diSTruct v1.0: Generating Biomolecular Structures from Distance Constraints

Oskar Taubert [1], Ines Reinartz [1], Henning Meyerhenke [2], and Alexander Schug [3,*]

January 17, 2020

Abstract

Summary: The distance geometry problem is often encountered in molecular biology and the life sciences at large, as a host of experimental methods produce ambiguous and noisy distance data. In this note, we present `diSTruct`; an adaptation of the generic MaxEnt-Stress graph drawing algorithm to the domain of biological macromolecules. `diSTruct` is fast, provides reliable structural models even from incomplete or noisy distance data and integrates access to graph analysis tools.
Availability and Implementation: diSTruct is written in C++, Cython and Python 3. It is available from https://github.com/KIT-MBS/distruct.gitor in the Python package index under the MIT license.
Contact: al.schug@fz-juelich.de
Supplementary information: Supplementary data is available at *Bioinformatics* online.

1 Introduction

Biomolecules orchestrate life at the molecular level. Any detailed insight into biomolecular function requires information about their structure. Typically, this structural information stems from experimental techniques such as nuclear magnetic resonance or X-ray scattering, although theoretical structure prediction techniques have become more and more accurate. The primary experimental information are, however, not directly structural models but refraction patterns (X-ray) or chemical shifts (NMR), which have to be interpreted. Particularly for NMR, assigned chemical shifts can be translated into distances between specific atoms. Large numbers of such pairwise distances define an incomplete distance matrix, which must then be processed to derive a structure for the specific biomolecule. The distance constraints are used as input for a range of modeling tools, often based on molecular force fields. For NMR, ARIA (CNS) ([Rieping et al., 2007, Bruger, 2007]), CYANA ([Güntert and Buchner, 2015]), or XPLOR-NIH ([Schwieters et al., 2006]) are popular examples. Similarly, the co-evolutionary analysis of large sequence alignments by techniques such as Direct Coupling Analysis provides information about spatial adjacency between residue pairs([Schug et al., 2009]). Apart from proteins, the challenge of interpreting distance matrices into structural models, the *distance geometry problem*, occurs also on other biological scales, *e. g.* in the case of HI-C data of chromatin strands providing distance information about chromosomal structure ([Lieberman-Aiden et al., 2009]). Often the structure cannot be derived uniquely

from the provided data. Instead, computationally demanding simulations based on empirical force fields and constrained by experimental data are run.

Here, we directly solve the distance geometry problem for arbitrary distance data from experimental or other sources. The key achievements are speed of interpretation, minimizing the error with respect to constraints and the ability to deal with sparse and noisy data. The rapid construction of 3D-models even for a large protein of 700 amino acids takes only on the order of 10 seconds on common CPUs. This allows to interactively interpret possible sources of ambiguity or errors in the input distance data and helps to improve, e.g., NMR shift assignment.

The algorithm is implemented in the python package `diSTruct`. `diSTruct` builds on Biopython ([Cock et al., 2009, Hamelryck and Manderick, 2003]), a toolkit for computational biology, and the NetworKit ([Staudt et al., 2016]) package for graph analysis. Our design focus is to provide an interactive python-based toolkit that a) provides structural models from distance data with minimal constraint errors, b) is able to handle noisy and incomplete data, c) maintains familiarity for users that know Biopython, and d) provides an interface from biological context to a rich graph analysis suite.

2 Method

At a basic level, `diSTruct` requires a matrix of inter-vertex distances (edges) as input. The individual vertices can be atoms or arbitrary beads. The edges have to form a connected graph and can be weighted (see below). For biomolecules basic constraints can be derived from their sequence and the known chemical structures of amino or nucleic acids. Due to theoretical considerations, a structure of N vertices requires $3N$ independent constraints; one per degree of freedom. Sequence information for biomolecules provides roughly two non-redundant constraints per atom: The bond distance between two neighboring atoms and the angle between two neighboring bonds. In this case, one additional constraint per atom should suffice for a well-defined structure.
The structure generation algorithm employed is MaxEnt-Stress graph drawing ([Hu et al., 2013]), similar to the implementation in the NetworKit suite ([Wegner et al., 2017]). It minimizes the total stress of the system

$$T = \sum_{uv \in E} \omega_{uv} (||\vec{r}_u - \vec{r}_v|| - d_{uv})^2 \; - \; \alpha S(\{\vec{r}\}) \stackrel{!}{=} \min\,, \tag{1}$$

where d_{uv} is the target distance between vertices u and v, ω_{uv} is the edge-weight and αS realizes a repulsive force between all vertices.
This can be written in terms of Laplacian matrices of the graph:

$$\mathbf{L}\mathbf{r}^i = \mathbf{L}_\mathrm{d}(\vec{r})\mathbf{r}^i + \alpha\, \mathbf{b}(\vec{r}), \tag{2}$$

for $i \in 1, 2, 3$ in three spatial dimensions, where each systems dimension is equal to the number of atoms. The three systems are solved iteratively in parallel. A detailed description is given in the SI. Fig. 1 shows different observable results for an example protein (PDB code `1mqq` ([Golan et al., 2004])). Input edges are a random subset of all atom pairs in the reference structure closer than 5 Å. The squared edge error (cf. SI Eq. 15) measures how well the input constraints are realized. It can be considered globally (averaged over all edges), locally (averaged

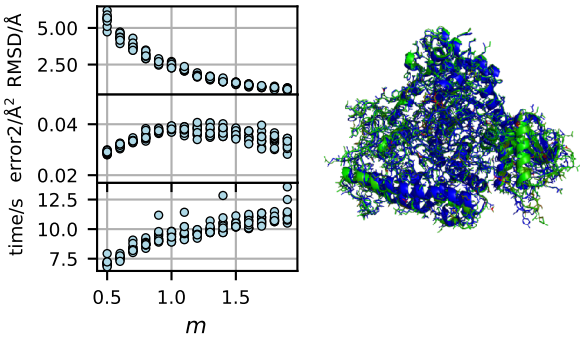

Figure 1: Reconstruction of protein structure for (PDB code `1mqq` 675 amino acids, [Golan et al., 2004]) based on an incomplete distance matrix. We use the protein sequence and randomly chosen additional inter-atomic distances (m is the number of additional edges per vertex). On the left from top to bottom: RMSD to blue reference structure, error2 (see SI Eq. 15) and runtime. For every number of edges used, results are shown for ten different sets of randomly chosen edges. The right hand side displays an exemplary output structure for $m = 1.5$ colored by squared local error from $0\,\text{Å}^2$ (green) to $0.05\,\text{Å}^2$ (red).

over one participating vertex), or individually. Since distance matrices are typically noisy or even erroneous, a structure generation engine has to be robust to such input. The supporting information contains more information on different systems, robustness and evaluation of the resulting model.

3 Conclusion

We present the first release of `diSTruct`, our python based tool to generate molecular structures from distance constraints. It provides a familiar interface to a generic algorithm and powerful analysis tools.

Acknowledgements

We would like to thank Michael Wegner for giving helpful advice.

Funding

This work was supported by the Helmholtz Analytics Framework of the Helmholtz Association under project number ZT-I-0003 and a Google Research Award. HM was partially supported by grant ME 3619/3-2 within German Research Foundation (DFG) Priority Programme 1736.

References

[Bruger, 2007] Bruger, A. (2007). Version 1.2 of the crystallography and nmr system. *Nature Protocols*, 2:2728–2733.

[Cock et al., 2009] Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M. J. L. (2009). Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423.

[Golan et al., 2004] Golan, G., Shallom, D., Teplitsky, A., Zaide, G., Shulami, S., Baasov, T., Stojanoff, V., Thomson, A., Shoham, Y., and Shoham, G. (2004). Crystal structures of geobacillus stearothermophilus alpha-glucuronidase complexed with its substrate and products: mechanistic implications. *The Journal of Biological Chemistry*, 297(4):3014–3024.

[Güntert and Buchner, 2015] Güntert, P. and Buchner, L. (2015). Combined automated noe assignment and structure calculation with cyana. *Journal of Biomolecular NMR*, 62(4):453–471.

[Hamelryck and Manderick, 2003] Hamelryck, T. and Manderick, B. (2003). Pdb file parser and structure class implemented in python. *Bioinformatics*, 19(17):2308–2310.

[Hu et al., 2013] Hu, Y., Gansner, E. R., and North, S. (2013). A maxent-stress model for graph layout. *IEEE Transactions on Visualization & Computer Graphics*, 19:927–940.

[Lieberman-Aiden et al., 2009] Lieberman-Aiden, E., Van Berkum, N. L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B. R., Sabo, P. J., Dorschner, M. O., et al. (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science*, 326(5950):289–293.

[Rieping et al., 2007] Rieping, W., Habeck, M., Bardiaux, B., Bernard, A., Malliavin, T. E., and Nilges, M. (2007). Aria2: Automated noe assignment and data integration in nmr structure calculation. *Bioinformatics*, 23(3):381–382.

[Schug et al., 2009] Schug, A., Weigt, M., Onuchic, J. N., Hwa, T., and Szurmant, H. (2009). High-resolution protein complexes from integrating genomic information with molecular simulation. *Proceedings of the National Academy of Sciences*, 106(52):22124–22129.

[Schwieters et al., 2006] Schwieters, C., Kuszewski, J., and Clore, G. (2006). Using xplor-nih for nmr molecular structure determination. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 48:47–62.

[Staudt et al., 2016] Staudt, C. L., Sazonovs, A., and Meyerhenke, H. (2016). Networkit: A tool suite for large-scale complex network analysis. *Network Science*, 4(4):508–530.

[Wegner et al., 2017] Wegner, M., Taubert, O., Schug, A., and Meyerhenke, H. (2017). Maxent-stress optimization of 3d biomolecular models. *25th Annual European Symposium on Algorithms*.