein like contact patterns. PLoS Comput Biol. 2014;10(11):1–14. https://doi.org/10.1371/journal.pcbi.1003889.
15. Schmidt M, Hamacher K. Three-body interactions improve contact prediction within direct-coupling analysis. Phys Rev E. 2017;96:052405. https://doi.org/10.1103/PhysRevE.96.052405.
16. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Rev. 2017;59(1):65–98. https://doi.org/10.1137/141000671.
17. Baldassi C. https://github.com/carlobaldassi/argparse.jl.
18. Baldassi C, Pagnani A, Weigt M, Feinauer C, Procaccini A, Zecchina R, Zamparo M. GaussDCA.jl - First release. 2014. https://doi.org/10.5281/zenodo.10814. https://github.com/carlobaldassi/GaussDCA.jl.
19. Chávez E, Navarro G, Baeza-Yates R, Marroquín JL. Searching in metric spaces. ACM Comput Surv. 2001;33(3):273–321.
20. Feinauer C, Skwark MJ, Pagnani A, Aurell E. Improving contact prediction along three dimensions. PLoS Comput Biol. 2014;10(10):1–13. https://doi.org/10.1371/journal.pcbi.1003847.
21. Dunn SD, Wahl LM, Gloor GB. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. Bioinformatics. 2008;24(3):333–40. https://doi.org/10.1093/bioinformatics/btm604. http://bioinformatics.oxfordjournals.org/content/24/3/333.full.pdf+html.
22. Waechter M, Jaeger K, Weissgraeber S, Widmer S, Goesele M, Hamacher K. Information-theoretic analysis of molecular (co)evolution using graphics processing units. In: Proceedings of the 3rd International Workshop on Emerging Computational Methods for the Life Sciences. ECMLS ’12. New York, NY, USA: ACM; 2012. p. 49–58. https://doi.org/10.1145/2483954.2483963. http://doi.acm.org/10.1145/2483954.2483963.
23. Waechter M, Jaeger K, Thuerck D, Weissgraeber S, Widmer S, Goesele M, Hamacher K. Using graphics processing units to investigate molecular coevolution. Concurr Comput Pract Experience. 2014;26(6):1278–96. https://doi.org/10.1002/cpe.3074.
24. Grantham R. Amino acid difference formula to help explain protein evolution. Science. 1974;185(4154):862–4. https://doi.org/10.1126/science.185.4154.862. http://science.sciencemag.org/content/185/4154/862.full.pdf.