
Graphein - a Python Library for Geometric Deep Learning and Network Analysis on Protein Structures and Interaction Networks

Arian R. Jamasb^{1,2*}, Ramon Viñas², Eric J. Ma³, Charlie Harris⁴, Kexin Huang⁵,
Dominic Hall¹, Pietro Lió^{2*}, Tom L. Blundell¹

¹ Department of Biochemistry, University of Cambridge

² Department of Computer Science & Technology, University of Cambridge

³ PyMC Labs

⁴ Department of Life Sciences, Imperial College London

⁵ Department of Computer Science, Stanford University

Abstract

1 Geometric deep learning has well-motivated applications in the context of biology,
2 a domain where relational structure in datasets can be meaningfully leveraged.
3 Currently, efforts in both geometric deep learning and, more broadly, deep learning
4 applied to biomolecular tasks have been hampered by a scarcity of appropriate
5 datasets accessible to domain specialists and machine learning researchers
6 alike. However, there has been little exploration of how to best to integrate
7 and construct geometric representations of these datatypes. To address this, we
8 introduce Graphein as a turn-key tool for transforming raw data from widely-used
9 bioinformatics databases into machine learning-ready datasets in a high-throughput
10 and flexible manner. Graphein is a Python library for constructing graph and
11 surface-mesh representations of protein structures and biological interaction
12 networks for computational analysis. Graphein provides utilities for data retrieval
13 from widely-used bioinformatics databases for structural data, including the
14 Protein Data Bank, the recently-released AlphaFold Structure Database, and
15 for biomolecular interaction networks from STRINGdb, BioGrid, TRRUST
16 and RegNetwork. The library interfaces with popular geometric deep learning
17 libraries: DGL, PyTorch Geometric and PyTorch3D though remains framework
18 agnostic as it is built on top of the PyData ecosystem to enable inter-operability
19 with scientific computing tools and libraries. Graphein is designed to be highly
20 flexible, allowing the user to specify each step of the data preparation, scalable
21 to facilitate working with large protein complexes and interaction graphs, and
22 contains useful pre-processing tools for preparing experimental files. Graphein
23 facilitates network-based, graph-theoretic and topological analyses of structural
24 and interaction datasets in a high-throughput manner. As example workflows, we
25 make available two new protein structure-related datasets, previously unused by
26 the geometric deep learning community. We envision that Graphein will facilitate
27 developments in computational biology, graph representation learning and drug
28 discovery.

29

30

31

32

Availability and implementation: Graphein is written in Python. Source code,
example usage and tutorials, datasets, and documentation are made freely available
under the MIT License at the following URL: graphein.ai

*To whom correspondence should be addressed: arj39@cam.ac.uk, pl219@cl.cam.ac.uk

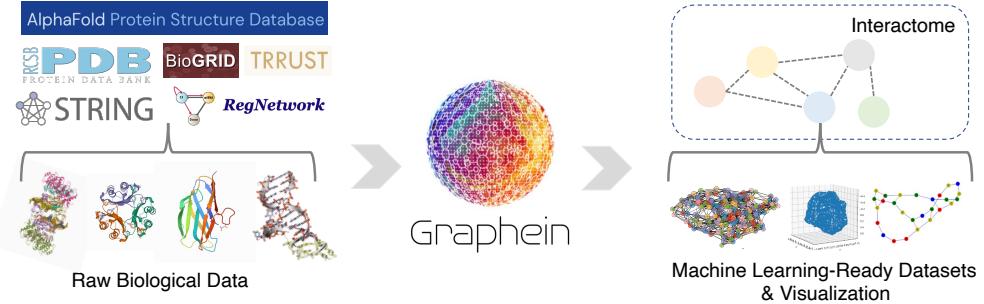


Figure 1: Graphein rapidly transforms and integrates raw biological data into actionable machine learning-ready datasets.

33 1 Introduction

34 Geometric deep learning refers to the application of deep learning methods to data with an underlying
35 non-Euclidean structure, such as graphs or manifolds [1]. These methods have already been applied
36 to a number of problems within computational biology and computational structural biology [2, 3,
37 4, 5, 6, 7, 8], and have shown great promise in the contexts of drug discovery and development [9].
38 Geometric deep learning libraries have been developed, providing graph representation functionality
39 and in-built datasets - typically with a focus on small molecules [10, 11]. Featurisation schemes
40 and computational analysis of small molecular graphs are a mature area of research. However, data
41 preparation for geometric deep learning in structural biology and interactomics is yet to receive the
42 same attention. Protein function is intricately tied to the underlying molecular structure which is
43 significantly more complex than small molecules. Protein graphs can be populated at different levels
44 of granularity, from atomic-scale graphs similar to small molecules, to residue-level graphs. The
45 relational structure of the data can be captured via spatial relationships or higher-order intramolecular
46 interactions which are not present in small molecule graphs. Furthermore, many biological functions
47 are mediated by interacting biomolecular entities, often through direct physical contacts governed by
48 their 3D structure. As a result, greater control over the data engineering and featurisation process of
49 structural data is required. Little attempt has been made to explore the effect of graph representations
50 of biological structures and to unify structural data and interaction data in the context of machine
51 learning. Graphein is a tool to address these issues by providing flexibility to researchers, decrease
52 the time required for data preparation and facilitate reproducible research.

53 Proteins form complex three dimensional structures to carry out cellular functions. Decades of
54 structural biology research and recent developments in protein folding have resulted in a large pool
55 of experimentally-determined and modelled protein structures with massive potential to inform
56 future research [12, 13]. However, it is not clear how best to represent these data in machine
57 learning experiments. 3D Convolutional Neural Networks (3DCNNs) have been routinely applied
58 to grid-structured representations of protein structures and sequence-based methods have proved
59 commonplace [14, 15, 16]. Nonetheless, these representations fail to capture relational information
60 in the context of intramolecular contacts and the internal chemistry of the biomolecular structures.
61 Furthermore, these methods are computationally inefficient due to convolving over large regions of
62 empty space, and computational constraints often require the volume of the protein considered to
63 regions of interest, thereby losing global structural information. For instance, in the case of protein-
64 ligand interaction and binding affinity prediction, central tasks in data-driven drug discovery, this often
65 takes the form of restricting the volume to be centred on a binding pocket, thereby losing information
66 about allosteric sites on the protein and possible conformational rearrangements that contribute
67 to molecular recognition. Furthermore, 3D volumetric representations are not translationally and
68 rotationally invariant, deficiencies that are often mitigated using costly data augmentation techniques.
69 Graphs suffer relatively less from these problems as they are translationally and rotationally invariant.
70 Structural descriptors of position can be leveraged and meaningfully exploited by architectures such
71 as Equivariant Neural Networks (ENN), which ensure geometric transformations applied to their
72 inputs correspond to well-defined transformations of the outputs.

73 Proteins and biological interaction networks can very naturally be represented as graphs at different
74 levels of granularity. Residue-level graphs represent protein structures as graphs where the nodes con-
75 sist of amino-acid residues and the edges the relations between them - often based on intramolecular
76 interactions or euclidean distance-based cutoffs. Atom-level graphs represent the protein structure in
77 a manner consistent with small-molecule graph representations, where nodes represent individual
78 atoms and edges the relations between them - often chemical bonds or, again, distance-based cutoffs.
79 The graph structure can further be elaborated by assigning numerical features to corresponding nodes
80 and edges as well as the whole graph. These features can represent, for instance, chemical properties
81 of the residue or atom-type, secondary structure assignments or solvent accessibility metrics of
82 the residue. Edge features can include bond or interaction types, or distances. Graph features can
83 include functional annotations or sequence-based descriptors. In the context of interaction networks,
84 structural data can be overlaid on protein nodes providing a multi-scale view of biological systems
85 and function. Graphein provides a bridge for geometric deep learning into structural interactomics.
86 Graph representations of proteins have a history of successful applications in machine learning
87 and structural analysis projects in structural biology [17, 18, 19]. Web-servers for computing
88 protein structure graphs exist [20, 21], however the lack of fine-grained control over the construction
89 and featurisation, public APIs for high-throughput programmatic access, ease of integrating data
90 modalities, and incompatibility with deep learning libraries motivated the development of Graphein.

91 2 Related Work

92 Geometric deep learning methods have demonstrated their suitability for tasks across domains. In
93 part, this has been fuelled by the development of libraries that provide easy access to non-Euclidean
94 data objects and models from the literature. Deep Graph Library (DGL) [10] and PyTorch Geometric
95 [11] are the main open-source frameworks built for PyTorch [22]. Other, less established, tools
96 include: Graph Nets [23] for Sonnet [24]/Tensorflow [25] and Jraph [26] for JAX [27].

97 In-built dataset support is a common feature of geometric deep learning frameworks. More specialised
98 libraries, such as DGL-LifeSci, DeepChem and TorchDrug, provide datasets, featurisation, neural
99 network layers and pre-trained models for tasks involving small molecules in the life sciences,
100 computational chemistry and drug development [28, 29, 30]. TorchDrug and DeepChem provide
101 reinforcement learning environments to fine tune generative models for physicochemical properties
102 such as drug-likeness (QED) and lipophilicity (LogP). Therapeutics Data Commons provides ML-
103 ready datasets for small molecule and biologics tasks but with no protein structural datasets [31].

104 Biomolecular tasks are included in many graph representation learning benchmarks. The Open
105 Graph Benchmark (OGB) includes graph property prediction tasks on small molecules, link pre-
106 diction tasks (ogbl-ppa) based on protein-protein interaction prediction and a biomedical knowl-
107 edge graph (ogbl-biokg), and a node classification task based on prediction of protein function
108 (ogbn-proteins) [32]. The TUDataset contains three biologically-motivated benchmark datasets
109 for graph classification, (PROTEINS, ENZYMES and DD) relevant to applications in structural biology
110 [33]. For PROTEINS and DD the goal is to predict whether or not a protein is an enzyme and these are
111 derived from the same data under differing graph construction schemes [34, 35]. ENZYMES provides
112 a task based on assigning Enzyme Commission (EC) numbers to graph representations of enzyme
113 structures derived from the BRENDA database [36]. More recently, ATOM3D provides a collection
114 of benchmark datasets for structurally-motivated tasks on biomolecules and show leveraging struc-
115 tural information consistently improves performance, and that the choice of architecture significantly
116 impacts performance depending on the context of the task [37].

117 Whilst tools exist for converting protein structures into graphs, they typically focus on visualisation
118 and leave much to be desired for deep learning practitioners. GraProStr is a web-server that enables
119 users to submit structures for conversion into a graph which can be downloaded as textfiles [38].
120 This provides users with limited control over the construction process, low-throughput and limited
121 featurisation support. Furthermore GraProStr provides no utilities for machine learning or unifying
122 structural and interactomic data o. Mayavi, and GSP4PDB & LIGPLOT provides utilities for
123 visualising protein structures and protein-ligand interaction as graphs, respectively. [39, 40, 41].
124 Bionoi is a library for representing protein-ligand interactions as voronoi diagram images specifically
125 for applications in machine learning [42],

126 The protein structure prediction model AlphaFold2 is perhaps the most promising example of
127 geometric deep learning applied to structural biology. Highly-accurate protein structure prediction
128 using AlphaFold2 has been applied at the proteome scale to humans and 20 key model organisms
129 [13, 43]. As a result of these developments, we anticipate a significant amount of growth in the
130 availability of protein structural and interaction data in the coming years. In particular, we identify
131 structural interactomics as an emerging area for geometric deep learning as sparse structural coverage
132 of the interactome can be infilled with modelled structures. The question of how to best leverage
133 and integrate these data with other modalities remains. A recent review of biomedical knowledge
134 graph datasets identifies graph composition, feature and metadata incorporation and reproducibility
135 as key challenges [44]. We have developed Graphein to address these issues and ensure these data are
136 accessible to computational scientists.

137 **3 Graphein**

138 Graphein provides utilities for constructing geometric representations of protein and RNA structures,
139 protein-protein interaction networks, and gene regulatory networks. The library is designed for both
140 novice and expert users through the use of a high or low-level API. The high-level API takes standard
141 biological identifiers and a configuration object as input to yield basic geometric representations of the
142 input data. The low-level API offers a detailed customisation of the graph selection from the input data,
143 allowing users to define their own data preparation, graph construction and featurisation functions in
144 a consistent manner. Graphein is built on the PyData Stack to allow for easy inter-operability with
145 standard scientific computing tools and deep learning framework agnosticism. Graphein is organised
146 into submodules for each of the modalities it supports (Figure 2).

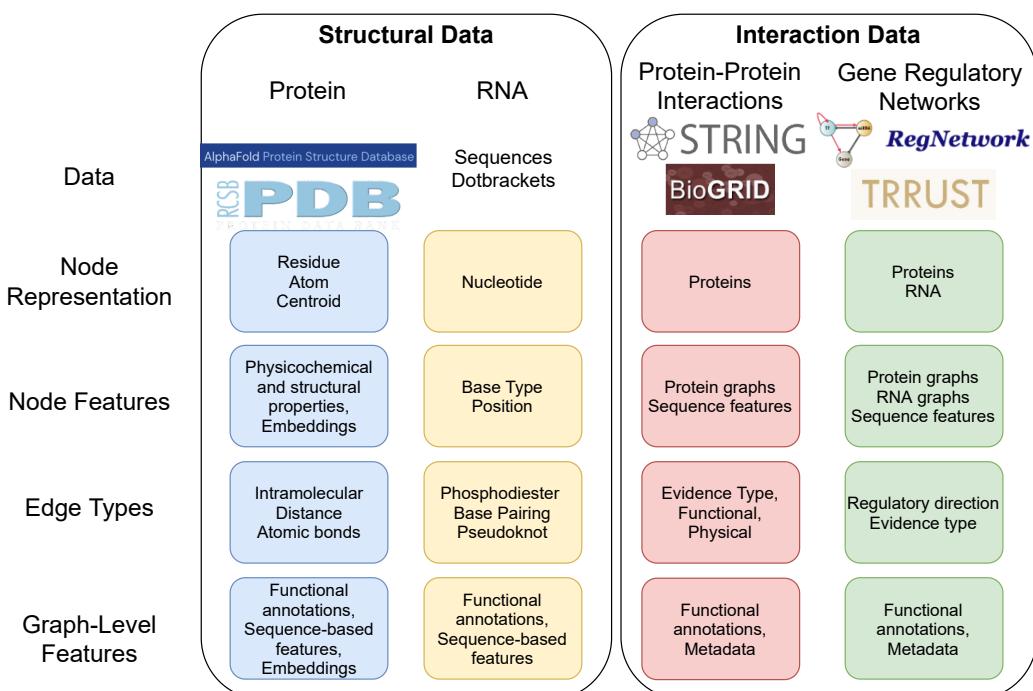


Figure 2: Overview of graph and mesh construction and featurisation schemes for data modalities supported by Graphein. Modules are inter-operable allowing protein or RNA structure graph construction to be applied to nodes in regulatory networks.

147 **3.1 Protein Structure Graphs**

148 Graphein interfaces with the PDB and the AlphaFold Structure Database to create geometric rep-
149 resentations of protein structures. Furthermore, users can supply their own .pdb files, enabling
150 pre-processing with standard bioinformatics tools and pipelines. An overview of featurisation
151 schemes is provided in Supplementary Information A.

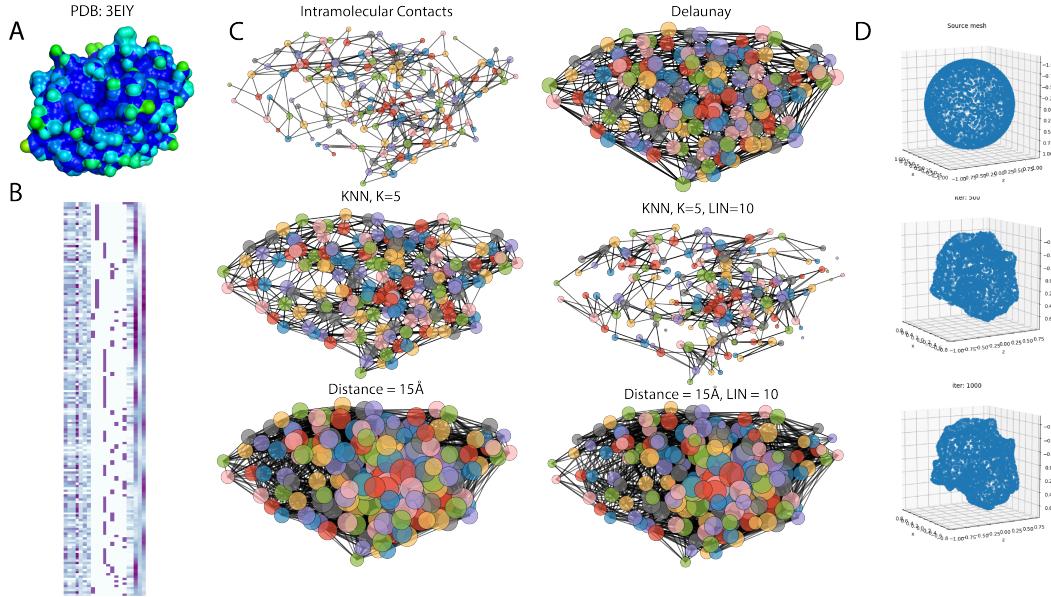


Figure 3: Example protein structure outputs from Graphein. **(A)** Example protein surface (3eiy). **(B)** Example node feature matrix for a residue-level graph. Features include low-dimensional amino acid embeddings, secondary structure assignments, solvent accessibility and x , y , z positions of nodes. **(C)** Graphs computed from α -carbons in the example protein under a variety of edge construction schemes. Node sizes correspond to degree. **(D)** Deformation of an icosphere to the example protein mesh constructed by Graphein in PyTorch3D

152 3.1.1 Node Representations

153 Graphs can be constructed for all chains contained within a polypeptide structure, or for a user-
154 defined selection of chains. This is useful in contexts where regions of interest on a protein may be
155 localised to a single chain. For residue-level graphs, users can choose between atom-based positional
156 information, or sidechain centroid. Sidechain centroids are calculated as the centre of gravity of the
157 deprotonated residue. Residue-level graph nodes can be featurised with a one-hot encoding of amino
158 acid type, physicochemical and biochemical properties retrieved from the ExPaSY ProtScale [45]
159 which includes 61 descriptors such as iso-electric points, mutability and transmembrane tendencies.
160 Additional numerical features can be retrieved from AAIndex [46]. Low-dimensional embeddings
161 of amino acid physicochemical properties are provided from Kidera et al. [47] and Meiler et al
162 [48]. In addition to fixed embeddings, sequence embeddings can be retrieved from large pre-trained
163 language models, such as the ESM-1b Transformer model [49] and BioVec [50]. Secondary structural
164 information can be included via a one-hot encoded representation of eight state secondary structure
165 and solvent accessibility metrics (ASA, RSA, SSA) computed by DSSP [51]. x , y , z positions are
166 added as node features. Functionality for user-defined node or edge features is also provided with
167 useful utilities allowing for computation or aggregation of features over constituent chains. Figure
168 2 illustrates an overview of the mesh and graph construction methods as well as the node and edge
169 featurisation schemes; Figure 3 shows example visualisations of graph and meshes produced by
170 Graphein.

171 3.1.2 Edge Representations

172 Graphein provides utilities in the high-level API for a number of edge-construction schemes. The
173 low-level API provides a simple and intuitive way for users to define novel edge construction schemes.
174 Edge construction methods are organised into distance-based, intramolecular interaction-based, and
175 atomic structure-based submodules. Each of these edge construction methods are composable to

176 produce multirelational graphs. This is particularly useful for models that operate on different levels to
177 capture varying aspects of the underlying network. As a motivating example, a multi-track approach
178 has been successfully applied to the protein folding problem [52].

179 Functionality for computing intramolecular graph edges is provided through distance-based heuristics
180 as well as through an optional dependency, GetContacts [53]. Euclidean distance-based edges can be
181 computed with a user-defined threshold. Functionality for constructing k -nearest neighbour graphs,
182 where two vertices are connected by an edge if they are among the k nearest neighbours by Euclidean
183 distance is included. Graph edges can also be added on the basis of the Delaunay triangulation.
184 Delaunay triangles correspond to joining points that share a face in the 3D Voronoi diagram of the
185 protein structures. For distance-based edges, a Long Interaction Network (LIN) parameter controls
186 the minimum required separation in the amino acid sequence for edge creation. This can be useful in
187 reducing the number of noisy edges under distance-based edge creation schemes. Edge featurisation
188 for atom-level graphs is provided by annotations of bond type and ring status.

189 3.2 Protein Structure Meshes

190 Geometric deep learning applied to surface representations of protein structures have demonstrated
191 promise on a variety of tasks in the context of structural biology and structural interactomics [8, 54].
192 The protein structure mesh module consists of a wrapper for PyMOL, a commonplace molecular
193 informatics visualisation tool, and Pytorch3D [55]. PyMol is used to produce a .Obj file from either a
194 PDB accession code or a provided .PDB file, enabling the use of pre-processed structures. Pytorch3D
195 is used to produce a tensor-based representation of the protein surface as vertices and faces. Users
196 can pass any desired parameters or commands controlling the surface calculation to PyMol via a
197 configuration object. These parameters include specifying solvent inclusion, solvent probe radius,
198 surface mode ({triangles, dots}), surface quality (resolution of mesh). We provide sane defaults for
199 first-time users. To our knowledge, this is the first application of PyTorch3D for protein structure
200 data.

201 3.3 RNA Structures

202 Ribonucleic Acid (RNA) is a nucleotide biopolymer capable of forming higher-order structural ar-
203 rangements through self-association mediated by complementary base pairing interactions. Graphein
204 provides utilities for constructing secondary structure graphs of RNA structures, taking as input an
205 RNA sequence and an associated string representation of the secondary structure in dotbracket nota-
206 tion [56]. Graphs can be constructed using two types of bonding between nucleotides: phosphodiester
207 bonds between adjacent bases, and base-pairing interactions between complementary bases specified
208 by the dotbracket string (Figure 4). Graphein also supports addition of pseudoknots - structural motifs
209 composed of interactions between intercalated hairpin loops specified in the dotbracket structure
210 notation.

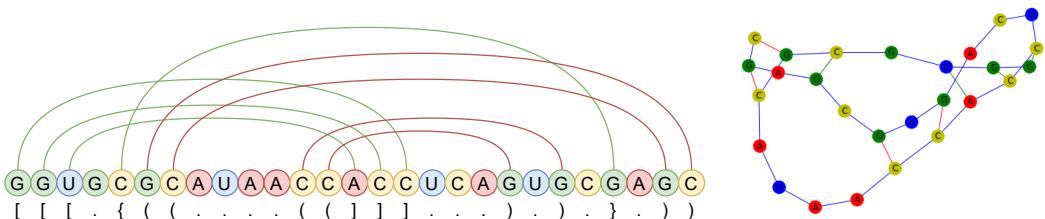


Figure 4: Example RNA Secondary Structure Graph. RNA Secondary structures can be represented as dotbracket strings and multi-relational graphs. Blue edges indicate phosphodiester backbone linkages, red edges indicate base-pairing interactions and green edges indicate pseudoknot pairings.

211 3.4 Interaction Networks

212 Interactomics presents a clear application of geometric deep learning as these data are fundamentally
213 relational in structure. Biomolecular entities can be represented as nodes, and their associated func-

214 tional relationships and physical interactions can be represented as edges with associated metadata,
215 such as the direction and nature of regulation. For a full discussion of applications, datasets and mod-
216 ellng techniques we refer readers to the reviews by [44] and [9]. Graphein implements interaction
217 graph construction from protein-protein interaction and gene regulatory network databases. Interac-
218 tion graphs integrate networks from several sources and can be constructed in a highly customisable
219 way (see Supplementary Information B, C for a summary of user-definable parameters).

220 **3.4.1 Protein-Protein Interaction Networks**

221 Many of the functional roles of proteins are carried out by larger assemblies of protein complexes
222 and many biological processes are regulated through interactions mediated by physical contacts.
223 Understanding these functions is central to characterising healthy and diseased states of biological
224 systems. Graphein interfaces with widely-used databases of biomolecular interaction data for easy
225 retrieval and construction of graph-based representations of protein-protein interactions.

226 **STRING** is a database of more than 20 billion known and predicted functional and direct physical
227 protein-protein interactions between 67.6 million proteins across 14,094 organisms [57]. Predicted
228 interactions in STRING are derived from genomic context, high-throughput experimental procedures,
229 conservation of co-expression, text-mining of the literature and aggregation from other databases.
230 STRING is made freely available by the original authors under a Creative Commons BY 4.0 license.

231 **BioGRID** is a database of 2,127,726 protein and genetic interactions curated from 77,696 publications
232 [58]. BioGRID is made available for academic and commercial use by the original authors under the
233 MIT License.

234 **3.4.2 Gene Regulatory Networks**

235 Gene regulatory networks (GRNs) consist of collections of
236 genes, transcription factors (TFs) and other regulatory elements,
237 and their associated regulatory interactions. Reconstructing
238 transcriptional regulatory networks is a long-standing problem
239 in computational biology in its own right due to its relevance
240 to characterising healthy and diseased states of cells, and these
241 data can provide meaningful signal in other contexts such as
242 multi-modal modelling of biological systems and phenomena.
243 Graphein supports GRN graph construction from two widely-
244 used databases, allowing users to easily unify datasets and
245 construct graph representations of these networks.

246 **TRRUST** is a database of regulatory interactions for human and
247 mouse interactomes curated from the literature via a sentence-
248 based text-mining approach [59]. The current release contains
249 8,427 / 6,490 regulatory interactions with associated regulatory
250 directions (activation/repression) over 795 / 827 transcription
251 factors and 2,067 / 1,629 non-transcription factor genes for
252 humans and mice, respectively. TRRUST is made freely available
253 by the original authors for non-commercial research under a
254 Creative Commons Attribution-ShareAlike 4.0 International
255 License.

256 **RegNetwork** is a database of transcription-factor and miRNA mediated regulatory interactions for
257 humans and mice [60]. RegNetwork is an aggregation of 25 source databases from which the
258 regulatory network is populated and annotated. The latest release contains 14,981 / 94,876 TF-gene,
259 361 / 129 TF-TF, 21,744 / 25,574 TF-miRNA, 171,477 / 176,512 miRNA-gene and 25,854 / 26,545
260 miRNA-TF interactions over 1,456 / 1,328 transcription factors, 1,904 / 1,290 miRNAs and 19,719
261 / 18,120 genes for humans and mice, respectively. The dataset is made publicly available by the
262 original authors.

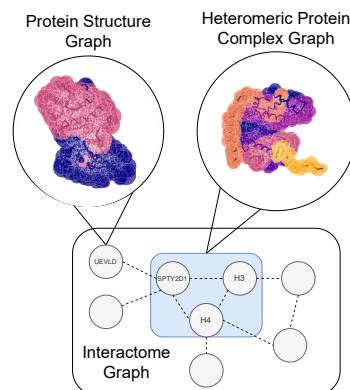


Figure 5: Graphein can facilitate the integration of structural and biomolecular interaction data to enable geometric deep learning research in structural interactomics. 3D visualisations of graphs are generated using Graphein.

263 4 Data

264 4.1 Database Interfaces

265 Graphein interfaces with a number of standard biological data repositories to retrieve data for each
266 modality it supports (summarised in Table 1).

Table 1: Graphein database interfaces for data retrieval.

Name	Description
Protein Structure	
Protein Data Bank (PDB)	Experimentally-determined biomolecular structures
AlphaFold Protein Structure Database	Protein structures modelled by AlphaFold2
Protein-Protein Interaction	
BioGrid	Protein-protein interactions
STRING	Protein-protein interactions
Gene Regulatory Network	
RegNetwork	TF & miRNA mediated regulatory interactions in humans and mice
TRRUST	Regulatory interactions in humans and mice.

267 4.2 Example Datasets

268 As example workflows, we make available two graph-based protein structure datasets focussed on
269 tasks where relational inductive biases appear intuitively useful and demonstrate how Graphein can
270 help formulate different tasks from the same underlying dataset.

271 **PPISP - Protein Protein Interaction Site Prediction** The first, based on the collections outlined
272 in [61], consists of 420 protein structures, with node labels indicating whether a residue is involved
273 in a protein-protein interaction - a task central to structural interactomics [62]. The data originate
274 from co-crystallised structures of the complexes in the RCSB PDB. The authors make available
275 a set of additional node features based on Position-Specific Scoring Matrices (PSSMs), providing
276 evolutionary context as to which protein-protein interaction sites are typically conserved, which can
277 be incorporated with the structural node features calculated by Graphein. This dataset was used in
278 [63] in conjunction with Graphein to compute the protein structure graph inputs to a Message-Passing
279 Neural Process model which achieved state-of-the-art performance.

280 **PSCDB - Protein Structural Change Database** The second dataset, based on Protein Structural
281 Change Database (PSCDB) [64], consists of 904 paired examples of bound and unbound protein struc-
282 tures that undergo 7 classes of conformational rearrangement motion. Prediction of conformational
283 rearrangement upon ligand binding is a longstanding problem in computational structural biology
284 and has significant implications for drug discovery and development. Two tasks can be formulated
285 with this dataset. The first is the graph classification task of predicting the type of motion a protein
286 undergoes upon ligand binding, the second is framing prediction of the rearrangement itself as an
287 edge prediction task between the paired bound and unbound protein structure graphs. These tasks
288 provide utility in improving understanding of protein structural dynamics in drug development, where
289 molecules are typically docked into largely rigid structures with limited flexibility in the binding
290 pockets in high-throughput *in silico* screens. PSCDB is made publicly available by the original
291 authors and we provide a processed version in our repository.

292 **ccPDB** We derive four datasets, each with a graph and a node classification task from the ccPDB
293 [65]. The ccPDB provides collections of protein structures and annotations of interactions with
294 various molecular species. The proteins are high-quality, non-redundant sets (25% sequence identity)
295 with maximum resolution of 3 Å, minimum sequence length of 80 residues. Node-level annotations
296 of interaction are provided in each case with the cutoff set at 4 Å. ccPDB is made freely available
297 online.

- 298 • PROTEINS_METAL contains protein structures that bind 7 types of metal ions (Fe, Mg, Ca,
299 Mn, Zn, Co, Ni; $n = 215 / 1,908 / 1,402 / 521 / 1,660 / 201 / 355$).

- 300 • PROTEINS_NUCLEOTIDES contains protein structures that bind 8 species of nucleotides
301 (ATP, ADP, GTP, GDP, NAD, FAD, FMN, UDP; $n = 313 / 353 / 83 / 120 / 140 / 172 / 117 /$
302 68)
303 • PROTEINS_NUCLEIC contains protein structures that bind DNA or RNA polymers ($n = 560$
304 / 415).
305 • PROTEINS_LIGAND contains protein structures that bind 7 species of ligands (SO₄, PO₄,
306 NAG, HEM, BME, EDO, PLP; $n = 3312 / 1299 / 727 / 176 / 191 / 1507 / 65$).

307

5 Benchmarks

308 To demonstrate the ease-of-use of Graphein we apply a selection of geometric deep learning models
309 to protein structure graphs generated by Graphein to these datasets. In particular, we consider
310 two graph construction schemes from the same dataset: one based on hydrogen and peptide bonds
311 (Bond), and another based on K-Nearest Neighbours clustering of the residues (KNN). Full details
312 of our experimental procedure are provided in Supplementary Information D. We consider a graph
313 classification task on PSCDB, classifying structures on the basis of the type of structural rearrangement
314 they undergo upon ligand binding. Our results show strong differences in performance between the
315 two schemes and that these differences are architecture-dependent (Table 2). We suggest that this
316 motivates further exploration of the role of graph construction in the context of structural biological
317 data.

Table 2: Baseline results for PSCDB. The task is graph classification to predict the class of structural
rearrangement a given protein undergoes upon ligand binding.

Edge Type	Bond		KNN		
	Model	Multi-ACC	Macro-F1	Multi-ACC	Macro-F1
GCN	0.221±0.009	0.110±0.062	0.261±0.111	0.154±0.084	
GraphSAGE	0.170±0.091	0.078±0.046	0.247±0.133	0.136±0.090	
GAT	0.241±0.066	0.145±0.037	0.258±0.136	0.154±0.105	

318

6 Machine Learning Utilities

319 **Conversion** Convenience utilities for converting between NetworkX [66] graph objects and
320 commonly-used geometric deep learning library data objects are provided for DGL and PyTorch
321 Geometric. Underlying graph objects are based on NetworkX, enabling conversion to other formats.

322 **Adjacency Tensors & Diffusion Matrices** Graphein provides utilities for computation of diffusion
323 matrices (and related adjacency matrices) to (1) facilitate exploration of biological data with models
324 that leverage these representations, and (2) aid in the construction of diffusion matrices for graph
325 neural networks.

326 **Visualisation** Built-in tools are provided for each of the modalities supported to allow inspection of
327 data in pre and post-processing. Interactive visualisation tools are provided for protein structures.

328

7 Usage

329 Example usage and workflows are provided in the documentation at: www.graphein.ai. Examples
330 and tutorials are provided as runnable notebooks detailing use of the high and low-level APIs for the
331 data modalities currently supported by Graphein, and the ease of ingesting novel structural datasets
332 into a suite of geometric deep learning benchmarks (see section 5). Source code is made available via
333 GitHub: [www.github.com/a-r-j/graphein](https://github.com/a-r-j/graphein).

334

8 Conclusion

335 Geometric deep learning has shown promise in computational biology and structural biology. How-
336 ever, the availability of processed datasets is a research bottleneck. Graphein is a Python library

337 designed to facilitate construction of datasets for geometric deep learning applied to biomolecular
338 structures and interactions. By providing tools for these modalities, we hope to facilitate research in
339 data-driven structural interactomics. In addition, we make available two datasets for protein-protein
340 interaction site prediction (node classification) and protein conformational rearrangement prediction
341 (graph classification and edge prediction).
342 A current limitation of the library is the lack of support for some informative features based on
343 evolutionary information. For example, the PPISP dataset provides PSSMs and the protein folding
344 model, AlphaFold2 is heavily reliant on Multiple Sequence Alignments presenting clear utility for
345 the addition of these features. Whilst graphs are a natural representation of biological interaction data,
346 hypergraphs may provide a higher-fidelity representation of the underlying biological relationships.
347 Many interactions are contextual, which can be represented by hyperedges between several entities
348 required for a functional or structural relationship. We are also interested in addressing representations
349 of dynamics, both in structural data and in interactions as these are central biological components that
350 are beyond the scope of the initial release. These features will be included in subsequent releases and
351 the API design of Graphein makes it simple for users to write and contribute their own workflows.
352 Graphein implements a high-level and low-level API to enable rapid and fine-grained control of data
353 preparation. Graphein is provided as Free Open Source Software under a permissive MIT License
354 which we hope will encourage the community to contribute customised workflows to the library. We
355 hope that Graphein serves to further progress in the field and reduce friction in processing structural
356 and interaction data for geometric deep learning. The library also provides utility in preparing protein
357 structure and interactomics graphs for graph-theoretic and topological data analyses.

358 Acknowledgments and Disclosure of Funding

359 We thank Ben Day, Paul Scherer and Cristian Bodnar for useful feedback on the manuscript and
360 project. We thank Sean Aubin for contributing the interactive visualisation code for protein structure
361 graphs. We thank our users for bug reports, feature suggestions and their support. ARJ is funded by a
362 BBSRC DTP studentship. DH has received funding from both the Wellcome Trust and the United
363 Kingdom Medical Research Council (MRC). TLB thanks the Wellcome Trust for an Investigator
364 Award (200814/Z/16/Z; 2016 -) for support of this research.

365 References

- 366 [1] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst.
367 Geometric deep learning: going beyond euclidean data. 2016. doi: 10.1109/MSP.2017.2693418.
- 368 [2] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using
369 graph convolutional networks. In *Proceedings of the 31st International Conference on Neural
370 Information Processing Systems*, NIPS'17, page 6533–6542, Red Hook, NY, USA, 2017. Curran
371 Associates Inc. ISBN 9781510860964.
- 372 [3] Wen Torng and Russ B. Altman. Graph convolutional neural networks for predicting drug-target
373 interactions. *Journal of Chemical Information and Modeling*, 59(10):4131–4149, October 2019.
374 doi: 10.1021/acs.jcim.9b00628. URL <https://doi.org/10.1021/acs.jcim.9b00628>.
- 375 [4] Vladimir Gligorijevic, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman,
376 Kyunghyun Cho, Tommi Vatanen, Daniel Berenberg, Bryn Taylor, Ian M. Fisk, Ram-
377 nik J. Xavier, Rob Knight, and Richard Bonneau. Structure-based function prediction us-
378 ing graph convolutional networks. October 2019. doi: 10.1101/786236. URL <https://doi.org/10.1101/786236>.
- 380 [5] Michael Uhl, Van Dinh Tran, and Rolf Backofen. GraphProt2: A novel deep learning-based
381 method for predicting binding sites of RNA-binding proteins. November 2019. doi: 10.1101/
382 850024. URL <https://doi.org/10.1101/850024>.
- 383 [6] Rafael Zamora-Resendiz and Silvia Crivelli. Structural learning of proteins using graph con-
384 volutional neural networks. April 2019. doi: 10.1101/610444. URL <https://doi.org/10.1101/610444>.
- 386 [7] Soumya Sanyal, Ivan Anishchenko, Anirudh Dagar, David Baker, and Partha Talukdar. Prote-
387 inGCN: Protein model quality assessment using graph convolutional networks. April 2020. doi:
388 10.1101/2020.04.06.028266. URL <https://doi.org/10.1101/2020.04.06.028266>.
- 389 [8] P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, and B. E. Correia.
390 Deciphering interaction fingerprints from protein molecular surfaces using geometric deep
391 learning. *Nature Methods*, December 2019. doi: 10.1038/s41592-019-0666-6. URL <https://doi.org/10.1038/s41592-019-0666-6>.
- 393 [9] Thomas Gaudelot, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu,
394 Jeremy B R Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell,
395 Michael M Bronstein, and Jake P Taylor-King. Utilizing graph machine learning within drug
396 discovery and development. *Briefings in Bioinformatics*, 05 2021. ISSN 1477-4054. doi:
397 10.1093/bib/bbab159. URL <https://doi.org/10.1093/bib/bbab159>. bbab159.
- 398 [10] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou,
399 Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang
400 Li, Alexander J Smola, and Zheng Zhang. Deep graph library: Towards efficient and scalable
401 deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*,
402 2019. URL <https://arxiv.org/abs/1909.01315>.
- 403 [11] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric.
404 In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

- 405 [12] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig,
406 Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28
407 (1):235–242, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL <https://doi.org/10.1093/nar/28.1.235>.
- 409 [13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ron-
410 neberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex
411 Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino
412 Romera-Paredes, Stanislav Nikolov, Rishabh Jain, Jonas Adler, Trevor Back, Stig Petersen,
413 David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas
414 Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Ko-
415 ray Kayukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure
416 prediction with AlphaFold. *Nature*, July 2021. doi: 10.1038/s41586-021-03819-2. URL
417 <https://doi.org/10.1038/s41586-021-03819-2>.
- 418 [14] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes.
419 Protein–ligand scoring with convolutional neural networks. *Journal of Chemical Infor-
420 mation and Modeling*, 57(4):942–957, April 2017. doi: 10.1021/acs.jcim.6b00740. URL
421 <https://doi.org/10.1021/acs.jcim.6b00740>.
- 422 [15] Rin Sato and Takashi Ishida. Protein model accuracy estimation based on local structure quality
423 assessment using 3d convolutional neural network. *PLOS ONE*, 14(9):e0221347, September
424 2019. doi: 10.1371/journal.pone.0221347. URL <https://doi.org/10.1371/journal.pone.0221347>.
- 426 [16] Limeng Pu, Rajiv Gandhi Govindaraj, Jeffrey Mitchell Lemoine, Hsiao-Chun Wu, and Michal
427 Brylinski. DeepDrug3d: Classification of ligand-binding pockets in proteins with a convolu-
428 tional neural network. *PLOS Computational Biology*, 15(2):e1006718, February 2019. doi: 10.
429 1371/journal.pcbi.1006718. URL <https://doi.org/10.1371/journal.pcbi.1006718>.
- 430 [17] Douglas EV Pires, Raquel C de Melo-Minardi, Marcos A dos Santos, Carlos H da Silveira,
431 Marcelo M Santoro, and Wagner Meira. Cutoff scanning matrix (CSM): structural classifica-
432 tion and function prediction by protein inter-residue distance patterns. *BMC Genomics*, 12(S4),
433 December 2011. doi: 10.1186/1471-2164-12-s4-s12. URL <https://doi.org/10.1186/1471-2164-12-s4-s12>.
- 435 [18] Douglas E. V. Pires, David B. Ascher, and Tom L. Blundell. mCSM: predicting the ef-
436 fects of mutations in proteins using graph-based signatures. *Bioinformatics*, 30(3):335–342,
437 November 2013. doi: 10.1093/bioinformatics/btt691. URL <https://doi.org/10.1093/bioinformatics/btt691>.
- 439 [19] Tammy M. K. Cheng, Yu-En Lu, Michele Vendruscolo, Pietro Lió, and Tom L. Blundell.
440 Prediction by graph theoretic measures of structural effects in proteins arising from non-
441 synonymous single nucleotide polymorphisms. *PLoS Computational Biology*, 4(7):e1000135,
442 July 2008. doi: 10.1371/journal.pcbi.1000135. URL <https://doi.org/10.1371/journal.pcbi.1000135>.
- 444 [20] Broto Chakrabarty, Varun Naganathan, Kanak Garg, Yash Agarwal, and Nita Parekh. NAPS
445 update: network analysis of molecular dynamics data and protein–nucleic acid complexes.
446 *Nucleic Acids Research*, 47(W1):W462–W470, May 2019. doi: 10.1093/nar/gkz399. URL
447 <https://doi.org/10.1093/nar/gkz399>.
- 448 [21] M. S. Vijayabaskar, V. Niranjan, and Saraswathi Vishveshwara. Graprostr - graphs of protein
449 structures: A tool for constructing the graphs and generating graph parameters for protein
450 structures. 2011.
- 451 [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan,
452 Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative
453 style, high-performance deep learning library. *Advances in neural information processing
454 systems*, 32:8026–8037, 2019.

- 455 [23] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zam-
456 baldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner,
457 Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani,
458 Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra,
459 Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational
460 inductive biases, deep learning, and graph networks, 2018.
- 461 [24] Deepmind. Sonnet. <https://github.com/deepmind/sonnet>, 2017.
- 462 [25] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu
463 Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for
464 large-scale machine learning. In *12th {USENIX} symposium on operating systems design and
465 implementation ({OSDI} 16)*, pages 265–283, 2016.
- 466 [26] Jonathan Godwin*, Thomas Keck*, Peter Battaglia, Victor Bapst, Thomas Kipf, Yujia Li,
467 Kimberly Stachenfeld, Petar Veličković, and Alvaro Sanchez-Gonzalez. Jraph: A library for
468 graph neural networks in jax., 2020. URL <http://github.com/deepmind/jraph>.
- 469 [27] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal
470 Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and
471 Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL
472 <http://github.com/google/jax>.
- 473 [28] Mufei Li, Jinjing Zhou, Jiajing Hu, Wenzuan Fan, Yangkang Zhang, Yixin Gu, and George
474 Karypis. Dgl-lifesci: An open-source toolkit for deep learning on graphs in life science, 2021.
- 475 [29] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin
476 Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. <https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837>.
- 477 [30] URL <https://torchdrug.ai/>.
- 478 [31] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W
479 Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine
480 learning datasets and tasks for drug discovery and development. *NeurIPS Track on Datasets
481 and Benchmarks*, 2021.
- 482 [32] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele
483 Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs,
484 2020.
- 485 [33] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion
486 Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML
487 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL
488 www.graphlearning.io.
- 489 [34] Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes
490 without alignments. *Journal of Molecular Biology*, 330(4):771–783, July 2003. doi: 10.1016/
491 s0022-2836(03)00628-4. URL [https://doi.org/10.1016/s0022-2836\(03\)00628-4](https://doi.org/10.1016/s0022-2836(03)00628-4).
- 492 [35] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M.
493 Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):
494 2539–2561, 2011. URL <http://jmlr.org/papers/v12/shervashidze11a.html>.
- 495 [36] I. Schomburg. BRENDa, the enzyme database: updates and major new developments. *Nucleic
496 Acids Research*, 32(90001):431D–433, January 2004. doi: 10.1093/nar/gkh081. URL <https://doi.org/10.1093/nar/gkh081>.
- 497 [37] Raphael J. L. Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers,
498 Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann,
499 Risi Kondor, Russ B. Altman, and Ron O. Dror. Atom3d: Tasks on molecules in three
500 dimensions, 2020.

- 503 [38] M.S. Vijayabaskar. GraProStr – graphs of protein structures: A tool for constructing the
504 graphs and generating graph parameters for protein structures. *The Open Bioinformatics*
505 *Journal*, 5(1):53–58, February 2011. doi: 10.2174/1875036201105010053. URL <https://doi.org/10.2174/1875036201105010053>.
- 507 [39] Prabhu Ramachandran and Gael Varoquaux. Mayavi: 3d visualization of scientific data.
508 *Computing in Science & Engineering*, 13(2):40–51, March 2011. doi: 10.1109/mcse.2011.35.
509 URL <https://doi.org/10.1109/mcse.2011.35>.
- 510 [40] Renzo Angles, Mauricio Arenas-Salinas, Roberto García, Jose Antonio Reyes-Suarez, and
511 Ehmke Pohl. GSP4pdb: a web tool to visualize, search and explore protein-ligand structural
512 patterns. *BMC Bioinformatics*, 21(S2), March 2020. doi: 10.1186/s12859-020-3352-x. URL
513 <https://doi.org/10.1186/s12859-020-3352-x>.
- 514 [41] Roman A. Laskowski and Mark B. Swindells. LigPlot: Multiple ligand–protein interaction
515 diagrams for drug discovery. *Journal of Chemical Information and Modeling*, 51(10):2778–2786,
516 October 2011. doi: 10.1021/ci200227u. URL <https://doi.org/10.1021/ci200227u>.
- 517 [42] Joseph Feinstein, Wentao Shi, J. Ramanujam, and Michal Brylinski. Bionoi: A voronoi
518 diagram-based representation of ligand-binding sites in proteins for machine learning applica-
519 tions. In *Methods in Molecular Biology*, pages 299–312. Springer US, 2021. doi: 10.1007/978-1-0716-1209-5_17. URL https://doi.org/10.1007/978-1-0716-1209-5_17.
- 521 [43] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Au-
522 gustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer
523 Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael
524 Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J.
525 Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishabh Jain, Ellen Clancy, David
526 Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet
527 Kohli, John Jumper, and Demis Hassabis. Highly accurate protein structure prediction
528 for the human proteome. *Nature*, July 2021. doi: 10.1038/s41586-021-03828-1. URL
529 <https://doi.org/10.1038/s41586-021-03828-1>.
- 530 [44] Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Andreas Bender,
531 Charles Tapley Hoyt, and William Hamilton. A review of biomedical datasets relating to drug
532 discovery: A knowledge graph perspective, 2021.
- 533 [45] Elisabeth Gasteiger, Christine Hoogland, Alexandre Gattiker, Severine Duvaud, Marc R.
534 Wilkins, Ron D. Appel, and Amos Bairoch. Protein identification and analysis tools on the Ex-
535 PASy server. In *The Proteomics Protocols Handbook*, pages 571–607. Humana Press, 2005. doi:
536 10.1385/1-59259-890-0:571. URL <https://doi.org/10.1385/1-59259-890-0:571>.
- 537 [46] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa.
538 AAindex: amino acid index database, progress report 2008. *Nucleic Acids Research*, 36
539 (Database):D202–D205, December 2007. doi: 10.1093/nar/gkm998. URL <https://doi.org/10.1093/nar/gkm998>.
- 541 [47] Akinori Kidera, Yasuo Konishi, Masahito Oka, Tatsuo Ooi, and Harold A. Scheraga. Statistical
542 analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of*
543 *Protein Chemistry*, 4(1):23–55, February 1985. doi: 10.1007/bf01025492. URL <https://doi.org/10.1007/bf01025492>.
- 545 [48] Jens Meiler, Anita Zeidler, Felix Schmuschke, and Michael Muller. Generation and evaluation of
546 dimension-reduced amino acid parameter representations by artificial neural networks. *Journal*
547 *of Molecular Modeling*, 7(9):360–369, September 2001. doi: 10.1007/s008940100038. URL
548 <https://doi.org/10.1007/s008940100038>.
- 549 [49] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo,
550 Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function
551 emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the*
552 *National Academy of Sciences*, 118(15), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2016239118.
553 URL <https://www.pnas.org/content/118/15/e2016239118>.

- 554 [50] Ehsaneddin Asgari and Mohammad R. K. Mofrad. Continuous distributed representation
555 of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287,
556 November 2015. doi: 10.1371/journal.pone.0141287. URL <https://doi.org/10.1371/journal.pone.0141287>.
- 558 [51] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: Pattern recog-
559 nition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, Decem-
560 ber 1983. doi: 10.1002/bip.360221211. URL <https://doi.org/10.1002/bip.360221211>.
- 561 [52] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov,
562 Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán,
563 Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira,
564 Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo
565 Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi,
566 Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J.
567 Read, and David Baker. Accurate prediction of protein structures and interactions using a three-
568 track neural network. *Science*, 373(6557):871–876, July 2021. doi: 10.1126/science.abj8754.
569 URL <https://doi.org/10.1126/science.abj8754>.
- 570 [53] GetContacts. Getcontacts. URL <https://getcontacts.github.io/>.
- 571 [54] Bowen Dai and Chris Bailey-Kellogg. Protein interaction interface region prediction by geo-
572 metric deep learning. *Bioinformatics*, March 2021. doi: 10.1093/bioinformatics/btab154. URL
573 <https://doi.org/10.1093/bioinformatics/btab154>.
- 574 [55] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.
- 575 [56] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast
576 folding and comparison of RNA secondary structures. *Monatshefte fr Chemie Chemical
577 Monthly*, 125(2):167–188, February 1994. doi: 10.1007/bf00818163. URL <https://doi.org/10.1007/bf00818163>.
- 578 [57] Damian Szkłarczyk, Annika L Gable, Katerina C Nastou, David Lyon, Rebecca Kirsch, Sampo
579 Pyysalo, Nadezhda T Doncheva, Marc Legeay, Tao Fang, Peer Bork, Lars J Jensen, and Christian
580 von Mering. The STRING database in 2021: customizable protein–protein networks, and
581 functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Research*,
582 49(D1):D605–D612, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1074. URL <https://doi.org/10.1093/nar/gkaa1074>.
- 583 [58] Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew
584 Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, Sonam Dolma, Jasmin
585 Coulombe-Huntington, Andrew Chatr-aryamontri, Kara Dolinski, and Mike Tyers. The biogrid
586 database: A comprehensive biomedical resource of curated protein, genetic, and chemical
587 interactions. *Protein Science*, 30(1):187–200, 2021. doi: <https://doi.org/10.1002/pro.3978>. URL
588 <https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.3978>.
- 589 [59] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo
590 Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, Sungho Lee, Byunghee Kang, Dabin
591 Jeong, Yaeji Kim, Hyeon-Nae Jeon, Haein Jung, Sunhwee Nam, Michael Chung, Jong-Hoon
592 Kim, and Insuk Lee. TRRUST v2: an expanded reference database of human and mouse tran-
593 scriptional regulatory interactions. *Nucleic Acids Research*, 46(D1):D380–D386, 10 2017. ISSN
594 0305-1048. doi: 10.1093/nar/gkx1013. URL <https://doi.org/10.1093/nar/gkx1013>.
- 595 [60] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. RegNetwork: an integrated database
596 of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*,
597 2015, 09 2015. ISSN 1758-0463. doi: 10.1093/database/bav095. URL <https://doi.org/10.1093/database/bav095>. bav095.
- 598 [61] Min Zeng, Fuhao Zhang, Fang-Xiang Wu, Yaohang Li, Jianxin Wang, and Min Li. Protein-
599 protein interaction site prediction through combining local and global features with deep
600 neural networks. *Bioinformatics*, September 2019. doi: 10.1093/bioinformatics/btz699. URL
601 <https://doi.org/10.1093/bioinformatics/btz699>.

- 605 [62] Arian R. Jamasb, Ben Day, Cătălina Cangea, Pietro Liò, and Tom L. Blundell. *Deep learning*
606 *for Protein–Protein Interaction Site Prediction*, pages 263–288. Springer US, New York,
607 NY, 2021. ISBN 978-1-0716-1641-3. doi: 10.1007/978-1-0716-1641-3_16. URL https://doi.org/10.1007/978-1-0716-1641-3_16.
- 609 [63] Ben Day, Cătălina Cangea, Arian R. Jamasb, and Pietro Liò. Message passing neural processes,
610 2020.
- 611 [64] T. Amemiya, R. Koike, A. Kidera, and M. Ota. PSCDB: a database for protein structural
612 change upon ligand binding. *Nucleic Acids Research*, 40(D1):D554–D558, November 2011.
613 doi: 10.1093/nar/gkr966. URL <https://doi.org/10.1093/nar/gkr966>.
- 614 [65] Piyush Agrawal, Sumeet Patiyal, Rajesh Kumar, Vinod Kumar, Harinder Singh, Pawan Kumar
615 Raghav, and Gajendra P S Raghava. ccPDB 2.0: an updated version of datasets created
616 and compiled from Protein Data Bank. *Database*, 2019, 01 2019. ISSN 1758-0463. doi:
617 10.1093/database/bay142. URL <https://doi.org/10.1093/database/bay142>. bay142.
- 618 [66] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics,
619 and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors,
620 *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

621 Checklist

- 622 1. For all authors...
 - 623 (a) Do the main claims made in the abstract and introduction accurately reflect the paper's
624 contributions and scope? **[Yes]**
 - 625 (b) Did you describe the limitations of your work? **[Yes] Yes, please see the conclusion**
(Section 8).
 - 627 (c) Did you discuss any potential negative societal impacts of your work? **[Yes] Please**
628 **see Supplementary Information E**
 - 629 (d) Have you read the ethics review guidelines and ensured that your paper conforms to
630 them? **[Yes] We have read the ethics review guidelines.**
- 631 2. If you are including theoretical results...
 - 632 (a) Did you state the full set of assumptions of all theoretical results? **[N/A] We are not**
including theoretical results.
 - 634 (b) Did you include complete proofs of all theoretical results? **[N/A] We are not including**
theoretical results.
- 636 3. If you ran experiments (e.g. for benchmarks)...
 - 637 (a) Did you include the code, data, and instructions needed to reproduce the main ex-
638 perimental results (either in the supplemental material or as a URL)? **[Yes] Please**
639 **see the repository: <https://www.github.com/a-r-j/graphein> and documen-**
640 **tation: www.graphein.ai**
 - 641 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they
642 were chosen)? **[Yes] A full description is given in Supplementary Information D**
 - 643 (c) Did you report error bars (e.g., with respect to the random seed after running experi-
644 ments multiple times)? **[Yes] Please see Table 2.**
 - 645 (d) Did you include the total amount of compute and the type of resources used (e.g.,
646 type of GPUs, internal cluster, or cloud provider)? **[Yes] Please see Supplementary**
Information D.
- 648 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - 649 (a) If your work uses existing assets, did you cite the creators? **[Yes] We have cited the**
650 **appropriate creators in the text.**
 - 651 (b) Did you mention the license of the assets? **[Yes] We have described the licensing**
652 **conditions in the text.**
 - 653 (c) Did you include any new assets either in the supplemental material or as a URL?
654 **[Yes] Please see our repository: <https://www.github.com/a-r-j/graphein>**
655 **and documentation: graphein.ai**
 - 656 (d) Did you discuss whether and how consent was obtained from people whose data you're
657 using/curating? **[Yes] We use publicly available data and specify the original**
658 **licenses.**
 - 659 (e) Did you discuss whether the data you are using/curating contains personally identifiable
660 information or offensive content? **[N/A] No datasets include personally identifiable**
661 **information or offensive content.**
- 662 5. If you used crowdsourcing or conducted research with human subjects...
 - 663 (a) Did you include the full text of instructions given to participants and screenshots, if
664 applicable? **[N/A] We do not use crowdsourcing or human subjects.**
 - 665 (b) Did you describe any potential participant risks, with links to Institutional Review
666 Board (IRB) approvals, if applicable? **[N/A] We do not use crowdsourcing or**
667 **human subjects.**
 - 668 (c) Did you include the estimated hourly wage paid to participants and the total amount
669 spent on participant compensation? **[N/A] We do not use crowdsourcing or human**
670 **subjects.**