Master Project Report

Hi-C Predictions based on protein levels

Andre Bajorat

Supervisors: Joachim Wolff, Anup Kumar

Albert-Ludwigs-University Freiburg

Faculty of Engineering

Department of Computer Science

Chair of Bioinformatics

September 23thth, 2019

Contents

List of Figures

1 Introduction

Deoxyribonucleic acid (DNA) provides the instructions for building molecules in organisms and is responsible for genetic expression by storing the instructions on how to make proteins. To get a better understanding of the role DNA has, it is important to have a look a the 3-D structure of the genome. It helps to understand interactions on an inter-chromosomal(interactions of chromosomes with each other) level, e.g. for Ribonucleic acid(RNA) secondary structure (compare [Tagami et al., 2010]), and how the chromosomes arrange themselves inside the nucleus. It also shows interesting features on an intra-chromosomal (chromosome regions interacting within the same chromosome) level, like Topically Associating Domains (TAD) and A-B compartments. One of the most common approach to investigate chromosomal structure nowadays is the Hi-C-technique, extending previous methods like chromosome conformation capture (3C, one vs. one), Chromosome conformation capture-on-chip (4C, one vs. all) and Chromosome conformation capture carbon copy (5C, many vs. many). Hi-C is considered an all-vs-all approach. Hi-C was proposed by Lieberman-Aiden et al. [2009] and calculates a contact matrix for the loci of the whole genome, where each entry denotes the count of contacts between the regions. The resulting matrix is denoted as Hi-C matrix and is crucial to gain information on the structure of chromosomes and their relation to each other and themselves. With the help of Hi-C matrices, it is possible to detect A-B-compartments in the genome, where one ("A" compartment) entails typically gene-rich regions where histones act as gene-enabling whereas the "B" compartments' histones tend to silence their genes. Usually the

"A" compartments lie in the inner parts of the nucleus while the "B" compartment make up the peripheral regions. Another interesting feature to be deduced by Hi-C matrices are Topically Associating Domains (TAD), which represent regions where loci tend to interact a lot with each other. These regions form triangles in the Hi-C matrix where all of the contact pairs have high read values. That shows that loci inside the TAD's tend to interact more than on average. They represent loops in the genome and the outer most loci define the region of the base pairs which are located at the binding site. These loops are related to gene-regulatory functions (compare [Lajoie, Dekker, Kaplan, 2015]). Other experiments have shown that Hi-C matrices can be used to determine chromosomal regions inside the nucleus and how the whole genome is structured on an inter-chromosomal level(compare [Cristescu, Borsos, Lygeros, Rodríguez Martínez and Rapsomaniki, 2018]). Unfortunately, it is expensive in terms of memory and time effort to calculate said Hi-C matrices. Therefore it might be helpful to look out for new approaches to create or simulate Hi-C matrices. One of them was suggested by Zhang, Chasman, Knaack and Roy [2018] and uses protein data that is available for many cell lines and organisms and not as complex to measure. Hi-C matrices are correlating with several proteins and histones, especially at the boundaries of TAD's, the supposed binding sites of loops in the chromosomes. Therefore Zhang, Chasman, Knaack and Roy [2018] tried to predict Hi-C contact reads by using Machine Learning algorithms. Machine Learning(ML) can be helpful in this scenario, as many ML algorithm predict targets, using huge input data, by searching for patterns and structures in the data. In this case, protein and histone data were used as input data along with the genomic distance between the loci. Since the idea and the first experiments showed to be promising, this project aims to replicate their results and to possibly find new approaches to improve the algorithm.

If it can be done to predict Hi-C matrices just by protein and histone levels at specific loci it may help to drastically improve the availability of Hi-C matrices for research purposes. Protein and histone levels can be easily calculated and are widely available,

whereas the creation of Hi-C matrices is expensive in terms of computation time and material costs. It is also prone to a lot of errors if the procedure is not undertaken carefully. Even then Hi-C matrices might miss out on a lot of data since there will be impure data and mistakes in the biological procedure.

Even though no prediction can truly replace real data it still is of great value to have an approximation of Hi-C matrices easily available to conduct biological experiments that can later be validated with real Hi-C matrices.

It might also help to improve the process of creating these Hi-C matrices in the long term.

2 Related Work

In the first part of this section, the process of creating a Hi-C matrix will be explained. Then the role of proteins and histones in relation with the Hi-C matrix will be discussed. Finally, the approach of Zhang, Chasman, Knaack and Roy [2018] that motivates this project will be presented.

2.1 Hi-C

Hi-C as proposed by Lieberman-Aiden et al. [2009] aims at calculating contact matrices for the whole genome by linking close regions together and processing the resulting library. The contact matrix then denotes the amount of reads that have been identified for each pair of regions. In a first step, spatially adjacent segments of the DNA are cross-linked with the organic compound formaldehyde and then cut into pieces with a restriction enzyme(compare Figure 1 [Lieberman-Aiden et al., 2009]). The resulting ends are then filled with nucleotides. The ends are marked with the vitamin biotin and the ends of each side are ligated between the cross-linked fragments. The DNA is then purified and sheared into many pieces. The next step is to filter out the ligated pieces that contain fragments of two fragments that were close to each other in the nucleus. Since the protein streptavidin is attracted to biotin, streptavidin beads serve as a selector, binding to the biotin. The resulting library is then processed with massively parallel DNA sequencing. Each pair is then uniquely aligned to a reference sequence to find out to which regions the two fragments of

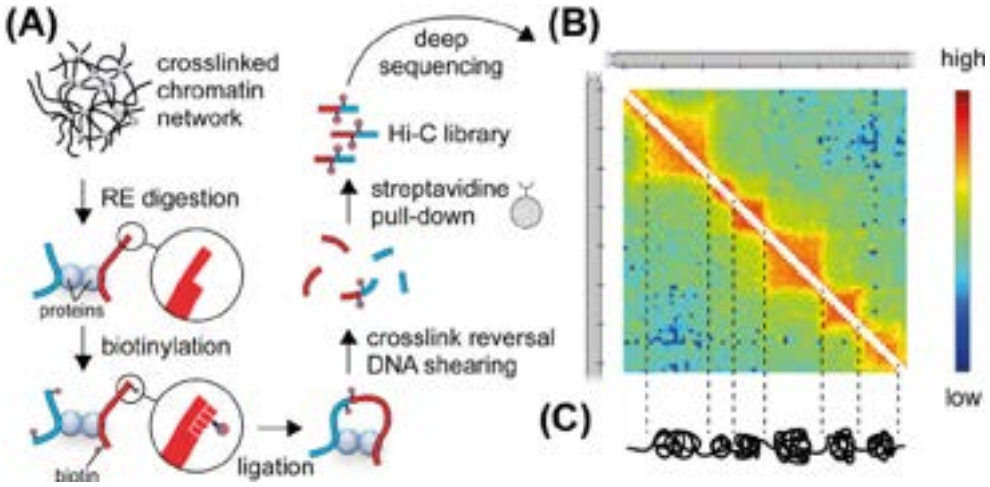


Figure 1: Example for Hi-C process

each pair belong. If there is no unique alignment for either one, the pair is dropped. The contact matrix is then defined by setting the value m_{ij} to the number of ligation pairs that have been identified as belonging to region i and region j. The matrix can then be used for different procedures. Principal components, linearly uncorrelated vectors, can be calculated to determine the compartments or the Hi-C matrix can be visualized as a heat map.

Unfortunately there are a lot of limitations to the Hi-C approach. First, the procedure requires abundant samples, since the efficiency of the steps is quite low, and it requires a lot of replications to construct coherent results. Second, the approach requires several steps that take a couple of hours resulting in complex and time-consuming experiments to create a Hi-C matrix.

2.2 Proteins

Lan et al. [2012] have shown that some proteins are highly correlated with the aforementioned Hi-C reads and should therefore be included when trying to predict Hi-C matrices.

CTCF might be one of the most important proteins for prediction since it is a binding factor that binds segments of DNA to form loops. It is often observed when long-range loops occur. Furthermore, it is important to include several histone proteins that act in different ways related to the expression of genes. DNA is spooled around histones as these give the DNA a frame for its structure. H3k27me3 and H3k9me3 play a role in repressing genes whereas H3k36me3, H4k20me1, H3k79me2 relate to elongation (producing several copies). H3k4me1 and H3k27ac serve as enhancer marks and H3k9ac, H3k4me2, H3k4me3 are said to be related to active genes(compare [Karlić et al., 2010]).

Besides the histones, some more proteins correlate to Hi-C reads. The cohesin component RAD21 is important for chromosome segregation and DNA repair, the general transcription factor TBP is also relevant to the binding of proteins. DNase I is often observed in regions with active genes. Finally, we have SMC3, which is important for the regulation and structure of the chromosomes, e.g. by holding sister chromatids together during cell replication. All of these proteins are to be included to use them for the prediction of Hi-C matrices. Since they all play a relevant role in the structure of chromosomes they will help to deduce dependencies which can predict the structure of a genome.

2.3 HiC-Reg

Zhang, Chasman, Knaack and Roy [2018] proposed an approach called Hi-C-Reg that uses Random Forest Regression to predict Hi-C reads based on protein data. Only contact pairs on a chromosome level are considered and only loci that are in a certain range from each other. That is consistent with the observation that most of the structures are recognizable when the loci are close to each other at up to 2 million base pairs (mb). For each contact pair, a data entry is created. For each of the chosen proteins and each of the two loci, the binned protein data for that region is stored, as well as the genomic distance. To train the classification model it is also necessary to

store the target of the regression, which would be the entry of the Hi-C matrix for the corresponding loci, the interaction read value. One suggestion of Zhang, Chasman, Knaack and Roy [2018] was to include information about proteins that lay in between the two loci so that the regressor has additional information about the protein levels in the surrounding region. But there are two obstacles to this approach. First, there might be a lot of values in between since the genomic distance can be up to 2mb. Second, the amount of regions in between varies since it is necessary to predict values for different genomic distances. But, as long as only a single regressor is used, the data must have the same format. It is not possible to vary the length of the input data to include all the values. Therefore the authors decided to take the mean of all the proteins to normalize the procedure. This approach is called the "Window approach" and resulted to be much more efficient than just considering the proteins of the exact loci.

Experiments were conducted with 5 chromosomes (9,11,14,17,19) over 5 different cell lines (Gm12878, K562, Huvek, Nhek, Hmek). One experiment investigated how models trained by one cell line can predict chromosomes on other cell lines and there was an indication that it works. But, generally the results were not conclusive and it was just a tendency that was observed. It worked in some cases but also did poorly in others.

3 Methods

In the following the framework implemented during this project is to be presented. The framework consists of many different steps to train a model that can predict Hi-C-reads by using protein data as input. The protein data has to be binned to fit the intervals of the Hi-C matrix and can be optionally altered. The training and test sets have to be created with many options on how to do so. Then the training sets are used to train a regression model, that can have some different parameters. In the current version only Random Forests are supported. Finally, we can predict Hi-C-reads using the regression model and there are several ways to evaluate the results or even to plot the Hi-C-matrix. Each step will be explained in the following while pointing out the different optional settings that were implemented for testing. For all of these settings, experiments will be presented along with expectations on the effects. It is worth mentioning that the framework in its current version only works with already existing Hi-C matrices and is, therefore, a mere testing and experimenting framework as a proof of concept, instead of being able to predict unknown genomes. For now, it relies on the binning structure of existing Hi-C matrices even when predicting. Each data set can, therefore, be used for training and testing, all of them contain the read values as given by the Hi-C matrix. In the case of predictions, these read values are cut from the test set and only the protein data is passed along.

3.1 Protein Binning

Since the goal of the framework is to predict Hi-C reads by just using protein data as an input, the protein must be binned and might be preprocessed in other ways like normalization. First of all, a set of proteins and histones were chosen that are correlated to Hi-C reads or have an important role in the binding of loops in the DNA. In this experiment, as proposed by Zhang, Chasman, Knaack and Roy [2018], the following proteins and histones were chosen: CTCF, RAD21, SMC3, H2az, H3k4me1, H3k4me2, H3k4me3, H3k9ac, H3k9me3, H3k27ac, H3k27me3, H3k36me3, H3k79me2, H4k20me1. Those proteins are easily available in the narrowpeak-format, assigning protein peaks to specific positions in each chromosome. Since Hi-C matrices always use binned data by grouping strands of a specific length (resolution or bin size) into one region, the same procedure must be applied to the proteins. Corresponding to the bins of the Hi-C matrix used for training and/or testing, the protein data must be binned with the same positions to create coherent input data. The script "createBaseFile" takes care of this part of the process. The specific cell line and the resolution in kilo base pairs(kb) can be defined. Additionally, the user can set three optional parameters.

- Normalize Protein Data to 0-1-Range

  This approach should not have strong effects on the quality of training, but it is important to ensure that protein data extracted from different experiments have the same effects and normalization is an approach often used for Machine Learning algorithms.

- Bin Operation

  When binning the protein peaks there are several options to be chosen. The maximum value of all of the peaks in the specific region can be chosen as representative for this region or the average can be calculated. Other options might be possible but so far these are the ones implemented in this project. In

an earlier version there was also the option to use the sum of all peaks. But since the bins have always the same size, there is no meaningful information to be gained by doing so. The user can set the desired option. There could be a considerable effect when changing this setting. Binning by using the maximum would highlight big peaks as in an average scenario small peaks might have a bigger influence on the resulting value. This might lead to the better representation of high peak areas.

- Protein Peak Column

  The user can opt to choose the column of the data format that is used to extract the scores from provided protein files. The standard setting is 6 for the used narrowPeak format (7 when not using zero-based indexing) since this column denotes the signal value in standard narrowpeak file formats. In early intents of predicting Hi-C matrices during this project a wrong column (4, score) has been chosen, so this option enables the evaluation of the improvement gained by choosing the correct column. It also allows to use other data formats like broadPeak or gappedPeak if the user wishes to.

3.2 Creating datasets

The next step consists of creating the necessary data sets for training and testing. To train and predict the whole Hi-C matrix it would be necessary to calculate one row for each possible pair of regions. For n bins, n x n rows would have to be calculated. This is not feasible because of the length of the genomes as it would result in enormous data sets that would consume too much memory and runtime. But the properties of the Hi-C-matrix and the special circumstances of this experiment can be exploited to reduce runtime and memory consumption. First of all, only the upper or lower triangular matrix has to be considered since it is symmetric. Furthermore, the interesting areas of the matrix to recognize DNA loops are close

to the diagonal. Therefore we can define a moving window approach considering only the next m regions to combine with each region. That reduces runtime and memory time drastically since the interesting area is about 2mb wide, which e.g. for a resolution of 5kb ensures that only mxn with $m = 400$ rows have to be calculated. An average chromosome has tens of thousands of bins for a resolution of 5kb, so reducing that to only 400 basepairs is very impacting for memory and runtime. A positive side-effect is that we do not use those parts of the matrix that are not as relevant for loops as training input. This allows the regressor to concentrate its efforts on the relevant parts. For each of these chosen pairs, a set with the following features is calculated. The 14 proteins of the first binned region, the 14 proteins of the second binned region, a representative value for each of the 14 proteins for all the regions in between, and the genomic distance between the two regions. Furthermore, some metadata is stored, the chromosome, the absolute position of the first region and the cell line. The metadata is not used for training so far. Additionally, the Hi-C-value at the specific position determined by the two regions is stored as a target in case of the training sets and, if available, for the test sets.

The following options can be set for the creation of the data sets.

- Bin operation for in-between regions

  Again there is a choice to be made on how to bin the protein data. In this case, it is about the proteins of the regions between the two regions on focus. This data has a lot of potential of great influence on the training since one of the regions might be inside of a loop while the other one is outside of it. The proteins can be binned by applying the average, the maximum or the sum . Again the sum is redundant, since the average function together with the distance between the regions gives us the same information. But it might be that this relation is hard to detect for the regression algorithm, so it is going to be included here, hopefully yielding the same results as the average function. The maximum function, on the other hand, may give different results, similar

to the previous binning case.

- Ignore Centromeres

  Chromosomes usually have a centromere region that connects a pair of sister chromatids. These regions are not interesting in terms of the Hi-C when looking for loops or compartments as they feature the same value for the whole centromere. Taking these regions out should help the regressor to improve its predictions by focusing on the relevant data. Therefore it can be decided to eliminate these regions from the data sets by providing a file that enlists the centromere regions for each chromosome. Otherwise the default file of the package is used. There is a distinction to be made between the different types of centromeres a chromosome can have. Some of the chromosomes are classified as metacentric(Chromosomes 1, 3, 16, 19, and 20), meaning that both arms are equally long, some as submetacentric (Chromosomes 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17 and 18) with arms of different sizes and finally acrocentric chromosomes (Chromosomes 13, 14, 15, 21, 22) where one arm is barely detectable. It will be interesting to see if this has any influence on the prediction.

- Equalize Proteins (deprecated)

  This option was an approach to manually redesign the input data to fit our knowledge about the situations in which loops can occur. As shown by Lan et al. [2012] a loop is normally accompanied by protein peaks in both regions. If just one of the regions or none of them shows a peak it entails that these specific regions do not interact together to form a loop. When creating the data entries for each possible interaction between regions, the idea is to set a protein peak to 0 if there is no activity shown in the protein of the partner region. This was supposed to be helpful since any of these interactions can not

feature a loop correlated to the protein since they would have to be activated in both proteins, as it works according to biological research. Unfortunately, this approach resulted in worse results. One of the reasons might be that we manually manipulate and falsify data instead of letting the algorithm figure out this connection. Furthermore, we do not just try to predict loops but any interaction value. Loops only make up a small fraction of the data entries and with this framework, the intent is to predict the whole matrix, and therefore a lot of regions, where not a lot of activity takes place. Especially the edges of triangles in the visualized Hi-C matrices did suffer since a peak in one region with no activity in the other region can be a hint for a loop that lies inside or outside of the regions. Eliminating this information might disable the regression algorithm to predict these interactions. Aside from the bad results, it resulted that this approach could not be combined with the vectorized creation of data sets in a newer version of this project, because each pairs would need to be evaluated separately. Because of these reasons, it was taken out of the framework.

3.3 Training

In the next step, the created training sets can be used to train models. The user just has to pass the training set to the script and the model will be trained by applying Random Forest (using the sklearn implementation of Random Forests) regression and stored in a zip file (UNIX-compressed z). In an earlier version, it was possible to combine several data sets of the same genome featuring different chromosomes, but research has shown that there is a higher connection between the same chromosome along different cell lines than there is between chromosomes on the same cell line. Therefore this approach was taken out(compare [Nagano, Lubling, Stevens, Schoenfelder, Yaffe, Dean, Laue, Tanay and Fraser, 2013]). Additionally, the user can decide to set one more option.

- Conversion method

  As the read values of the interaction matrix range from rather small to large values, the usual approach is to plot them after applying the log function. This allows us to detect rather small differences in the lower ranges whereas the differences in the high ranges are no that much accentuated. Because the intensity of the interaction is also highly dependent on the genomic distance, applying the log function helps to decrease that effect. In the same manner, the user can decide to train on just the read values or to use the log values. This is supposed to have a huge effect on the regressor's performance. Without applying the log function the regressor is supposed to focus much more on interactions with a lower genomic distance since those have generally higher read values. This is an effective way to reduce the error of the loss function the model uses for its training. Training on log reads is supposed to cause an improvement for the prediction of middle to long-range interactions while the small range interactions might decrease in terms of prediction accuracy. Because the log function changes higher values proportionally much more than smaller values, the loss function will change its behavior and thereby the regressor its focus.

3.4 Prediction

After training the model the framework can be used for the actual prediction. A model file and a test set file must be chosen. It is optionally possible to provide a path to a Comma Separated Values (CSV) file where some evaluation metrics and parameter settings can be stored for each prediction. The script then passes the test set to the model and predicts the interaction values for the test set. It then converts the prediction back to an actual Hi-C matrix. Since only a subset of the

Hi-C matrices is considered for testing according to the moving window approach, the remaining entries are set to zero. In the last step, the necessary evaluation metrics are calculated in comparison to the Hi-C values.

3.5 Evaluation Metrics

The following metrics are calculated to evaluate the accuracy of the prediction: R^2 score, Mean Squared Error (MSE), Mean Absolute Error (MAE) and Mean Squared Log Error (MSLE). Furthermore, metrics will be used as proposed by Zhang, Chasman, Knaack and Roy. The Pearson correlation stratified by genomic distance is computed, that means that for each genomic distance the correlation of predicted and true values is calculated separately. This relation can then be plotted to see how the prediction accuracy changes when the genomic distance increases or decreases.

$$\rho_d = \rho_{True_d, Predicted_d}$$

with $True_d$ and $Predicted_d$ as the true and predicted values of the subset of pairs that have the genomic distance d.

The calculated correlations can also be summarized by applying the Area under the curve calculation to get a scalar value for each prediction. This metric is denoted as Distance Stratified Pearson Correlation Area Under Curve (AUC) and was proposed by Zhang, Chasman, Knaack and Roy.

$$AUC = \int \rho_d$$

Furthermore, a distance stratified average matrix is calculated that denotes for each genomic distance the average read value of the original matrix. Correlation is then also calculated for the comparison between each bilateral combination of these three

matrices - the original, the predicted and the average matrix. Just as before Pearson as well as Spearman correlation is calculated.

3.6 Visual Evaluation

Evaluation metrics can detect overall changes in the accuracy or correlation, and compare methods to each other. But there has also to be a visual evaluation of the actual plots. It is necessary to compare the prediction with the correct matrices on a plot to compare recognizable structures. There are several approaches to do so. One of them is to just put the plots next to each other. Another reasonable approach is to subtract the predicted values from the correct values and plot the absolute values as a Hi-C matrix, so differences (prediction error) can be highlighted. It might be helpful to compare the resulting image to a similar approach where the average value at the specific genomic distance is subtracted from the original matrix. The genomes are very long and some structures like loops can not be seen well when displaying the whole chromosome. Therefore only subplots will be displayed.

4 Experiments

4.1 Default setting

The following setting is going to be used as a default setting. In both binning cases, the mean is used as a representative value. Furthermore, the centromeres are taken out, as this was helpful during the implementation phase. The binned protein peaks are not normalized and not manually manipulated, as both approaches did not seem to be helpful during earlier tests. The loss function will be Mean Squared Error and the read values will not be logarithmized. The "Signal Value" peak column of the narrowPeak format will be used.

4.2 Setup

The experiment will be conducted in the following way. Two cell lines of the human genome will be used, Gm12878 and K562. The resolution will be 5mb in all the experiments. The computational effort of creating the data sets, training on them and predicting is high because of the length of the genomes and the high resolution. Because of that, only the default setting will be used for the first part. With this setting, the models will be trained for the whole genome on Gm12878 and every model will be used to predict every chromosome, resulting in a total of 484 predictions. Afterwards, a subset of chromosomes (Chromosomes 7,9,14,18,19) will

be used exclusively in the remaining experiments. The first step will be to show that the evaluation metrics correlate with good and bad results on a visual basis and to investigate the relations between the metrics so a focus can be laid onto the subsets

In the following, the effects of each parameter and if it behaves as expected will be investigated. Finally, an experiment with two cell lines trying to predict the same chromosome used for training, but on the other cell line, will be executed. By doing so, it is expected to detect, which parameters might influence the predictions. The aim is also to investigate the overall ability of the Random Forest regressor to predict Hi-C matrices.

4.3 Default with all chromosomes

The first experiment is using the default setting and predicting all of the chromosomes with each of the 22 models trained on each chromosome.

Figure 2 shows the results for the default setting. On the x-axis, the chromosomes used for training the model are enlisted. On the y-axis the chromosomes that are tested. The evaluation metric that was chosen for this graphic is Distance Stratified Pearson Correlation AUC (just AUC in the following). The red values are indicating that these predictions are good, but this is not surprising since they are all showcases where the training and testing chromosome is the same. Since no training split was used, these values are not informative. But there are other interesting patterns to be seen. It is clear to see that some test chromosomes have tendencies of higher accuracy values independent of the model that was used to predict them (Chromosomes 9, 13, 14, 15, 21, 22). This is interesting to see since all of the chromosomes with acrocentric centromeres are part of this group. Chromosome 19, on the other hand, seems to be hard to predict and that is telling since predictions by the model trained on 19 are also really bad as well as these of chromosome 22. Models that do rather well have been trained on chromosomes 5, 6, 7 and 11 among others. For the next experiments,

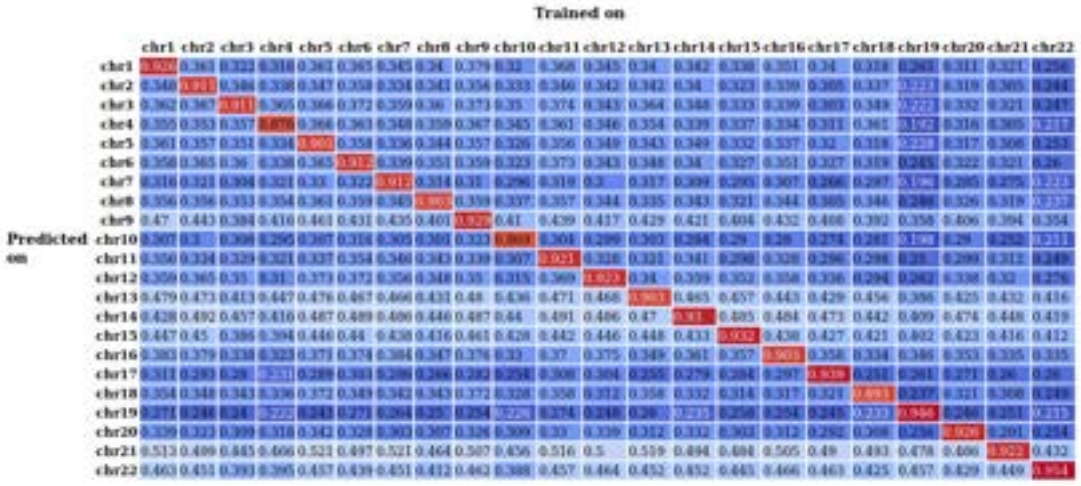


Figure 2: AUC for all chromosome model-prediction combinations

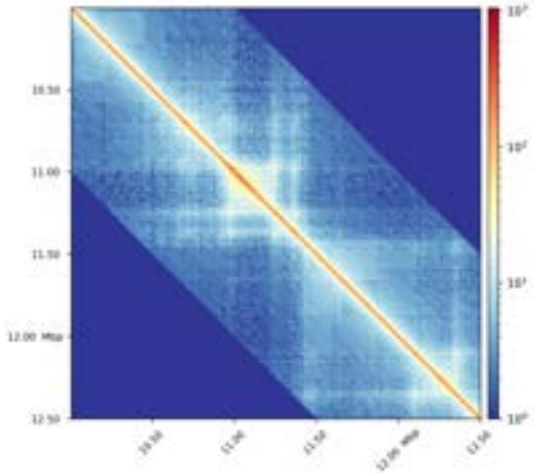
	R2	MSE	MAE	MSLE	AUC	Spearman	Pearson	Pearson_Difference	Spearman_Difference
R2	1.000000	-0.847515	-0.153255	-0.343372	0.116819	0.201068	0.944302	-0.016929	-0.117346
MSE	-0.847515	1.000000	0.242761	0.174734	-0.167068	-0.084610	-0.925853	-0.132425	-0.039876
MAE	-0.153255	0.242761	1.000000	0.817328	-0.634962	-0.513568	-0.077416	-0.261683	-0.315584
MSLE	-0.343372	0.174734	0.817328	1.000000	-0.536722	-0.650889	-0.205614	-0.070884	-0.128446
AUC	0.116819	-0.167068	-0.634962	-0.536722	1.000000	0.878343	0.089529	0.708169	0.766308
Spearman	0.201068	-0.084610	-0.513568	-0.650889	0.878343	1.000000	0.126631	0.637430	0.710872
Pearson	0.944302	-0.925853	-0.077416	-0.205614	0.089529	0.126631	1.000000	0.009904	-0.101330
Pearson_Difference	-0.016929	-0.132425	-0.261683	-0.070884	0.708169	0.637430	0.009904	1.000000	0.944617
Spearman_Difference	-0.117346	-0.039876	-0.315584	-0.128446	0.766308	0.710872	-0.101330	0.944617	1.000000

Figure 3: Correlations between Metrics

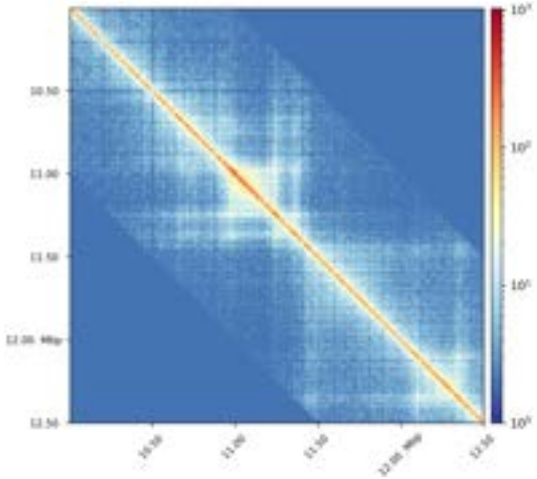
when trying to measure the effects of different settings, only the chromosomes 7, 9, 14, 18 and 19 will be regarded as representatives. Since there are a lot of possible metrics, it is necessary to compare them and to see which ones help and which are not apt to describe the accuracy of the predictions. Therefore the correlation between all of the metrics were calculated for this specific example. The metric used in Figure 2 (AUC) was proposed by Zhang, Chasman, Knaack and Roy [2018] and was also used in their analysis as the main indicator of a good prediction. Unfortunately, the model's built-in score function (R^2 score, 0.12, compare Figure 3) and the metric used during training (Mean Squared Error, -0.17) do not correlate with AUC. Mean Absolute

Error (MAE, -0.63) and Mean Squared Log Error (MSLE, -0.54) show much more correlation with the AUC metric. It might be interesting to change the loss function of the model to MAE instead of MSE since MAE shows a much higher correlation with AUC, a metric that is supposed to be indicative of the Hi-C predictions accuracy. One advantage of AUC is that it is distance stratified computing the correlation for each genomic distance. Thereby the high influence of the genomic distance is taken into account when evaluating the prediction. AUC also correlates highly with the Spearman correlation between the correct and the predicted values (0.88) but not with the Pearson correlation of the same (0.09). Additionally, the difference between the Pearson correlation of correct and predicted values and the Pearson correlation of correct and mean values were calculated. This metric is denoted as Pearson difference and its Spearman counterpart as Spearman difference. Both of these metrics show the improvement of the model's prediction in comparison to the simple average values. Both of them do also correlate highly with AUC, the Pearson difference correlates with 0.71 whereas the Spearman difference correlates with 0.77. This indicates that AUC is indeed a good measurement since it shows higher values when the prediction is better than the mere average. To show that the AUC metric is indicative of good predictions, it is necessary to have a look at the visual presentation of the predictions. Chromosome 7 will be used as an example. First of all the simple case of a Hi-C matrix predicted by a model trained on the same chromosome (chromosome 7, AUC=0.912, compare Figure 4) will be presented and then a supposedly average (model trained by chromosome 9, AUC=0.31, compare Figure 5) and a supposedly bad (model trained by chromosome 19, AUC=0.196, compare Figure 6) prediction.

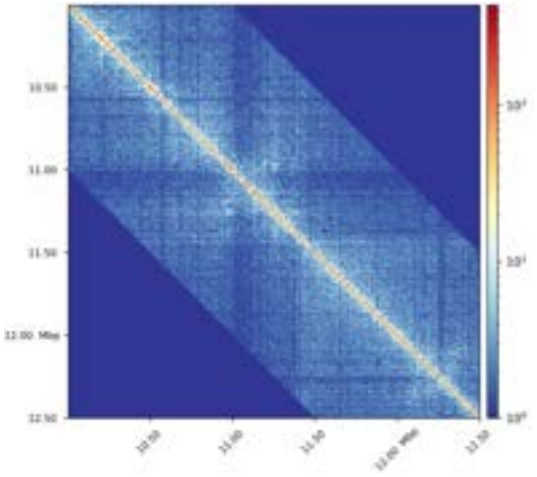
It is not surprising that the prediction made by training on the same chromosome (Figure 4) is good. It is just a blurry version of the original and the difference matrix confirms that the prediction is pretty close. But since that is testing on the training set, there is not much insight. What is interesting though is that the diagonal seems to be hard to predict even when predicting the same chromosome. And even some bigger scale structures are hard to reproduce in this setting. But there are good arguments



(a) Predicted Matrix

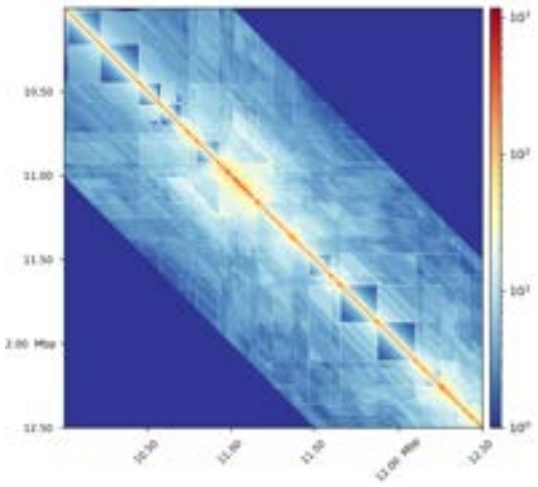


(b) Hi-C Matrix

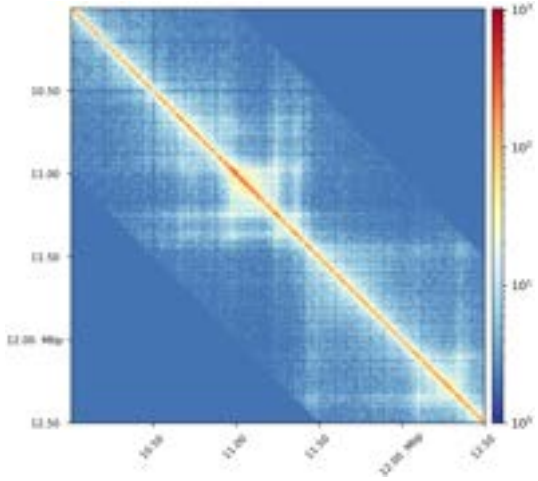


(c) Difference between true and predicted values

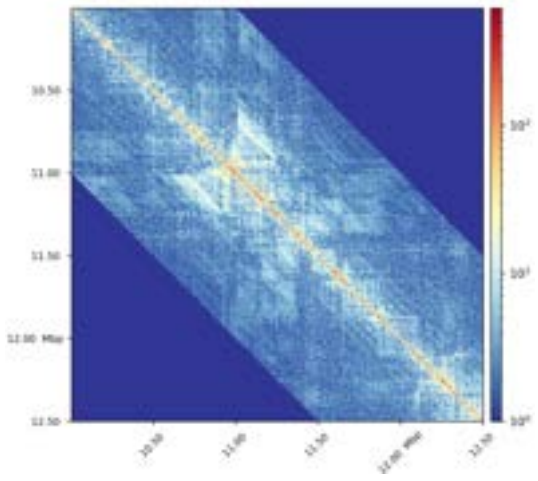
Figure 4: Extract of Chromosome 7 Predicted with 7



(a) Predicted Matrix

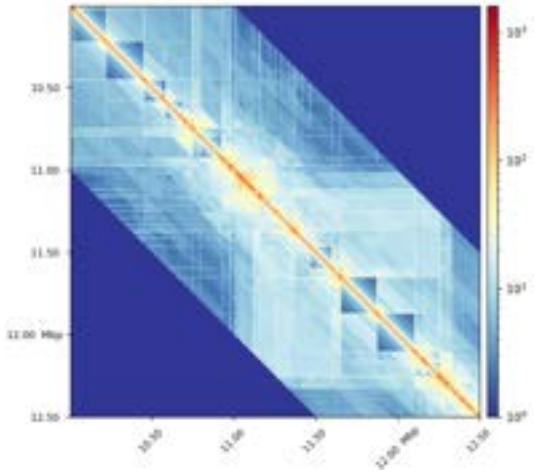


(b) Hi-C Matrix

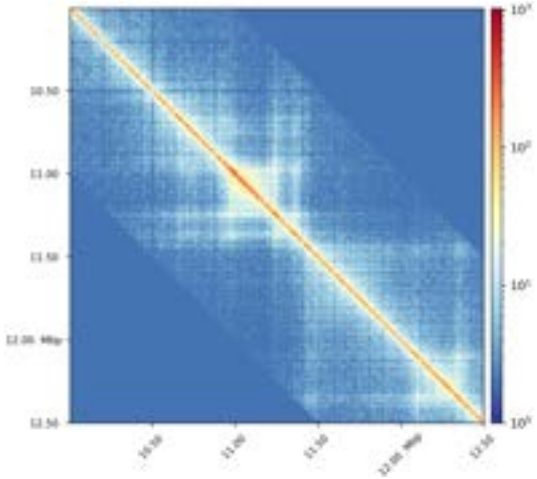


(c) Difference between true and predicted values

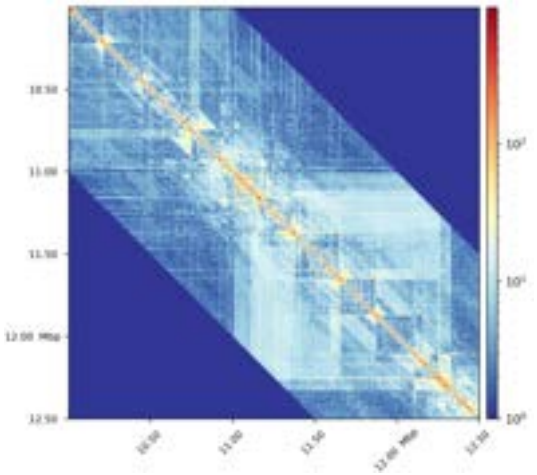
Figure 5: Extract of Chromosome 7 Predicted with 9



(a) Predicted Matrix



(b) Hi-C Matrix



(c) Difference between average and predicted values

Figure 6: Extract of Chromosome 7 Predicted with 19

to explain these observations. The closer the regions are in terms of genomic distance and especially in the case of comparing one region to itself, the harder it is to explain contacts by long-range loop structures, as many lower scale loops highly influence the reads. But the approach, using protein data as input focuses on these loop structures. Since the contacts in close regions have high numbers, there is also a lot of variance that is hard to predict with regression. The discreet big structures, on the other hand, suffer from the opposite. Those values are small and are hard to predict by any algorithm. Minor changes can make a huge difference in the prediction, also because of the logarithm that is applied to every value when plotting. The architecture of the whole framework also complicates recognizing these structures, since it focuses mainly on one specific point without taking the surrounding structures and values into account. The window approach (using proteins of regions in between) tries to account for that, but it seems that it is not enough and other ideas might need to be implemented to help predicting better in terms of structures. When predicting with models trained on other chromosomes (compare Figures 5 and 6)the prediction becomes naturally much worse. There are still structures recognized but there is also a lot of noise and structures where none are supposed to be. There is one repeating feature in the prediction that draws the attention. There are a few blue rectangles with a yellow frame like in Figure 7. This seems to happen when there are two regions with high protein peaks but no loop presented. While it is good to see that the regressor recognizes that there is no loop for most of the fields, it is still disturbing that it is framed as a rectangle at the positions of the peak regions. This often happens in the region between two TAD's in situations where a loop could present itself but is not there. But at least it can be shown that a higher AUC value indeed results in a better prediction, not only in the trivial case of predicting on the same chromosome but also in other cases. The prediction made by the model trained on 9 is much better and less noisy than the prediction made by a model trained on chromosome 19, just as indicated by the AUC values. This is recognizable when comparing the actual predictions, and even easier when comparing the difference

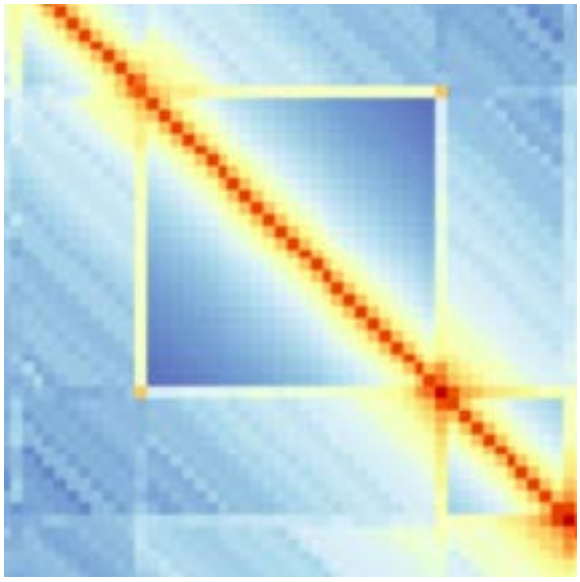
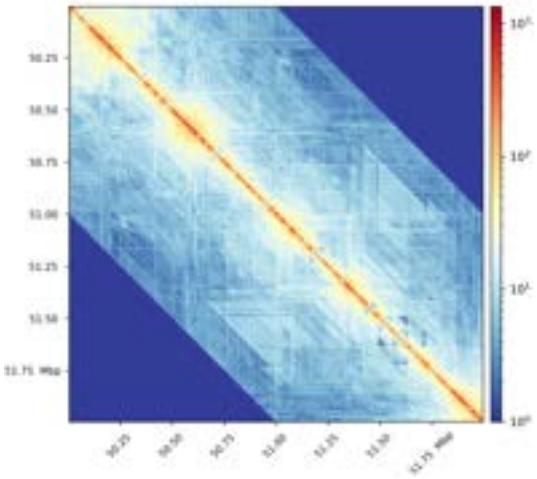


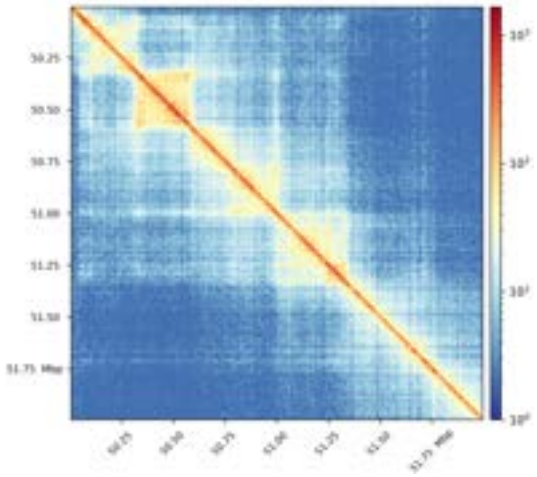
Figure 7: Example for interesting framing behavior

matrices. In the case of chromosome 19, a lot of structures are highlighted in the difference matrix, meaning that these structures have not been recognized by the prediction whereas the difference matrix of chromosome 9 is much smoother. But on the other hand, it still is not a good prediction and especially the areas close to the diagonal in the difference matrix highlight this observation. In the next step matrices with a higher AUC value will be compared. Chromosome 14 as predicted by chromosome 9 (compare Figure 8) has an AUC value of 0.487 whereas the same chromosome predicted by chromosome 19 (compare Figure 9) has an AUC value of 0.409. In general, all of the predictions of chromosome 14 have been good in terms of AUC as has been shown above.

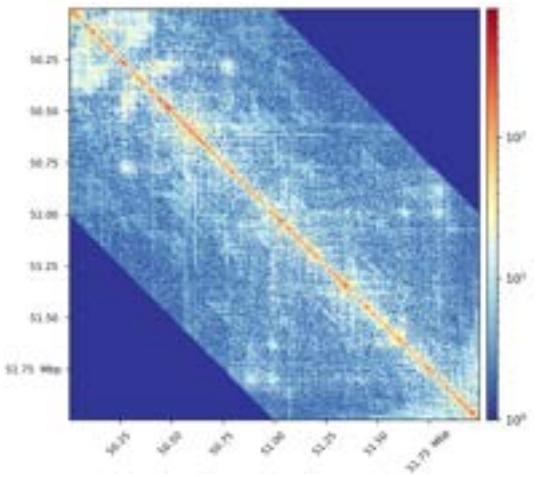
Again, by comparing the difference matrices it can be said that the higher AUC for predictions by the model trained on chromosome 9 indicates a slightly better prediction. But the predictions are bad in any case. It might, therefore, be hard to compare the accuracy of predictions over different chromosomes. This prediction is supposed to be much better than the first one. The differences in AUC between different predicted chromosomes do not seem to entail a lot of meaning. Some



(a) Predicted Matrix

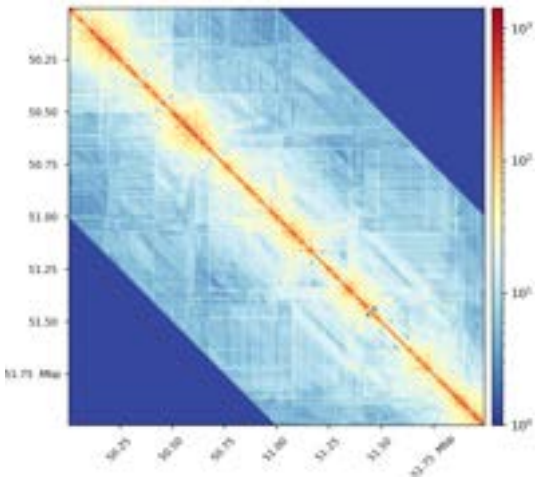


(b) Hi-C Matrix

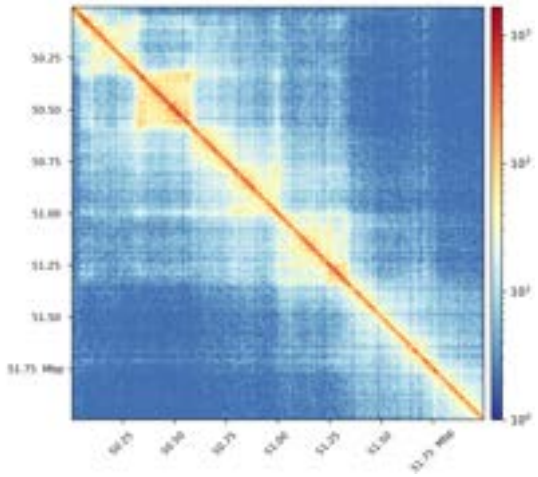


(c) Difference between true and predicted values

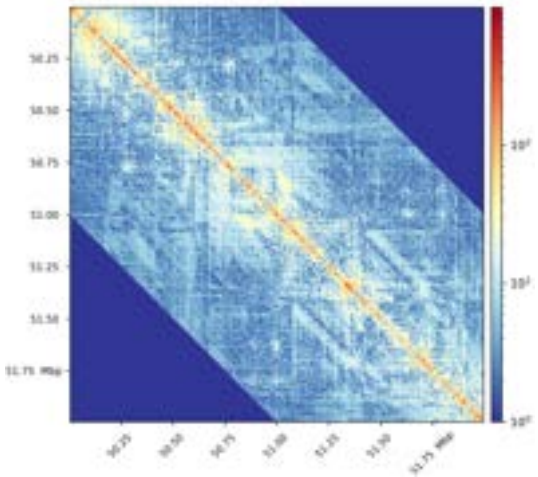
Figure 8: Extract of Chromosome 14 predicted with 9



(a) Predicted Matrix



(b) Hi-C Matrix



(c) Difference between average and predicted values

Figure 9: Extract of Chromosome 14 Predicted with 19

chromosomes might just be easier to predict, e.g. in terms of large non-loop regions and hereby produce high AUC values. But it seems like the AUC is at least able to compare the quality of predictions for the same chromosome. It is, therefore, possible to use the AUC value as an indicator for quality when comparing different settings. It also must be said and considered that it is not feasible to visually compare all of the regions and that these examples are just are a tiny subset of the whole matrix. It must be emphasized that comparing some of these samples visually can indicate a tendency but is not a substantial metric to measure the prediction quality. In the coming subsections, the effects of different settings will be measured.

4.4 Peak column

In an earlier version, the "Score" column of the narrowPeak format of the protein files was chosen. The approach still worked, but once it was changed, an improvement was noticed. To quantify this improvement all the AUC values for predictions based on the score column and based on the default signal value column were calculated. A subset of the aforementioned chromosomes is used. In 19 of 20 cases, the AUC was higher for the signal value column, averaging a difference of 0.05. This shows us that it is more efficient to use the signal value column that denotes the peak of the proteins instead of the score column that denotes a value for visualization of the peaks. Even though it is still a relevant value, using the actual peak is better(compare Figure 10).

4.5 Cutting out centromeres

The next parameter concerns the centromeres. If set, the centromeres of the chromosomes will be excluded from the training set since they mainly appear as big blurry regions in the visualization matrices. The same tests were conducted once with the

	modelChromosome	predictionChromosome	Pearson_difference_Score	Pearson_difference_Default	AUC_OP_P_Score	AUC_OP_P_Default	AUC_differenc
20	7	9	0.047789	0.064608	0.341534	0.434835	-0.09330
3	7	14	0.045683	0.058814	0.406433	0.485779	-0.07934
8	7	18	0.016410	0.022883	0.258528	0.341690	-0.08316
13	7	19	0.012966	0.023903	0.234308	0.264099	-0.02979
19	9	7	0.009402	0.017544	0.257095	0.309910	-0.05281
4	9	14	0.046798	0.064251	0.438771	0.486599	-0.04782
9	9	18	0.004951	0.008430	0.304970	0.372153	-0.06718
14	9	19	0.000539	0.019139	0.248706	0.254206	-0.00550
15	14	7	0.013080	0.016866	0.241497	0.308606	-0.06710
20	14	9	0.052968	0.072021	0.364427	0.420968	-0.05656
5	14	18	0.010039	0.011997	0.278647	0.332124	-0.05342
10	14	19	0.004781	0.020942	0.256138	0.234748	0.02139
16	18	7	0.019980	0.026577	0.259854	0.297013	-0.03715
21	18	9	0.045222	0.058692	0.342299	0.391931	-0.04963
1	18	14	0.045046	0.055624	0.405201	0.442097	-0.03689
11	18	19	0.013626	0.012148	0.205850	0.232899	-0.02684
17	19	7	0.017149	0.022944	0.164789	0.195518	-0.03072
22	19	9	0.042310	0.064085	0.309264	0.357922	-0.04865
2	19	14	0.042243	0.059279	0.358187	0.408677	-0.05049
7	19	18	0.014578	0.015749	0.191303	0.237367	-0.04606

Figure 10: Evaluation Metrics for Default Setting and Score Peak Column

centromeres excluded and once without any cutting. It turns out that the prediction gets worse when cutting the centromeres, averaging a difference of 0.018, again in 19 of the 20 test cases. This is not what was expected to happen, even though the difference is small. There is an aspect though that was not considered. Adding the centromere regions leads also to a slightly bigger test set. It might then be that the centromere regions, since they feature mostly the same values, are easily recognized and predicted and boost the accuracy in that way. Therefore this experiment was repeated with the same test sets to ensure the same conditions. That showed that the parameter does not have an important influence on the outcome. The predictions were on average slightly worse (0.002) when cutting the centromeres, but in 6 out of the 20 cases, the AUC was lower, when using the whole chromosome. This shows that the regressor can recognize the centromeres without harming its ability to predict the remaining chromosome.The differences in AUC values were so small, that there is no clear conclusion possible(compare Figure 11). The distance stratified plotting of

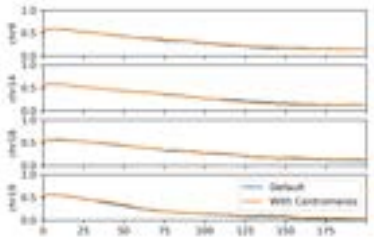
	modelChromosome	predictionChromosome	Pearson_difference_With	Pearson_difference_Default	AUC_OP_P_With	AUC_OP_P_Default	AUC_difference
23	7	9	0.065931	0.064606	0.447700	0.434835	0.012866
3	7	14	0.059842	0.059814	0.490602	0.485779	0.004823
8	7	18	0.022281	0.022883	0.348293	0.341690	0.006603
13	7	19	0.024008	0.023903	0.277579	0.264099	0.013480
19	9	7	0.014164	0.017044	0.318669	0.309910	0.008759
4	9	14	0.062687	0.064251	0.488163	0.486599	0.001564
9	9	18	0.005767	0.008430	0.354955	0.372153	-0.016098
14	9	19	0.017400	0.019139	0.268828	0.254206	0.014622
15	14	7	0.017366	0.019866	0.298363	0.308606	-0.010243
20	14	9	0.071777	0.072021	0.416325	0.420988	-0.004663
5	14	18	0.009383	0.011997	0.318034	0.332124	-0.014090
10	14	19	0.020410	0.020942	0.220415	0.234748	-0.014332
16	18	7	0.027033	0.026577	0.310464	0.297013	0.013452
21	18	9	0.060448	0.058692	0.396666	0.391931	0.004735
1	18	14	0.057547	0.055624	0.444337	0.442097	0.002239
11	18	19	0.011393	0.012148	0.226417	0.232699	-0.006282
17	19	7	0.022439	0.022944	0.212424	0.195518	0.016907
22	19	9	0.066042	0.064085	0.367036	0.357922	0.009114
2	19	14	0.060908	0.059279	0.417849	0.408677	0.009172
7	19	18	0.014548	0.015749	0.240509	0.237367	0.003142

Figure 11: Evaluation Metrics for Default Setting and Without Cutting Centromere

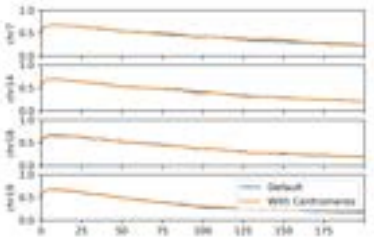
the correlation values confirms that, as there is no difference between the plots of the default setting and the ones where the centromere was included (compare Figure 12). Interesting is though, that for all the predictions made by the model trained on chromosome 14, the AUC was slightly higher, indicating again that the type of the centromeres influences the predictions.

4.6 Normalizing proteins

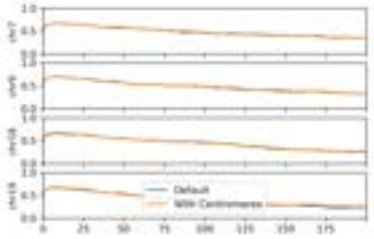
The next experiment was conducted with normalized proteins, which means that the binned proteins peaks were arranged to a 0-1 range. It is a common approach in Machine Learning to normalize the input data, and can have a big effect and it indeed has. Again the average difference between the AUC's was just 0.004 in favor of the default setting and 10 out of 20 test cases were better when normalized. But the



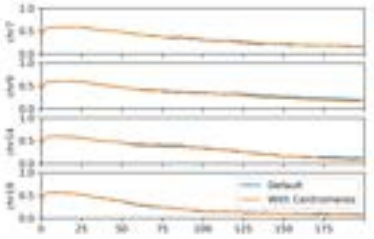
(a) Predictions by Chromosome 7



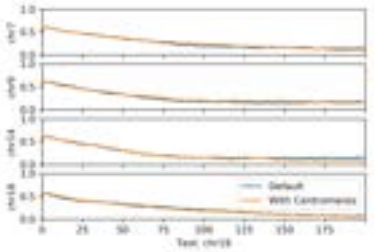
(b) Predictions by Chromosome 9



(c) Predictions by Chromosome 14



(d) Predictions by Chromosome 18



(e) Predictions by Chromosome 19

Figure 12: Comparison Distance Stratified Pearson Correlation between Default and Including Centromere

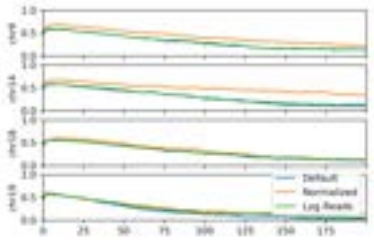
	modelChromosome	predictionChromosome	Pearson_difference_Norm	Pearson_difference_Default	AUC_OP_P_Norm	AUC_OP_P_Default	AUC_difference
23	7	9	0.016430	0.064608	0.305445	0.434835	-0.129390
3	7	14	0.019793	0.058814	0.301128	0.485779	-0.184651
8	7	18	0.026897	0.022883	0.291613	0.341690	-0.050078
13	7	19	0.023363	0.023903	0.206661	0.264099	-0.057437
19	9	7	0.064880	0.017044	0.425432	0.309910	0.115522
4	9	14	0.071776	0.064251	0.404512	0.486599	-0.082087
9	9	18	0.059145	0.006430	0.397927	0.372153	0.025774
14	9	19	0.064207	0.019139	0.348036	0.254206	0.093830
15	14	7	0.059023	0.019866	0.496030	0.308606	0.187424
20	14	9	0.063773	0.072021	0.481467	0.420988	0.060479
5	14	18	0.056297	0.011997	0.450691	0.332124	0.118567
10	14	19	0.059556	0.020942	0.405057	0.234748	0.170309
16	18	7	0.023071	0.026577	0.325997	0.297013	0.028984
21	18	9	0.008281	0.056692	0.354432	0.391931	-0.037499
1	18	14	0.011509	0.055824	0.326380	0.442097	-0.115718
11	18	19	0.015504	0.012148	0.202309	0.232699	-0.030391
17	19	7	0.023582	0.022944	0.265253	0.195518	0.069735
22	19	9	0.019330	0.064085	0.264445	0.357922	-0.093477
2	19	14	0.020844	0.059279	0.226276	0.408677	-0.182401
7	19	18	0.015112	0.015749	0.243557	0.237367	0.006190

Figure 13: Evaluation Metrics for Default Setting and Normalizing Proteins

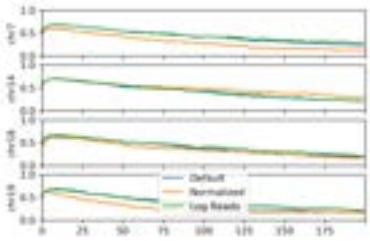
margins on the particular predictions are huge with outliers in both directions. This needs to be evaluated far more to make a coherent conclusion. Since it is a better practice to use normalized proteins, that is the setting that is recommended for now, but the AUC did not indicate which setting might be better overall (compare Figure 13). The distance stratified plotting of the correlation values confirms that, as there are huge difference between the plots of the default setting and the ones where the proteins were normalized was included (compare Figure 14), but there are both cases of the normalized setting being much better and much worse.

4.7 Convert reads with logarithm

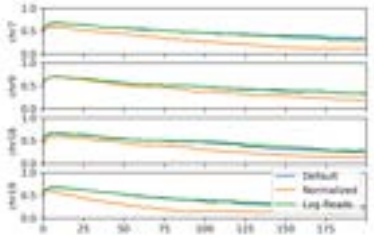
In this approach, the idea is to apply the logarithm to the read values to have smoother target values. Since the entries close to the diagonal tend to be big whereas



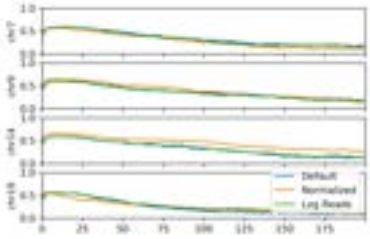
(a) Predictions by Chromosome 7



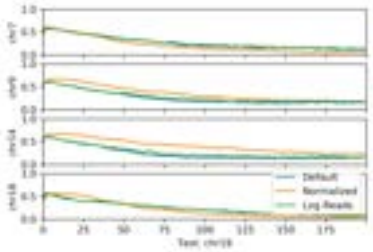
(b) Predictions by Chromosome 9



(c) Predictions by Chromosome 14



(d) Predictions by Chromosome 18



(e) Predictions by Chromosome 19

Figure 14: Comparison Distance Stratified Pearson Correlation between Default and Log and Normalizing

	modelChromosome	predictionChromosome	Pearson_difference_Log	Pearson_difference_Default	AUC_OP_P_Log	AUC_OP_P_Default	AUC_difference
23	7	9	0.068823	0.064608	0.451720	0.434835	0.016885
3	7	14	0.061309	0.058814	0.471663	0.485779	-0.014116
8	7	18	0.015734	0.022683	0.333036	0.341690	-0.008654
13	7	19	0.019009	0.023903	0.272944	0.264099	0.008846
19	9	7	-0.110512	0.017044	0.311701	0.309910	0.001791
4	9	14	-0.020396	0.064251	0.492372	0.486599	0.005773
9	9	18	-0.143651	0.008430	0.351813	0.372153	-0.020340
14	9	19	-0.023648	0.019139	0.280848	0.254206	0.026640
15	14	7	-0.095774	0.019866	0.302961	0.308606	-0.005645
20	14	9	0.002359	0.072021	0.432325	0.420988	0.011337
5	14	18	-0.123634	0.011997	0.337575	0.332124	0.005451
10	14	19	-0.017807	0.020942	0.264976	0.234748	0.030228
16	18	7	0.019530	0.026577	0.302721	0.297013	0.005709
21	18	9	0.066848	0.058692	0.421292	0.391931	0.029361
1	18	14	0.061348	0.055624	0.461591	0.442097	0.019494
11	18	19	0.016024	0.012148	0.234556	0.232699	0.001857
17	19	7	-0.044571	0.022944	0.231719	0.195518	0.036201
22	19	9	0.040238	0.064085	0.408039	0.357922	0.050117
2	19	14	0.028226	0.059279	0.414198	0.408677	0.005521
7	19	18	-0.064279	0.015749	0.271106	0.237367	0.033739

Figure 15: Evaluation Metrics for Default Setting and Log Conversion

values in the outer ranges are small, the algorithm would focus on predicting the diagonal entries since much more can be gained by reducing the error in terms of the loss function. Applying the logarithm is supposed to change that. This gets confirmed as the average difference (0.01) shows that the log setting is slightly better (compare Figure 15). The distance stratified plotting of the correlation values confirms this finding, as there is again clear evidence that the plots of the default setting are slightly worse than the ones where the reads were converted with the log function (compare Figure 14).

4.8 Binning proteins with max function

In this part of the experiments, the effects of the max function as a binning function is tested. Just as in some of the other cases before, it did not have a big influence on the prediction accuracy and was slightly worse (average difference is 0.003). That might

	modelChromosome	predictionChromosome	Pearson_difference_Max	Pearson_difference_Default	AUC_OP_P_Max	AUC_OP_P_Default	AUC_difference
23	7	9	0.065149	0.064608	0.432245	0.434835	-0.002590
3	7	14	0.060087	0.058814	0.487284	0.485779	0.001505
6	7	18	0.023000	0.022883	0.345229	0.341690	0.003539
13	7	19	0.027055	0.023903	0.271903	0.264099	0.007804
10	9	7	0.017817	0.017044	0.322204	0.309910	0.012293
4	9	14	0.064749	0.064251	0.489317	0.486599	0.002718
8	9	18	0.009468	0.008430	0.359082	0.372153	-0.013071
14	9	19	0.022179	0.019139	0.256147	0.254206	0.001941
15	14	7	0.020480	0.019888	0.312606	0.308606	0.004000
20	14	9	0.072650	0.072021	0.407063	0.420968	-0.013085
5	14	18	0.012241	0.011987	0.322113	0.332124	-0.010013
12	14	19	0.023915	0.020942	0.227772	0.234748	-0.006976
16	18	7	0.026753	0.026577	0.303027	0.297013	0.006014
21	18	9	0.059217	0.058692	0.381609	0.391931	-0.010321
1	18	14	0.055860	0.055624	0.424360	0.442097	-0.017747
11	18	19	0.014366	0.012148	0.248529	0.232699	0.015830
17	19	7	0.024747	0.022944	0.218015	0.195518	0.022497
22	19	9	0.065057	0.064085	0.359408	0.357922	0.001486
2	19	14	0.061019	0.059279	0.405772	0.408677	-0.002906
7	19	18	0.017139	0.015749	0.227637	0.237367	-0.009730

Figure 16: Evaluation Metrics for Default Setting and Max Binning

be especially true for small bins like in this particular experiment. With a resolution of 5kb there is often just one protein peak or none at all. A max function would have more effect for more populated bins in the case of higher resolutions (compare Figure 16). The distance stratified plotting of the correlation values confirms that, as there is no difference between the plots of the default setting and binning with the max function(compare Figure 18).

4.9 Window approach with sum function

The next part investigates the effects when using the sum function for the window binning. This might help to emphasize relationships dependent on genomic distance, but normally it should not have a big positive effect since the sum can be defined by average and genomic distance, both features are given as input data in the default setting. The results of the experiment show that the predictions get much worse

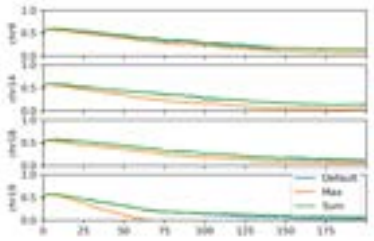
	modelChromosome	predictionChromosome	Pearson_difference_Sum	Pearson_difference_Default	AUC_OP_P_Sum	AUC_OP_P_Default	AUC_difference
23	7	9	0.063256	0.064808	0.295955	0.434835	-0.138880
3	7	14	0.058644	0.058814	0.412003	0.485779	-0.073776
8	7	18	0.022138	0.022883	0.256495	0.341690	-0.085196
13	7	19	0.025576	0.023903	0.232408	0.264099	-0.031690
19	9	7	0.015270	0.017044	0.258512	0.309910	-0.051398
4	9	14	0.063193	0.064251	0.441674	0.486599	-0.044925
9	9	18	0.006382	0.008430	0.283630	0.372153	-0.088523
14	9	19	0.019878	0.019139	0.214552	0.254206	-0.039653
15	14	7	0.018801	0.019886	0.209861	0.308606	-0.098745
20	14	9	0.067313	0.072021	0.237066	0.420988	-0.183921
5	14	18	0.008517	0.011997	0.225588	0.332124	-0.106536
10	14	19	0.020363	0.020942	0.136625	0.234748	-0.098123
16	18	7	0.025069	0.026577	0.228208	0.297013	-0.068805
21	18	9	0.055915	0.058692	0.228255	0.391931	-0.163675
1	18	14	0.055609	0.055624	0.410698	0.442097	-0.031400
11	18	19	0.017980	0.012148	0.119895	0.232699	-0.112804
17	19	7	0.013941	0.022944	0.048725	0.195518	-0.146793
22	19	9	0.056042	0.064085	0.213315	0.357922	-0.144607
2	19	14	0.053660	0.059279	0.269754	0.408677	-0.138923
7	19	18	0.005808	0.015749	0.096493	0.237367	-0.140874

Figure 17: Evaluation Metrics for Default Setting and Summed Window

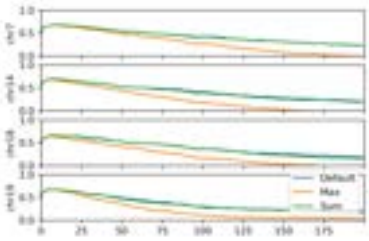
when summing the protein peaks instead of taking the mean. It got worse for all of the test cases with an average difference of 0.1 which shows that it is much better to take the average (compare Figure 17). There might still be other approaches that would be worth trying like the max function. The distance stratified plotting of the correlation values confirms that the sum function has a negative effect, as these plots are much worse than the ones of the default setting(compare Figure 18).

4.10 Cross cell line predictions

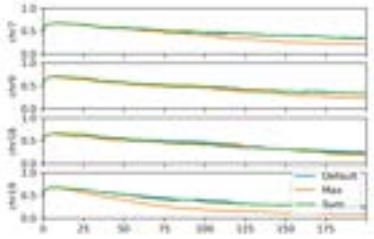
Research has shown that it is more promising to predict the same chromosome but learn on another cell line. This approach is presented in this section. Again the default setting was used. Each chromosome on cell line K562 was predicted by models trained on the same chromosome of cell line Gm12878. Additionally, each chromosome of K562 was predicted by models of all the other chromosomes, resulting



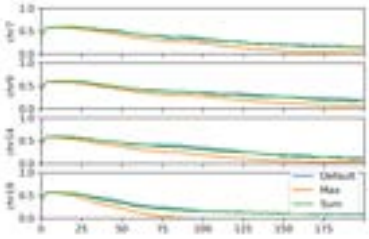
(a) Predictions by Chromosome 7



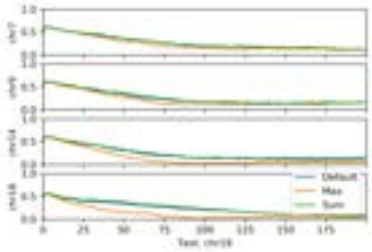
(b) Predictions by Chromosome 9



(c) Predictions by Chromosome 14



(d) Predictions by Chromosome 18



(e) Predictions by Chromosome 19

Figure 18: Comparison Distance Stratified Pearson Correlation between Default and Sum and Max Binning

	predictionChromosome	(AUC, min)	(AUC, max)	(AUC, mean)	AUC_ByGm12878
0	chr1	0.209265	0.346272	0.311336	0.349048
1	chr10	0.175778	0.260191	0.231557	0.223864
2	chr11	0.198516	0.304561	0.273304	0.299445
3	chr12	0.199307	0.320251	0.279417	0.290897
4	chr13	0.254096	0.340322	0.312412	0.317143
5	chr14	0.265971	0.372529	0.338153	0.372592
6	chr15	0.228059	0.379113	0.344503	0.369533
7	chr16	0.219426	0.334553	0.293341	0.318045
8	chr17	0.152551	0.245138	0.210978	0.222669
9	chr18	0.211520	0.318410	0.264256	0.266495
10	chr19	0.143803	0.231740	0.191832	0.209920
11	chr2	0.178732	0.275427	0.249737	0.271037
12	chr20	0.156144	0.327247	0.277355	0.268279
13	chr21	0.320966	0.498644	0.457938	0.477846
14	chr22	0.275867	0.522197	0.414835	0.346659
15	chr3	0.231686	0.322167	0.299172	0.305831
16	chr4	0.209710	0.309974	0.281158	0.297666
17	chr5	0.189966	0.281034	0.252902	0.277965
18	chr6	0.225504	0.346355	0.317566	0.354989
19	chr7	0.201406	0.294904	0.265968	0.273550
20	chr8	0.185071	0.310624	0.271961	0.292121
21	chr9	0.345490	0.443221	0.381290	0.395038

Figure 19: AUC for K562 Cross Cell Line Predictions

in 462 combinations. For each predicted chromosome the AUC of the worst and the best prediction as well as the mean is shown along with the AUC of the prediction made by the other cell line (Gm12878). The results show that the prediction across cell lines is in many cases close to the best of the 21 predictions that were conducted with chromosomes from the same cell line. This confirms that it is promising to learn and predict the same chromosome across cell lines.

5 Conclusion

Overall it can be said, that there are clear indications that predicting Hi-C matrices is possible, but the predictions are just not good at the moment. They recognize some structures, tend to have high values around the TAD's but lack of consistency and especially of showing the well-defined structures Hi-C matrices have.The different settings and options that were used so far, do not have a great (positive) influence. The best parameters only change the predictions by a small margin while most alternative settings do not cause any noticeable change or even decrease the accuracy. There are many different options on how to proceed. In a first step it might important to create a coherent metric that has clear ability to recognize good predictions. That might include a comparison of TAD's and compartments. It might be a good idea to focus on 2 chromosomes and try different approaches. During this project, it was often hard to evaluate the results since the AUC value is just one value and not necessarily the best metric. Comparing the images by visualizing them is helpful in the beginning as the human eye is very good at comparing structures, but it is not feasible in the long run due to the length of the chromosomes. Maybe it is possible to develop a good mathematical metric and test it just on a small subset of the chromosome and validate it by comparing the images. That will help in the long run to make clear distinctions between the quality of the predictions and choosing good parameters and settings with coherent and academic decision making.

A very promising result was the cross cell line prediction. It might not always be close to the best prediction, but it is constantly good. There might be also be many ways to

adapt the input data or general setup to apply better for cross cell line predictions, as the current predictions might suffer because of that. It also enables more experiments as one can focus on a single chromosome per cell line instead of several chromosomes. This would make the experiments much easier and more settings could be explored.
One of the greatest problems when trying to predict Hi-C matrices is the amount of data in general and the scarcity of input data for many entries. A lot of regions do not have any peak for most of their proteins or sometimes even none at all. Another problem is that most of the contacts are not caused by loops between the two regions we look at but by surrounding structures of other pairs. These have a high influence on the read values. The window approach tries to account for that by calculating representative values for the regions in between. But it does not account for surrounding and neighboring regions outside of the two regions. But these areas might be especially interesting since the regions in the focus could lie in a bigger loop. A straight forward approach to account for that behavior would be to extend the window approach to neighboring areas. It might be hard to define a reasonable area and just as with the window approach, another problem would be faced. Howis it possible to account for different genomic distances when applying the window approach? At the moment just one representative value is chosen. But it might be helpful to make a distinction on how to summarize the regions in the middle, when sometimes the distance is just 5kb and in other cases 2mb.
A new idea is to group the entries by genomic distance and train different models for all of them, but that would lead to a lot of different models and it seems rather unfeasible. But it is evident that the surrounding of the regions of interest are important for the predictions and it should be helpful to think about ways how to include more information.
Another approach heading into the same direction might be to predict from the top to the bottom. It could be possible to start predicting the entries with the highest genomic distance and use these predictions plus the proteins to predict the next level and so on. This might help to account for the influence long-scale loops have to

smaller range predictions. A similar approach would try to directly predict TAD's and compartments and possibly use this predicted information to have more information as input data for the Hi-C read predictions. All of these ideas have one idea in common, to exploit information of underlying structures and their influence for each region pair in the matrix, instead of limiting the data to the two regions. Interestingly that Zhang, Chasman, Knaack and Roy showed that the window approach improved their results, even though it only takes into account regions in between, which means that these regions can form no loop that frames the specific intersection of interest. The impact of using information about structures outside of the two regions must be even higher.

This might also help to deal with the general lack of structural prediction ability of the current regressor. Whereas it mostly recognizes regions of interest, it fails on displaying coherent and sharp structures. It is supposed that this is caused by focusing on every single intersection without taking the bigger image into account.

A completely different approach to improve the results might be to try out hyper parameter training or to use other regressors or even Neural Networks. This might certainly help, but the bigger issues are supposed to be the ones mentioned before. It is much more promising to focus on adding structural information to the input data.

Concluding it can be said that there are many approaches that can be tried in the future. The potential of predicting Hi-C matrices is certainly there, but new ways must be found to unleash it.

Bibliography

Cristescu, Borsos, Lygeros, Rodríguez Martínez and Rapsomaniki. Inference of the three-dimensional chromatin structure and its temporal behavior. *arXiv e-prints*, art. arXiv:1811.09619, Nov 2018.

R. Karlić, H.-R. Chung, J. Lasserre, K. Vlahoviček, and M. Vingron. Histone modification levels are predictive for gene expression. *Proceedings of the National Academy of Sciences*, 107(7):2926–2931, 2010. doi: 10.1073/pnas.0909344107. URL `https://www.pnas.org/content/107/7/2926`.

Lajoie, Dekker, Kaplan. The hitchhiker's guide to hi-c analysis: Practical guidelines. *Methods (San Diego, Calif.)*, 72:65–75, 2015.

X. Lan, H. Witt, K. Katsumura, Z. Ye, Q. Wang, E. H. Bresnick, P. J. Farnham, and V. X. Jin. Integration of Hi-C and ChIP-seq data reveals distinct types of chromatin linkages. *Nucleic Acids Research*, 40(16):7690–7704, 06 2012.

E. Lieberman-Aiden, N. L van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. Lajoie, P. Sabo, M. Dorschner, R. Sandstrom, B. Bernstein, M. Bender, M. Groudine, A. Gnirke, J. A. Stamatoyannopoulos, L. Mirny, E. S Lander, and J. Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science (New York, N.Y.)*, 326: 289–93, 10 2009. doi: 10.1126/science.1181369.

Nagano, Lubling, Stevens, Schoenfelder, Yaffe, Dean, Laue, Tanay and Fraser. Single-

cell hi-c reveals cell-to-cell variability in chromosome structure. *Nature*, 502, 09 2013. doi: 10.1038/nature12593.

S. Tagami, S.-i. Sekine, K. Thirumananseri, N. Hino, Y. Murayama, S. Kamegamori, M. Yamamoto, K. Sakamoto, and S. Yokoyama. Crystal structure of bacterial rna polymerase bound with a transcription inhibitor protein. *Nature*, 468:978–82, 12 2010. doi: 10.1038/nature09573.

Zhang, Chasman, Knaack and Roy. In silico prediction of high-resolution hi-c interaction matrices. *bioRxiv*, 2018. doi: 10.1101/406322. URL `https://www.biorxiv.org/content/early/2018/09/01/406322`.