Universiteit Leiden
The Netherlands

Bioinformatics

Classifying Protein Topology Matrices Using AI

W.R.D. Venemans

Supervisors:
J.N. van Rijn (LIACS) & A. Mashaghi Tabari (LACDR)

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

19/8/2022

Abstract

Proteins form the building blocks of life and the human body and their folding are quintessential for their functioning within the cell. Categorizing the folding of proteins has been researched for a long time and to classify the topology a new theory was proposed: the circuit topology theory. This theory defines the way how the molecule folds by categorizing contacts in the chain into three types of relations: Parallel, shortened as P, Series, shortened as S and Cross, shortened as X. This theory also makes it possible to visualise the protein topology in matrix form, with the possibility of finding patterns in the folding of proteins. This thesis will assess the applicability of several artificial intelligence techniques to determine these patterns by establishing if two different AI models, the random forest and the automated image classifier, are capable of classifying real and fake topology matrices. Using real-world protein topology matrices and computer-generated protein topology matrices, this thesis analyzes the compatibility of the two models to classify the proteins and which strategy to pursue to build of computer-generated protein topology matrices to use in further studies. The results show that the random forest could distinguish nature-based protein matrices from fake randomly generated matrices in 91.30% of the cases whereas it could distinguish nature-based matrices from statistically assembled matrices in 99.63% of the cases. Furthermore, it was found that overall, the random forest scored better accuracy, precision, recall and F1-scores on both sets than the neural network.

Contents

Acknowledgements

I want to thank everyone for the tremendous amount of patience and help during the writing of this thesis, mainly Jan & Alireza and his team. Choppie!

1 Introduction

Proteins are one of the most important substances in life; they are essential in many processes that enable animals and plants to stay alive, grow and mate. However, to work properly, the proteins need to be folded into the right form and failing this will mean the protein will not work or in the worst case, will do their work faulty and thus cause discrepancies within the process [Mal08] [AJL+14]. The determination of the topology of the folding of proteins has been researched for a long time with many theories proposed over time [YNK07], but none of them fully satisfied all proteins, where not all proteins fold could be defined using the theory. Up until recently, the circuit topology theory was proposed, in which the chain folding has been divided into three different categories: cross, parallel and series [GM22] [Tez13] [Mas21]. In this theory, all protein folding is fully satisfied and this makes it possible to find the topological equivalence of different proteins, which could be potentially interesting for finding relatedness between different proteins. Using these three categories, it is possible to create a matrix for the topology of each protein, a unique feature within this theory, called a topology matrix. Both axes of the matrix represent points on the chain where the chain folds and is given one of these categories. [MvWT14].

The integration of computer science with biology, often grouped under the term *Bioinformatics* [Hog11], has shown a lot of promise within the fields of biology [HJS+21] and pharmacology [NYL+19]. One of the most promising elements are artificial intelligence models, such as decision trees which are used to recognize potential Micro-RNA-disease associations [CZY19] or clustering algorithms which are used to recognize patterns in gene expression [LYY05]. Recognizing patterns in the aforementioned topology matrices could potentially be fruitful for more research into the folding of proteins, as wrongly folded proteins are connected to pathogens and diseases [Mal08]. The folding of proteins can be used to discover new potential drug targets [Wis08] or can be used to bind to a receptor of choice [Sca]. To try to find these patterns, this paper aims to train multiple Artificial intelligence models to classify protein topology matrices as real or fake, where the fake matrices would represent wrongly folded or non-existent protein topologies, generated by another program. To state the research question more precisely: "Can artificial intelligence classify wrongly folded proteins based upon their topology matrix?"

The approach of this paper will be to create a pipeline to answer the research question. [1] The pipeline will start with the generation of data, where the real matrices will be extracted from the proteins of the RCSB database and the fake matrices will be generated using two different techniques. Secondly, the datasets will be divided into categories by size and used in the two artificial intelligence models: a random forest classifier, a decision tree-based model and an automated neural network, a self-building neural network jin2019auto. Both will be explained in more detail in Chapter 2. The results produced by the two models will be scored using different metrics, where the individual performance of the model and the capability of classifying the matrices are calculated. Using this data, we can answer the research question.

[1] The full code is available on github: https://git.liacs.nl/s2325217/ct-classifier

2 Background

This section will provide more information on the protein folding, protein topology, circuit topology and the artificial intelligence models used in this thesis.

2.1 Protein Folding

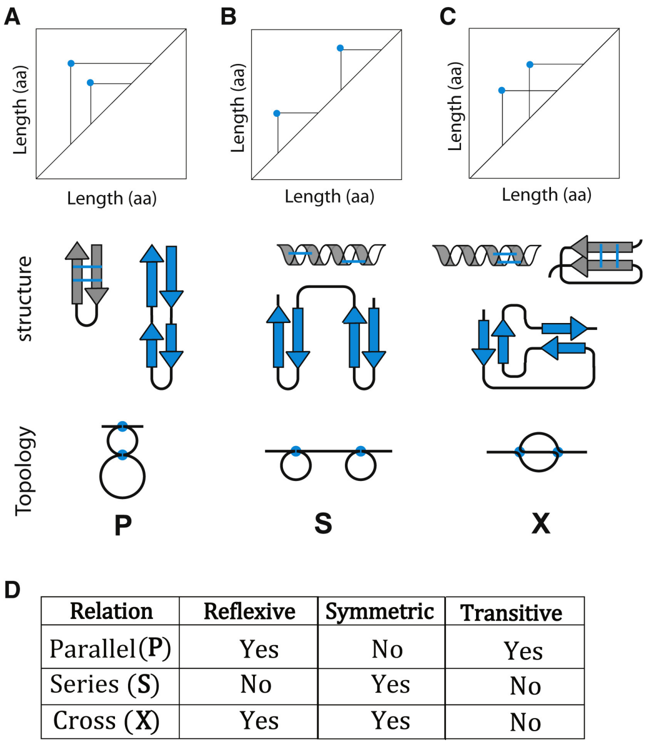
Proteins form the building blocks in life and the body, constituting most of a cell's dry mass. They execute a range of tasks in the cell, such as the pumping of enzymes through the cell or carrying messages through the cell. They are made of 20 amino acid types, that are distinct in physico-chemical properties including their size and charge. The folding of the amino acid chains and sometimes even with other amino acid chains is quintessential for the functioning of the protein: a wrongly folded protein can cause complete processes in the cell to fall apart or function erroneously. The complete and correct folding of a protein (complex) is called *the native conformation*. The amino acids are bound to each other by weak covalent bonds, such as hydrogen bonds, electrostatic attractions or van der Waals attractions. The conformation can be distinguished in 4 different levels of organisation: the primary structure, which describes the amino acids that can be found within the protein. Next is the secondary structure, which describes the local three-dimensional form of proteins, which mostly consists of α-helix and β-sheets. Thirdly, the tertiary structure describes the full three-dimensional shape of the amino acid chain. Lastly, only if the protein chain folds into a bigger protein complex, the quaternary structure describes this [AJL+14].

2.2 Protein Topology

Categorizing the folding of proteins has been researched for a long time, mainly starting with the interest in the three-dimensional shape of secondary structure of proteins and the conservation of it over all the years of evolution [MG95]. Over the years, three different methods of topology have already been researched: the branch topology [TO01], which describes topology under three classes: linear, branched and cyclic. However, this theory is not fit for protein folding as it puts all proteins into the same topological class [MvWT14]. Secondly, the knot topology proposes a theory where we can represent the topology using knots. This one is also not fit for protein folding as only less than 1% of the proteins in the online databases are knotted [SRM+12]. Thirdly, we have the network topology that describes the topology of proteins as a mathematical network, where statistical properties can be calculated. This also does not satisfy the protein topology, as it says nothing about the three-dimensional structure [Gol99].

2.3 Circuit Topology Theory

To classify the topology of proteins, a new topology method was proposed: the circuit topology theory [MvWT14] [Mas21]. This topology defines the way how the molecule fold by defining the aforementioned bonds between the DNA string and how they are positioned towards the other parts of the chain. Defining the bonds as binary contacts, three relations have been distinguished: Parallel, shortened as P, Series, shortened as S and Cross, shortened as X. Figure 1 illustrates these three categories: Panel A shows the Series relation; the blue arrows indicate contact between the chain and one of the contacts is in the loop of the other (this certain contact is defined as P^{-1} and the other contact as P) and such form a parallel connection. Panel B shows the series relation, where the blue contacts are in series with each other. A example is given in the middle of the panel, where the blue stripes in the DNA-strand represent the same contacts in the model below. Panel C shows the cross relation, which is shown in the first graph of the panel as two contacts crossing each other. These three relations all have different mathematical properties, as is shown in panel D, which can be used to prove and calculate the folding of every protein [MvWT14].



Relation	Reflexive	Symmetric	Transitive
Parallel (**P**)	Yes	No	Yes
Series (**S**)	No	Yes	No
Cross (**X**)	Yes	Yes	No

Figure 1: The three categories, including the structures and reflexive, symmetrical and transitive relations. (Figure taken from [MvWT14].)

Using these three categories it is possible to build a topology matrix of the folding of any protein, an unique property of the circuit topology theory. Figure 2 shows a simple example, using the three categories that were explained earlier. These matrices will be used in this thesis and will be more thoroughly explained in Section 3.

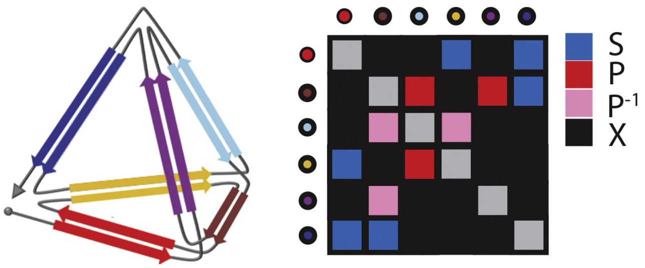


Figure 2: A simple example of a topology matrix. (Figure taken from [MvWT14].)

2.4 Machine learning

It has been theorized that in the future, machine learning and biology could enhance biological research more and more [HJS+21]. This thesis aims to make one of these steps by making use of two artificial intelligence models to classify the difference between fake and real protein topology matrices. However, to make use of the models, we have to make sure that the topology matrices or instances as they are called, are in the right (data)structure that is flexible and space efficient. The matrices can be presented in many forms but the format that stands out is the matrix or the *dataframe* format, as it is called in the python library *pandas* [RjM+20] [WM10]. This same library offers time and space efficient ways to import, export and edit the matrices. Furthermore, the python library *Numpy* [HMvdW+20] shall be used for a lot of editing as well, as it offers a lot of flexibility together with *pandas'* dataframes. The models will also need target values, which are the labels of the matrices if they are fake or real, as the task of the models will be to separate real topology matrices from real matrices,
Both the models that will be used in this thesis fall under the category of supervised learning, which means that the model first receives a *training set*, a set of matrices including the label real or fake. Using this set, the model will train itself by finding certain features within the matrices that match with the real or fake matrices. After the set has been trained, the model will receive a test set, a set of matrices without the labels real or fake. The model will try, using the training it got from the training set, to classify each of these matrices in the test set. This result of labels will then be calculated to be interpreted by humans and metrics that will be discussed in Section 3 [Mo111]. The next two paragraphs will explain the two different models that will be used in this thesis.

2.4.1 Random Forest Classifier

The random forest classifier is one of the oldest models in computer science [Bre01]. The random forest classifier consists of multiple decision trees, where the most voted label of all the trees becomes the final prediction. A decision tree is like a flow-chart, the tree starts at the root and a base test. Based on the outcome of the test, we move to a lower node in the tree. After the root, we will continue into the tree; All the nodes inside the tree also contain these tests, based on the same or other features. The decision tree ends in a leaf node, where the tree outputs a label. The random forest classifier creates multiple of these trees and the most selected outcome of the trees will be the output of the classifier (see Figure 3).

2.4.2 Neural Network

The other model that is used in this thesis is an automated image classifier from Autokeras [JSH19]. This classifier is based upon the neural network model, which is inspired by the neural network design of our brains and is made out of 3 parts: the input layer, the hidden layer(s) and the output layer. The input layer takes in the matrix and the predicted output is the label *True* or *False*. The most complex part of designing the neural network is constructing the hidden layer, which could be multiple layers. Every layer contains nodes, which all have (multiple) in- and outputs, which all have weights for the model to determine how important a certain feature is. Inside the nodes, an operation or test is being done on the inputs and is after that the output flows into the next layer. These layers can have varying amounts of nodes and input and outputs, which makes this model extra flexible. Automated machine learning offers the building and tuning of these layers with little

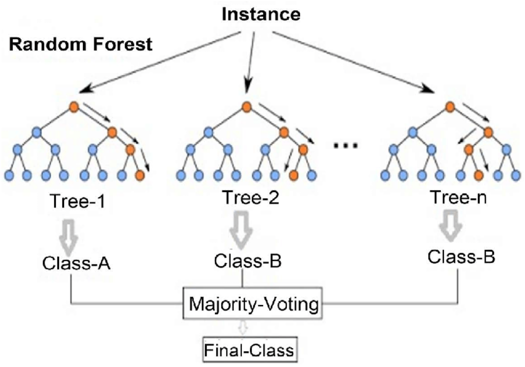


Figure 3: An example random forest classifier with 3 trees, where the instance represents a matrix and the class represent our output, ”true” or ”false”. (Figure taken from [AA19].)

input, resulting in often many hidden layers. The image classifier of the Autokeras library [JSH19] uses images or in this case matrices as input and gives the true and false as an output, creating the hidden layers automatically and such creating the neural network model (see Figure 4).

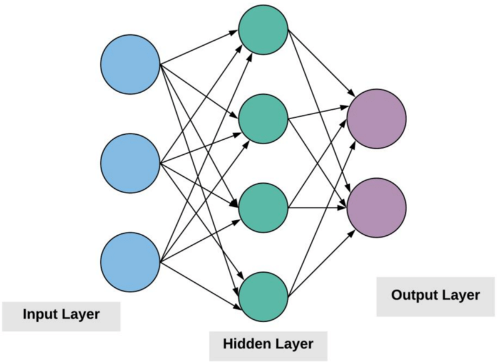


Figure 4: A simple neural network with one hidden layer, which is not the case in most instances. (Figure taken from [RFHF20].)

3 Method

This chapter will explain the data collection and generation as well as the models and model evaluation that are used.

3.1 Pipeline

To answer the research question, this thesis uses a pipeline for every size of naturally occurring and computer generated protein topology matrices. The naturally occurring protein topology matrices will get the label *True* and will be named the positive set and the computer generated protein topology matrices the label *False* and will be named the negative set. However, the total memory that is necessary to run all the matrices at once is not available at the University Leiden, so the matrices are split up in size by the power of 2, which will be both explained more in detail in the upcoming section.

- Collect all the protein topology matrices from the set of the real topology matrices of Section 3.2 with the same size.
- Generate the same amount of negative instances with the same maximum size, where the matrices represent fake topology matrices. Explanation of the generation of these matrices is in Section 3.4.
  - Both a complete random set and a statically matching set is generated, to compare both negative sets.
- The matrices are padded to the same size, otherwise the models will not accept the matrices.
  - This is done by adding zeroes on each side of the matrix to round the matrix to the maximum size that is chosen.
- Add all the instances together in one set.
- Shuffle and split the combined positive and negative instances 50/50 into a training and test set.
- Load the classifier and fit the training set into the model.
- Fit the test set on the classifier.
- Compare the prediction against the actual data to make a confusion matrix.
- Use the data in the confusion matrix to calculate the scoring metrics.

3.2 Positive set

The positive set will contain only real world protein topology matrices, that are built from existing proteins in the RCSB database.

3.2.1 The RCSB database

With the ever-increasing growth of the use of internet databases for microbiology research, the RCSB database has emerged as one of the biggest databases for proteins and other biological molecules. This database contains several useful structures, including DNA, RNA and protein structures, in total over 170,000 in 2021 [BBB+20]. To retrieve all the data used in this paper, the enclosed download batch file that can be found on the website of the RCSB is used to download all the protein structures in the so-called *.CIF* format. The *.CIF* format, also known as Crystallographic Information File, is a file format that contains the information about the atom sites in the structure, such as exact position and bonding with the protein itself and potentially other molecules [HM06]. Using the latter, we can build the topology matrices.

3.2.2 Building the matrices

Using the code made by Duane et al. [MBS+22], the database (accessed on 3-3-2021) is downloaded and converted into 2 sets of different matrices. The first set is the circuit diagram or as it is more commonly known as a touch-to-touch matrix, which is a matrix consisting of zeroes and ones, where the one denotes a connection between two points on the amino acid chain. In this matrix, it is not possible to see the topology; However, it is possible to calculate the distances between connections and count the number of connections. The second set is the topology matrix, which uses the legend described in Figure 5. This matrix can be used for calculating the distribution of the three common fold topologies as well as the sets that are going to be used as a positive set for the AI models, which will be explained later in this chapter. The first conversion delivered 174,212 different matrices, all of different size. Some of the files contained proteins that had to use logarithmic scale due to the scale of the proteins, so the *Biopython* could not convert these. After sorting out the empty matrices and the matrices that were too small, the set consisted 168,798 matrices for both the circuit diagram as the touch-to-touch matrix. The circuit diagram will be used as the positive instance that is mentioned in the pipeline of Section 3.1

Number	Abbreviation	Connection
1 , 2	P, P^{-1}	(Mirror) Parallel
3	S	Series
4	X	Cross
5 , 6	CP, CP^{-1}	Touching (Mirror) Parallel
7	CS	Touching Series
8	–	Itself

Figure 5: Table indexing the numbers in the csv files

3.3 Analysis of the positive set

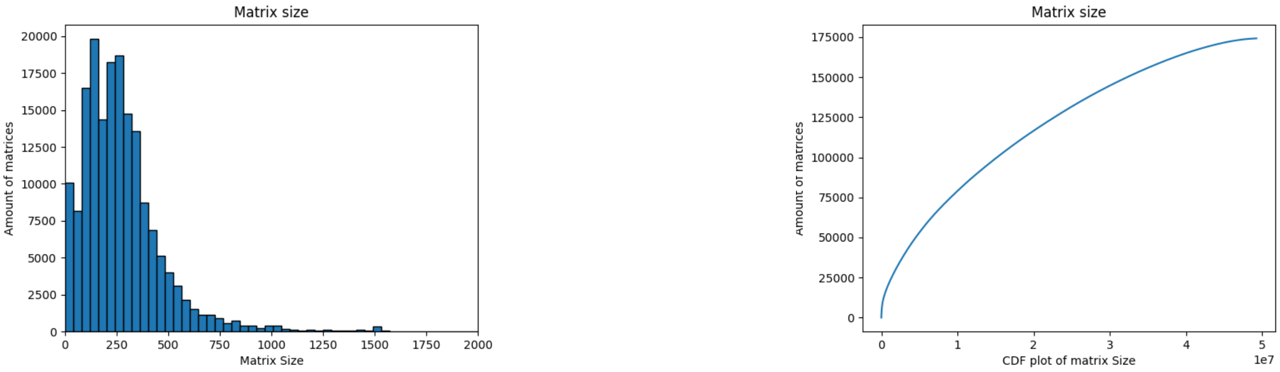
To create a more statiscally correct matrix, a number of metrics can be extracted from the positive set. In the next subchapters, multiple interesting metrics will be evaluated.

3.3.1 Basic structural properties

As mentioned before, the matrix translates to a chain, which means that the matrix is symmetric over the diagonal side; this can be seen in the matrices, where the diagonal line straight through the matrix represents the chain itself. This line also mirrors the matrix and such, this line is essential in the creation of a correct matrix.

3.3.2 Length of the chain

Every matrix represents, of course, an amino acid chain. The first interesting metric to consider is the length of the chain (or the size of the matrix). The average size of a chain is 282.69 amino acids with a rather large standard deviation of 251.86 amino acids. Also, the biggest matrix is 8,064 amino acids long and the smallest is only 1 amino acids. Figure 6a shows the distribution of the matrix size, where the x-axis of the graph is cut at 2,000 amino acids length, as after this length almost nothing can be seen. As can be seen, the distribution shows that the size is most likely to be in the lower range, between 100 and 500, which confirms the earlier mentioned mean. The left-skewed distribution can also be seen in the CDF plot (Figure 6b), where the almost vertical start in the left corner shows that most of the matrices sizes are located in the lower range.



(a) Size distribution

(b) CDF plot of the matrix size

Figure 6: Histogram and CDF plot of the matrix size

3.3.3 Length of connections

When looking at the touch-to-touch matrices (see Section 3.2.2), the length between the connections can be calculated, where each connection is placed on the amino acid chain. The average distance between two amino acids on the chain is 60 amino acids, with a standard deviation of 112 amino acids. Most fascinating is the maximum and minimum, 8,063 and 4 amino acids respectively. As shown in Figure 7, most of the distances are around this range. Again, the figure is cut off at 100 for better visuals.

In total, there were 61'551'909 connections in the set found, which means that a matrix had, on average, 365,64 connections. More in-depth analysis will be done in the next paragraph.

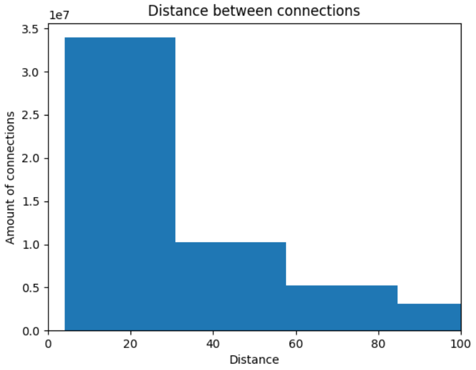


Figure 7: Histogram of the length of distances

3.3.4 Type of connections

The next interesting metric is the type of connections. As defined earlier, we have three basic types of topology connections, from which a distribution can be created for the negative set generator. The positive set generated an almost perfect mean for every type of connection, with Cross, Parallel and Series have a mean of 11.32% , 14.12% and 64.10% respectively. In the piechart (Figure 8) the big in balance between the three topology is clearly shown; the series topology dominates the matrices while the parallel and cross merely fill up the rest.

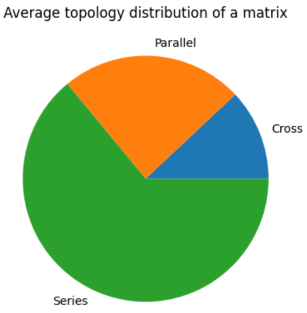


Figure 8: Piechart depicting the average % appearance rate of the connections

	Average	Maximum	Minimum	Standard deviation
Series	64.10%	96.84%	2.23%	13.39 percent point
Parallel	14.12%	91.82%	0.07%	9.48 percent point
Cross	11.32%	57.54%	0.49%	5.01 percent point

Figure 9: Average, maximum, minimum and standard deviation of each type of connection

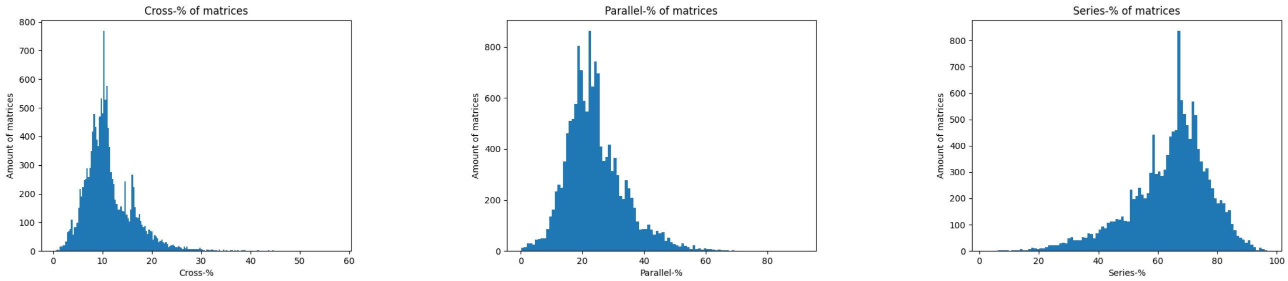
More interestingly is the distribution of each type of connection. The series has the biggest maximum, 96.84% and parallel has the lowest minimum, only appearing 0.07% in one matrix. Moreover, the standard deviation of all the connection is rather big, having a standard deviation of 13.39, 9.48 and 5.01 for series, parallel and cross respectively. The data can be seen in Figure 9.

3.4 Negative set

The negative set will only contain non-existent proteins, generated from the programs described in the next subchapters.

3.4.1 Generating random connections

The first negative set will be created using the random placement of connections on the chain. These connections can be placed on the aforementioned touch-to-touch matrix and these can be converted to a topology matrix. Firstly, the size of the matrix is taken from a uniformly distributed number with a minimum of 2 and a maximum of the maximum size in the pipeline. Secondly, a



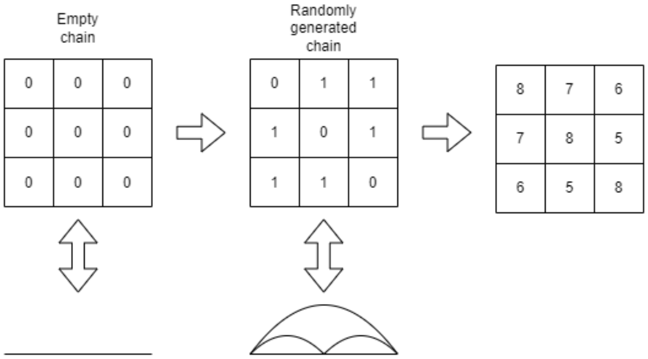
(a) Cross-% distribution

(b) Parallel-% distribution

(c) Series-% distribution

Figure 10: Distribution of the topologies

matrix full of zeroes is created with this size on both sides, which is equal to a chain with no connections. Thirdly, positions of connection are chosen at random, with exception of the diagonal line and already taken positions and filled in with ones accordingly, mirroring the diagonal line as well. After the matrix is filled with ones, the matrix is converted to the topology matrix. These topology matrices will be used as negative instances in the aforementioned pipeline in Section 3.1.



0	0	0
0	0	0
0	0	0

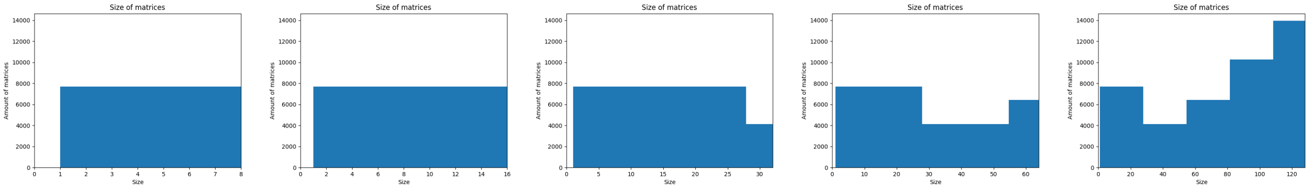
0	1	1
1	0	1
1	1	0

8	7	6
7	8	5
6	5	8

Figure 11: Generation of random connections in a matrix.

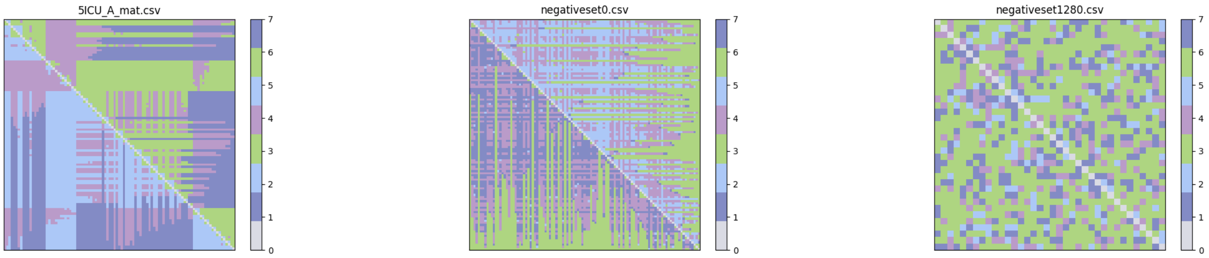
3.4.2 Further improving the realism of the negative set

The randomly generated matrices have been fairly random and as such, the next step is to make the negative matrices more realistic, by making the as statistically alike as possible to the positive matrices. Using the data collected from the analysis of the positive set, firstly, a less random number can be taken to determine the size of matrix. Using the different maximum sizes, calculated by the power of 2 (see Section 4), the different size distributions of the matrix size can be calculated. The next 5 graphs show these distributions, where a more realistic matrix size can be drawn from. Secondly, the amount of connections is generated from the percentages in Figure 8 and put randomly in the matrix. This causes the matrix to look fairly random, but statistically speaking, the matrix is correct (see Figure 13c). These matrices will also be used in the aforementioned pipeline mentioned in Section 3.1.



(a) Maximum size of 8 distribution (b) Maximum size of 16 distribution (c) Maximum size of 32 distribution (d) Maximum size of 64 distribution (e) Maximum size of 128 distribution

Figure 12: Distribution of the sizes, for each experimental maximum size



(a) A matrix from the positive set (b) A random matrix (c) A statically correct matrix

Figure 13: Examples of the three different matrices

3.5 Models

3.5.1 Random Forest

The default parameters that are in the *scikit-learn* [PVG+11] library were used. In the version uses, the number of trees created is set at 100 and no maximum depth of the tree with a time limit of 4 days, where the program is automatically stopped after this amount of time is reached.

3.5.2 Neural Network

The standard *Image classifier* was used in the *Autokeras* [JSH19] library, with a train time limit of 4 days, where the program is automatically stopped after this amount of time is reached.

3.6 Model evaluation

3.6.1 Measures

- True Postives (TP): A matrix or instance that is real and correctly classified as real.
- True Negatives (TN): A matrix or instance that is fake and correctly classified as fake.
- False Positives (FP): A matrix or instance that is fake and incorrectly classified as real.
- False Negatives (FN): A matrix or instance that is real and incorrectly classified as fake.

These measures on their own are not enough to grade the models. There are multiple other evaluators to calculate to see how the model performs.

3.6.2 Accuracy

Accuracy is calculated using the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Accuracy is one of the main evaluators we will be using as this metric visualizes well how good the machine learning model is at predicting the matrices.

3.6.3 Confusion Matrix

A confusion matrix is a visual metric, where the measures are shown in another matrix:

		Predicted	
		True	False
Actual	True	TP	FN
	False	FP	TN

Figure 14: A confusion matrix, with the predicted labels at the x-axis and the actual labels at the y-axis

The visualisation offers a better visual overview than the other measures, which are numbers. This is the reason it is used in this paper.

3.6.4 Precision

This metric shows how well the model is at classifying a true matrix as not fake. It is calculated using the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The precision is a number between 0 and 1, with 1 being the best and 0 being the worst. We will not be using this metric on its own, instead it will be used to calculate another metric.

3.6.5 Recall

Better known as the True Positive Rate, this metric shows how well the models classifies the positive samples and is calculated using the following formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The recall is a number between 0 and 1, with 1 being the best and 0 being the worst.We will again not be using this metric on it’s own, instead using it to calculate another metric.

3.6.6 F1-score

F1-score is the average of both the precision and the recall and is calculated using the following formula:

$$\text{F1-score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

The F1-score is also a number between 0 and 1 and is often used to calculate the ability of a model, which will be useful for comparing the two models against each other.

4 Results

The results are split up into 3 sections: The dataset split, the model comparison and the negative set comparison.

4.1 Matrix size

Due to the size of the data set, running the all the instances at once is not possible. The data set is split up based on matrix size which can be seen in Table 1. This table also contains the amount of matrices that were within this size. The larger the size, the more features the model has to take into account, which grows quadratically.

Dimensions	8	16	32	64	128	256
Size in memory	782	2454	4768	9398	26146	70308

Table 1: Matrix size and amount of matrices per size.

We can use these sizes to test whether first of all the server can handle this amount of matrices and second of all to find an answer to the research question.

4.2 Model comparison

4.2.1 Confusion Matrices

The confusion matrices show insight into how well the models performed using the different sets. Using these results, it is possible to see which model performs better overall and which set challenges the models the most.

	True	False
True	354	33
False	35	360

(a) 8-set

	True	False
True	1217	51
False	11	1175

(b) 16-set

	True	False
True	2369	48
False	17	2334

(c) 32-set

	True	False
True	4699	35
False	38	4626

(d) 64-set

	True	False
True	12920	60
False	16	13150

(e) 128-set

Figure 15: Confusion matrices of the random set using the random forest model.

	True	False
True	399	0
False	3	399

(a) 8-set

	True	False
True	1220	3
False	0	1231

(b) 16-set

	True	False
True	2364	15
False	0	2389

(c) 32-set

	True	False
True	4639	19
False	0	4740

(d) 64-set

	True	False
True	13315	38
False	0	12993

(e) 128-set

Figure 16: Confusion matrices of the statistical set using the random forest model.

	True	False
True	377	51
False	9	345

(a) 8-set

	True	False
True	1206	65
False	8	1175

(b) 16-set

	True	False
True	2385	77
False	9	2297

(c) 32-set

Figure 17: Confusion matrices of the random set using the neural network model

	True	False
True	332	54
False	60	336

(a) 8-set

	True	False
True	1193	35
False	38	1188

(b) 16-set

	True	False
True	2307	21
False	38	2402

(c) 32-set

	True	False
True	2307	21
False	38	2402

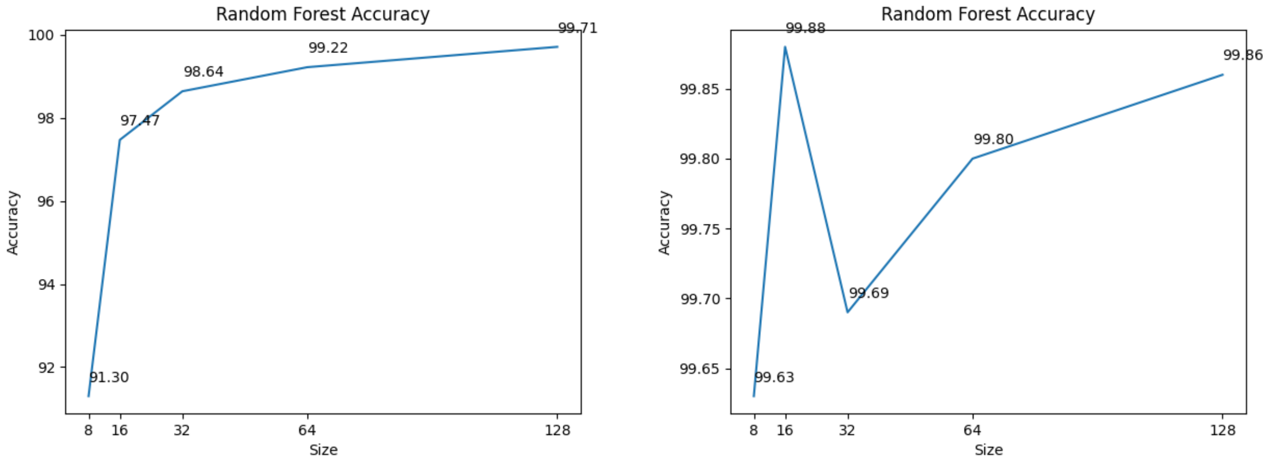
(d) 64-set

	True	False
True	13064	0
False	2	13080

(e) 128-set

Figure 18: Confusion matrices of the statistical set using the neural network model.

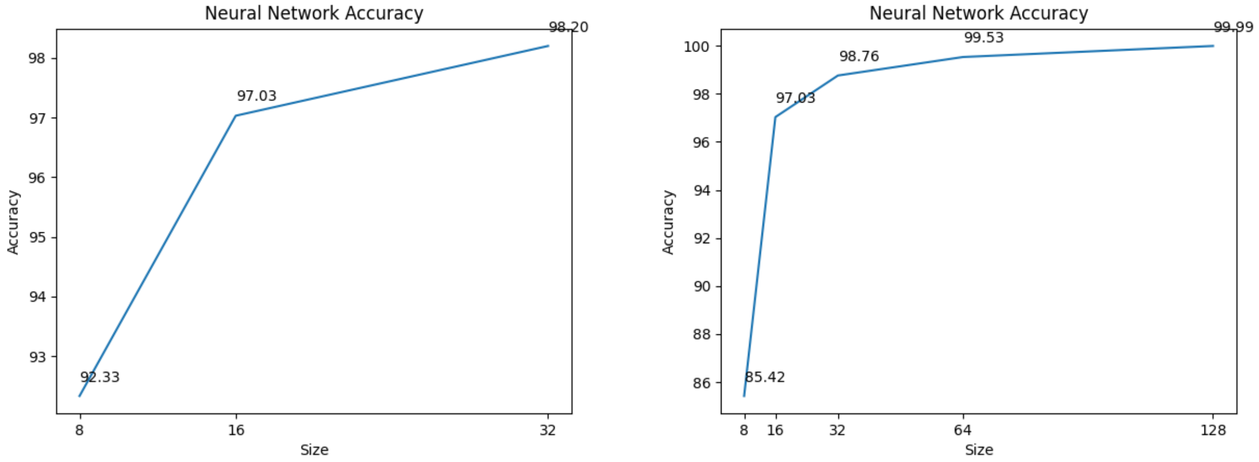
All of the 256-sets did not fit the server memory, so these are not shown in the figures. In Figure 18 the 64-set and 128-set are missing as well, because the neural network program extended beyond the 4 days time limit. While both the sets were correctly padded and put into the test and training set, the neural network took more than 4 days to train itself on the training set. Remarkable is that the random forest tended to make more False Negative errors in 3 of the 5 datasets, while making about the same amount of errors in the other two sets (see Figure 16). However, the neural network made approximately the same amount of false positive and true negative errors (see Figure 18 and Figure 16).



(a) Accuracy of random forest on the random set

(b) Accuracy of random forest on the statistical set

Figure 19: Accuracy of the random forest.



(a) Accuracy of neural network on the random set

(b) Accuracy of neural network on the statistical set

Figure 20: Accuracy of the neural network.

In Figure 19, the accuracy of the random forest is shown. The 8-set generally scores a lower accuracy, with the random set scoring 91.30% and 99.63% for the random set and statistical set respectively. The graphs show that the random forest generally scored better than the neural network in both sets, which can be seen in Figure 20, where the random forest scored more then 99 % on all the statistical sets when the neural network scored an 85.42% on the 8-set and scored 97.03% and 98.76% on the 16 and 32-set respectively.

4.2.2 Model score

The model scores show how well the models performed against each other on the same sets, making a good comparison means for the two models.

		Random	Forest			Neural	Network	
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
8-set	0.91	0.91	0.91	0.91	0.92	0.97	0.88	0.92
16-set	0.97	0.99	0.96	0.98	0.97	0.99	0.95	0.97
32-set	0.99	0.99	0.98	0.99	0.98	0.99	0.97	0.98

Table 2: Random set metrics

		Random	Forest			Neural	Network	
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
8-set	0.99	0.99	1	1	0.85	0.85	0.86	0.84
16-set	0.99	1	0.99	1	0.97	0.97	0.97	0.97
32-set	0.99	1	0.99	1	0.98	0.98	0.99	0.99

Table 3: Statistical set matrices

Table 2 shows a small difference between the accuracy, precision, recall and F1-score of the two models on the random set. Overall, the random forest performed a bit better. In contrast, Table 3 shows a significant difference between the models. The neural network scored lower on accuracy, precision, recall and the F1-score on the statistical set, where the random forest scored close to 1, the highest score.

5 Discussion

5.1 Data and technological changes

The RCSB database is still growing at an enormous rate of about 10% per year [BBB+20] and also receives regular updates for the formatting and the user interface. This also means that the data is keen to changes and updates, which means the data this thesis uses could be outdated soon. Moreover, the computational power of the University Leiden is still too small for some of the 128-sized sets and all the 264-sized sets. However, this thesis showed the first indicators that the bigger sets only improve the accuracy of the models.

5.2 Folding in nature versus artificial folding

A possible extension to this thesis could be the use of artificially folded proteins. This thesis only uses proteins that occur in nature and thus the models are only fitted for these kinds of proteins. With artificial folding already existing since 1988 [MAA+88] and more and more research being done into design proteins, strengthened by the use of bioinformatics [TKD+13], the models should also be trained and tested for these kind of proteins, as this is a promising field of protein research.

6 Conclusions and Further Research

In this thesis, a pipeline is created that uses a random forest or an automated neural network model and generates the topology matrices needed for the positive and the negative set, generating the positive set from the RCSB database and building two different negative sets, a random set and a statistical set. To answer the research question of this paper, "Can AI classify wrongly folded proteins based upon their topology matrix?", the paper has assessed multiple models and negative sets to test this question. It can be concluded that for at least the two AI models tested in this paper, the random forest and the neural network, it is possible to distinguish real matrices from fake matrices, with high accuracy.

Furthermore, the models were more prone to failure with the random set than the statistical set, where the accuracy was generally above the 99%. The random set seemed to challenge the models more than the statistical set, where the general accuracy was lower. This could be because the random set may have shown certain patterns that are present in real protein matrices and thus in nature. The scoring metrics, mainly the F1-score, also showed that the random forest model performed better than the neural network, on both the random and statistical sets. The statistical set scored significantly worse on the neural network, whereas the random set only showed a small difference in score between the models.

This thesis opens up a new gateway for the combination of the usage of AI with biology, but also on more levels. The circuit topology theory itself offers one of the first definitions of the folding of proteins, where this theory could also be used in other fields. This unique insight into the usage of AI on these kind of matrices makes way for multiple new innovation in the field of topology, but there is still a lot more to research. As mentioned earlier, this thesis only tests two different kind of negative matrices, but much more could be made by computers or by mutating proteins and such their folding. Secondly, the next generation of computers will hopefully offer more memory for more sets to run and quicker run times as well, which will make this research much quicker and more robust as the data-set can be way larger. Thirdly, moving the research scope towards which patterns are decisive and recognised by the AI models could be a big potential keystone in the world of protein topology.

References

[AA19] Bandar Alghamdi and Fahad Alharby. An intelligent model for online recruitment fraud detection. *Journal of Information Security*, 10:155–176, 2019.

[AJL+14] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell.* Garland Science, sixth edition, 2014.

[BBB+20] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, Sai Ganesan, David S Goodsell, Sutapa Ghosh, Rachel Kramer Green, Vladimir Guranović, Dmytro Guzenko, Brian P Hudson, Catherine L Lawson, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Persikova, Chris Randle, Alexander Rose, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Yi-Ping Tao, Maria Voigt, John D Westbrook, Jasmine Y Young, Christine Zardecki, and Marina Zhuravleva. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*, 49(D1):D437–D451, 11 2020.

[Bre01] Leo Breiman. *Machine Learning*, 45(1):5–32, 2001.

[CZY19] Xing Chen, Chi-Chi Zhu, and Jun Yin. Ensemble of decision tree reveals potential mirna-disease associations. *PLOS Computational Biology*, 15(7):1–24, 07 2019.

[GM22] Anatoly Golovnev and Alireza Mashaghi. Topological analysis of folded linear molecular chains. In *Topological Polymer Chemistry*, pages 105–114. Springer, 2022.

[Gol99] David P Goldenberg. Finding the right fold. *Nature structural biology*, 6(11):987–990, 1999.

[HJS+21] Soha Hassoun, Felicia Jefferson, Xinghua Shi, Brian Stucky, Jin Wang, and Epaminondas Rosa. Artificial intelligence for biology. *Integrative and Comparative Biology*, 61(6):2267–2275, 2021.

[HM06] Sydney R. Hall and Stephen B. McMahon, editors. *International Tables for Crystallography.* International Union of Crystallography, 2006.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[Hog11] Paulien Hogeweg. The roots of bioinformatics in theoretical biology. *PLoS Computational Biology*, 7(3):e1002021, March 2011.

[JSH19] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956. ACM, 2019.

[LYY05] Alan Wee-Chung Liew, Hong Yan, and Mengsu Yang. Pattern recognition techniques for the emerging field of bioinformatics: A review. *Pattern Recognition*, 38(11):2055–2073, 2005.

[MAA⁺88] Manfred Mutter, Eva Altmann, Karl-Heinz Altmann, René Hersperger, Pjotr Koziej, Kurt Nebel, Gabriele Tuchsecherer, Stéphane Vuilleumier, Hans-Ulrich Gremlich, and Klaus Müller. The construction of new proteins. part iii. artificial folding units by assembly of amphiphilic secondary structures on a template. *Helvetica chimica acta*, 71(4):835–847, 1988.

[Mal08] Edyta B Malolepsza. Modeling of protein misfolding in disease. *Molecular Modeling of Proteins*, pages 297–330, 2008.

[Mas21] Alireza Mashaghi. Circuit topology of folded chains. *Notices of the American Mathematical Society*, 68:420, 03 2021.

[MBS⁺22] Douane Mous, Elnaz Banijamali, Vahid Sheikhhassani, Barbara Scalvini, Jaie Woodard, and Alireza Mashaghi. Proteinct: An implentation of the protein circuit topology framework. Article under review, 2022.

[MG95] Kenji Mizuguchi and Nobuhiro Go. Seeking significance in three-dimensional protein structure comparisons. *Current Opinion in Structural Biology*, 5(3):377–382, 1995.

[Mo111] *Data mining: Practical machine learning tools and techniques.* Morgan Kaufmann, Oxford, England, 2011.

[MvWT14] Alireza Mashaghi, Roeland J. van Wijk, and Sander J. Tans. Circuit topology of proteins and nucleic acids. *Structure*, 22(9):1227–1237, September 2014.

[NYL⁺19] Nagasundaram Nagarajan, Edward K. Y. Yapp, Nguyen Quoc Khanh Le, Balu Kamaraj, Abeer Mohammed Al-Subaie, and Hui-Yuan Yeh. Application of computational biology and artificial intelligence technologies in cancer precision drug discovery. *BioMed Research International*, 2019:1–15, November 2019.

[PVG⁺11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RFHF20] Taki Hasan Rafi, Faisal Farhan, Md.Ziaul Hoque, and Mohd Farhan. Electroencephalogram (eeg) brainwave signal-based emotion recognition using extreme gradient boosting algorithm. 1:1–19, 2020.

[RjM⁺20] Jeff Reback, jbrockmendel, Wes McKinney, Joris Van den Bossche, Matthew Roeschke, Tom Augspurger, Simon Hawkins, Phillip Cloud, gfyoung, Sinhrks, Patrick Hoefler, Adam Klein, Terji Petersen, Jeff Tratner, Chang She, William Ayd, Shahar Naveh, JHM Darbyshire, Richard Shadrach, Marc Garcia, Jeremy Schendel, Andy

Hayden, Daniel Saxton, Marco Edward Gorelli, Fangchen Li, Torsten Wörtwein, Matthew Zeitlin, Vytautas Jancauskas, Ali McMaster, and Thomas Li. pandas-dev/pandas: Pandas, 2020.

[Sca] Barbara Scalvini. Unknown title.

[SRM+12] Joanna I Sułkowska, Eric J Rawdon, Kenneth C Millett, Jose N Onuchic, and Andrzej Stasiak. Conservation of complex knotting and slipknotting patterns in proteins. *Proceedings of the National Academy of Sciences*, 109(26):E1715–E1723, 2012.

[Tez13] Yasuyuki Tezuka. *Topological polymer chemistry: progress of cyclic polymers in syntheses, properties, and functions*. World Scientific, 2013.

[TKD+13] Christine E Tinberg, Sagar D Khare, Jiayi Dou, Lindsey Doyle, Jorgen W Nelson, Alberto Schena, Wojciech Jankowski, Charalampos G Kalodimos, Kai Johnsson, Barry L Stoddard, et al. Computational design of ligand-binding proteins with high affinity and selectivity. *Nature*, 501(7466):212–216, 2013.

[TO01] Yasuyuki Tezuka and Hideaki Oike. Topological polymer chemistry: systematic classification of nonlinear polymer topologies. *Journal of the American Chemical Society*, 123(47):11570–11576, November 2001.

[Wis08] David S Wishart. Identifying putative drug targets and potential drug leads. In *Molecular Modeling of Proteins*, pages 333–351. Springer, 2008.

[WM10] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[YNK07] Todd O Yeates, Todd S Norcross, and Neil P King. Knotted and topologically complex proteins as models for studying folding and stability. *Current Opinion in Chemical Biology*, 11(6):595–603, 2007. Model systems/Biopolymers.