Stanford University

GENE 245

Project Report

Converting manual machine learning to machine machine learning for chromosome conformation patterns

Author:
Dan Iter
Rob Bierman
Lan Huong Nguyen

Mentor:
Oana Ursu

June 13, 2016

1 Introduction

Human cells contain approximately three billion bases of DNA, which would measure three meters in length if laid in a line [14]. As these cells are only on the order of 10 μm in size, significant folding and compaction of the DNA is necessary. There are many levels of DNA organization, ranging from unorganized open regions called euchromatin, to compact heterocrhomatin sections wrapped around histones, all the way up to the most condensed chromosomal form [14]. A common misconception is that DNA exists in tight x-shaped chromosomes throughout the entirety of the cell cycle, when these bundles are actually only formed during cell division. In most of a cell's life its DNA is in a semi-compacted state within the nucleus. Some regions are tightly wound and inaccessible, while others are open. Perhaps even more surprising is the dynamic nature of DNA, where the accessibility of different portions is constantly changing depending on the state of the cell [6].

DNA compaction serves as a mechanism to both address space limitation, and also to regulate which genes are turned off and on. Genes within tight heterochromatin are inaccessible to RNA polymerase and transcription factors, and cannot be turned into an active protein. Euchromatin, on the other hand, is more actively expressed. Adding additional complexity, certain DNA elements called enhancers can regulate transcription of genes millions of bases away on the same or different chromosomes. In order for a cell to control which genes it is expressing at a certain time, it must therefore carefully fold it's entire chromosome to ensure appropriate interactions occur. Consequently, researchers have long desired to start with chromosome conformation data and work backwards to understand the state of a cell.

Understanding the folding interactions of chromosomes is problematic because of the shear length of DNA and difficulty to capture these folds in physiological conditions. Some of the first approaches to investigate DNA structure were based on microscopy and molecular probes [13]. Examples of such methods are Fluorescent In Situ Hybridization (FISH) and Fluorescence Resonance Energy Transfer (FRET). While these techniques are useful, they can only be used to observe a few specific loci at a time, and cannot easily be scaled up. With the increasing ease of sequencing, however, methods were produced to yield full chromosomal conformation maps.

As stated above, the question of how DNA folds from invisible meter length strands to tight observable chromosomes, has long been of interest. Chromatin conformation capture (3C) is a relatively new technique used to describe the shape of compact DNA with massive throughput. The original high-throughput method was published by Dekker et. al in a highly-cited 2002 paper [3]. The essence of the technique is that a population of cells are suddenly fixed with formaldehyde or an alternate cross linker, effectively taking a snapshot of which pieces of chromosome are neighbors [6]. Then through the clever use of restriction enzymes, the pieces of nearby DNA are joined into a single strand that is able to be sequenced. If two sequences not normally adjacent are read out, then a count is added to the interaction matrix between these two regions. Note that 3C requires that the chromosome sequence is known, and the technique cannot be used in areas of highly repetitive DNA. It is not possible to confidently map where the interaction took place in such regions. Additional extensions to 3C, such as 4C, 5C, HiC, and ChIA-PET have since been produced and have varying strengths and weaknesses [5]. The applications of these tools have helped describe the biology of chromatin conformation on incredible scales [9][12].

As with all large-data producing techniques, the analysis of 3C results is often difficult and conclusions can vary with different approaches [7]. Furthermore, quite a bit of human inter-

action with the data is still necessary to find even simple patterns, such as domains, with high confidence [4]. In this work we apply machine learning as a conducive method towards identifying previously unstudied patterns in chromosome interaction data sets. We first use supervised learning to show that patterns identified by a user can be learned by tensor flow models, and then transition into unsupervised methods to delve even more deeply into the possibilities of discovery without human intervention.

2 Supervised Learning Shape Classes

The goal of this work is to both find novel chromosomal interaction patterns as well as reliably identify known shapes. To do so we started with human chromosome interaction data, in bedpe format, and slid a megabase sized square down the diagonal to create square frames reflected over the $y = x$ axis. This diagonal line represents chromosomal sites interacting with themselves and such interactions are nearly always observed [cite]. This simplification of the data is warranted as most interactions and interesting behavior are close range. The following figures have only the upper triangular data to simplify viewing. The colors of the images represent the log of the frequency of observed data interactions divided by the expected number of interactions based on the distance between the two loci.

$$value = log\left(\frac{obs}{exp}\right)$$

Therefore a value of 2 indicates that there are 100 times more observed interactions than expected, while -2 indicates 100 times fewer. In order to cluster these different megabase squares of DNA interaction data, two different classification systems with six different descriptive classes were created.

Loop-Domain Based Classification

The first classification system is termed Loop-Domain Based Classification. The membership of an image to a group is determined by the presence and strength of well described domains and loops [cite]. Domains are displayed on chromosome interaction plots as red rectangles bisected by the diagonal. They represent regions of the chromosome within which loci associate with each other more than expected. Loops are easily identifiable as small high intensity circular peaks off the diagonal and often times at the corner of a domain. These loops represent two specific portions of a chromosome that are found colocalized frequently and could be held in such conformations by DNA binding proteins [cite]. Loops are also indicative of enhancers "looping out" to promoter regions to effect gene expression [cite].

The first class is simply defined as the "information loss" frames, where errors in sequencing, or regions of high sequence repetition prevent the counting of interactions [cite]. The ends of the chromosome and the centromere often fall into this category. Examples of images that have information loss are given in Figure 1. Notice the large swaths of missing data.

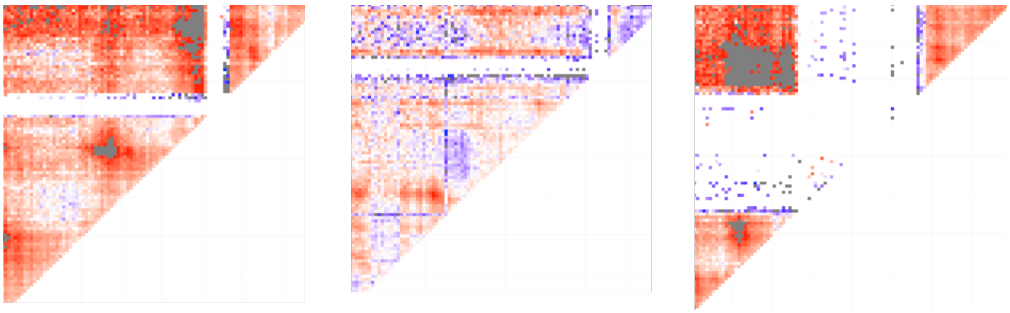


Figure 1: Example regions given the class 0, "information loss," label.

The second class in this system is called the "disperse" images and depict regions that don't seem to have either domains or loops. It is extremely difficult in the left image in Figure 2 to identify characteristic domain triangles or sharp loop peaks. The right image in Figure 2 arguably has two large domains, but these are lightly colored meaning that the observed amount of interactions is close to the expected.

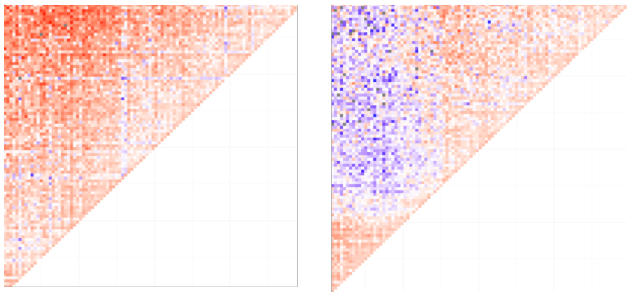


Figure 2: Example regions given the class 1, "disperse," label.

The third class is composed of frames that have obvious domains, but no loops. In Figure 3 the single image shows the obvious red right triangles with their hypotenuses on the diagonal, but there are no discrete strong loops.

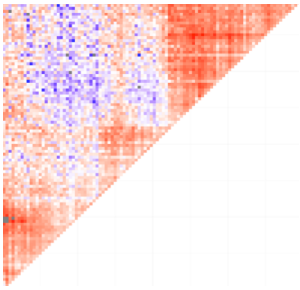


Figure 3: Example region given the class 2, "domain only," label.

Fourth are the frames containing numerous multiple weak loops. Since our frames span a million bases, it is unlikely to find just a single loop, and this was checked empirically. These loops are deemed "weak" since they are very small circles of bright red. The presence of domains is not taken into consideration if the weak loops are present. Examples are shown in Figure 4.

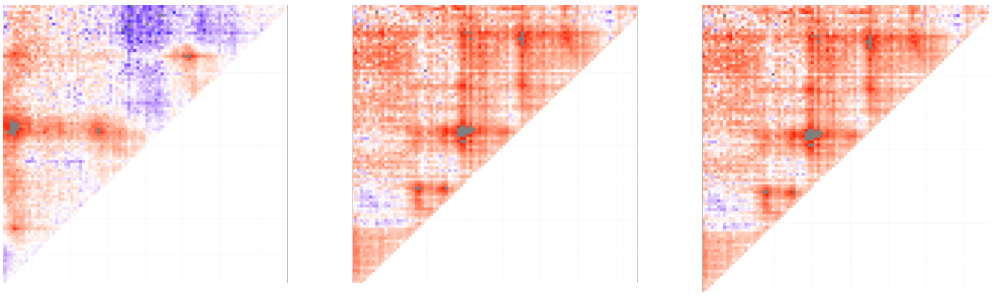


Figure 4: Example regions given the class 3, "multi-weak loops," label.

The next group is nearly identical to the previous, except the loops are better defined with very large bright circles off the diagonal. Physiologically, class 3 and class 4 may differ as to the mechanisms that tether their interactions. As a human classifier, these descriptive similar classes were often difficult to address.

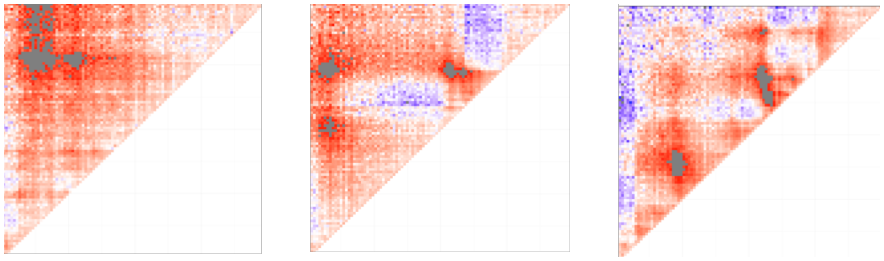


Figure 5: Example regions given the class 4, "multi-strong loops," label.

The final group in the Loop-Domain based classification is termed the "few strong loops", which is again similar to the previous two classes. The images in this class, Figure 6, are perhaps difficult to distinguish from the images in class 4, Figure 5, but might be biologically important.

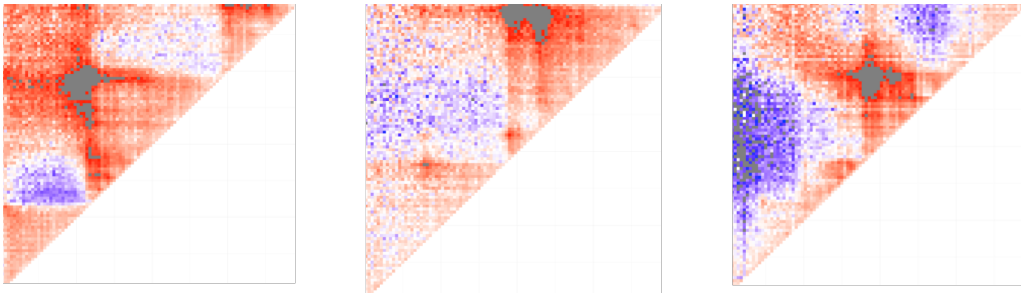


Figure 6: Example regions given the class 5, "few strong loops," label.

Negative Space Based Classification

In addition to creating a set of six "biologically inspired" classes above, we were interested to see how well our supervised learner would perform on an easier set of classes. The negative space based classifications are larger, more distinct patterns that are expected to be more obvious to a tensor flow model. Additionally, this classification system is interesting because patterns have nearly always been previously defined by the red positive space rather than the blue negative space.

Again the first class is the grouping of frames missing information. It was considered important to separately identify the missing data as the resulting shapes could be erroneous.

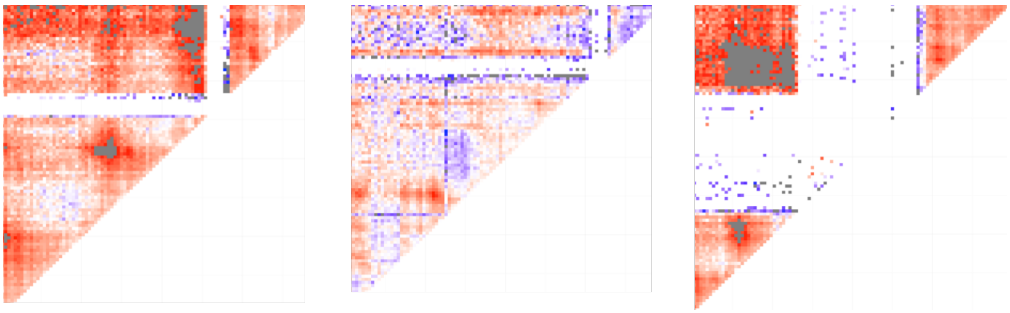


Figure 7: Example regions given the class 0, "information loss," label.

The second class is defined by having blue on the edges of the frame as depicted in Figure 8. This class was interesting because it implicitly defines large scale domains which take up the middle of the frame, but does not concern itself with the substructure.

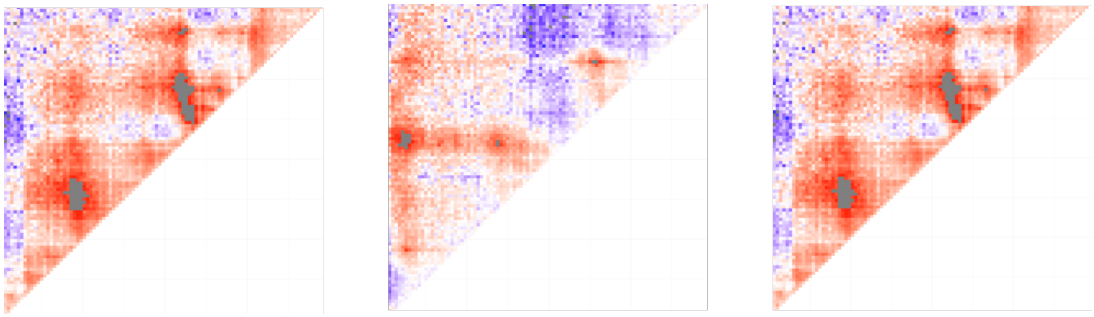


Figure 8: Example region given the class 1, "fringe blue," label.

The third class in the negative space is defined by blue pockets dispersed throughout the image. Note that in the frames shown in Figure 9 the blue pockmarks fit nicely between putative domains according to the above classification system.

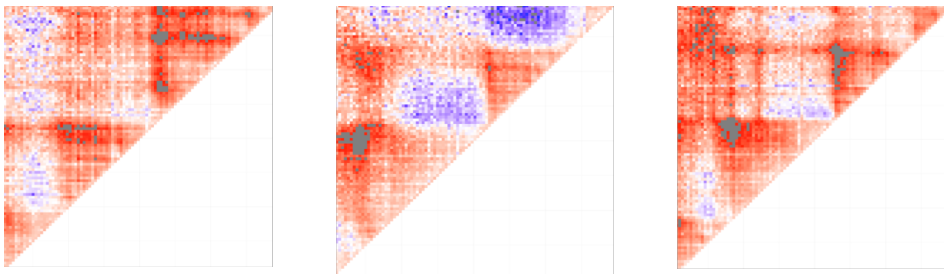


Figure 9: Example regions given the class 2, "blue pockets," label.

After randomly viewing many frames, it was surprising to find a large amount that had fewer diagonal interactions than expected and thus had blue on the diagonal. There are many ways for this to occur as shown in Figure 10.

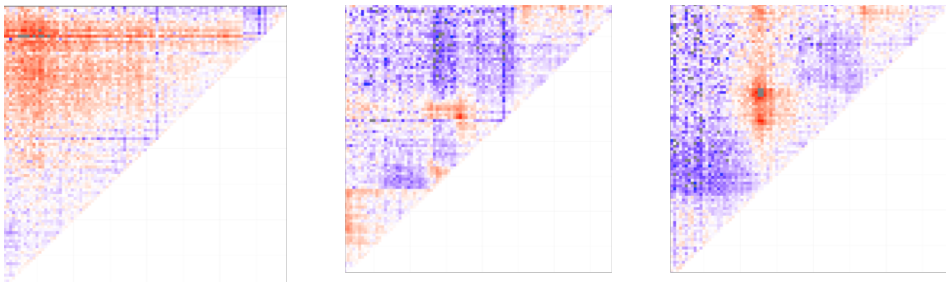


Figure 10: Example regions given the class 3, "blue diagonal," label.

Additional interesting patterns were observed, especially the "America" pattern, Figure 11, named after our great flag. It was surprising to find many examples of this specific pattern. It seemed that large squares in the top left of the image were also enriched for in dark blue colors, and therefore devoid of expected interactions.

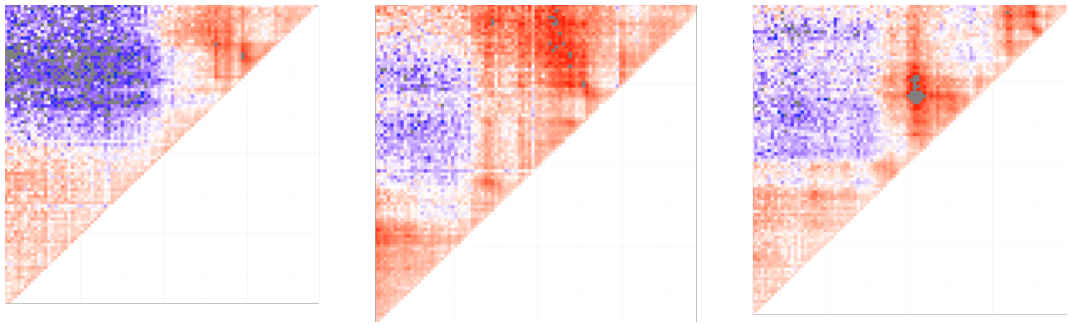


Figure 11: Example regions given the class 4, "America," label.

The final class is the "blue disperse" category and encompasses a wide amount of the images. This miscellaneous category is similar to the the "disperse" class from the Loop-Domain Based Classification.

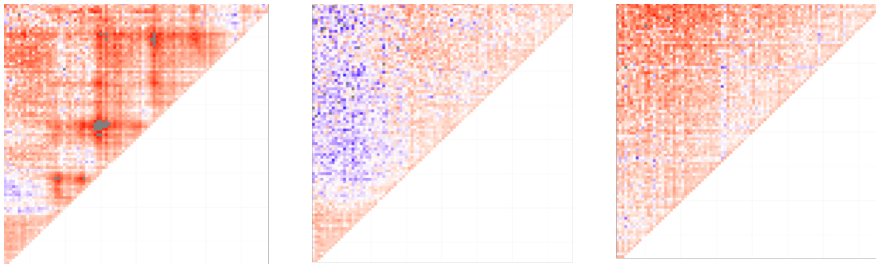


Figure 12: Example regions given the class 5, "blue disperse," label.

3 Supervised Learning

We can define the problem of recognizing various shapes in the DNA folding from the data described in the introduction as a multinomial classification problem. The input is a matrix of the interactions counts between pairs of chromosome sections normalized over their expected value. The output is their classification into one of the shapes described in section 2. There are two labeling schemes. One based on the shapes of the red (positive) sections and one based on the shapes of the blue (negative).

In order to apply supervised machine learning to the classification task we had to build

our own dataset of human labeled data. The data from the genome was broken up into 1 megabase sections of the chromosomes, using 10 kilobase resolution for collecting interaction information. Each labeled example was a 100 x 100 float array. While labeling the entire genome is doable (3,000 examples), in the interest of time and focusing on the technical aspects, only chromosomes 3, 5, 10, 12 13, 14, 15, 16, 20, 21, 22 were labeled. This provided a total of 1,117 labeled examples.

While there are numerous techniques for classification tasks, CNNs seemed to be a promising approach for this particular data set. In particular, while the data is numeric, it can be visualized by converting the values to pixel intensities or colorings. (Note in our experiments we use intensities so that the "images" had only 1 channel instead of colored pixels, which would be represented with 3 channels). Furthermore, the shapes that are being classified in the examples are analogous to shapes in simple images and are translation invariant because loops and domains and other shapes can appear at various positions in the "image". For these reasons the convolutional aspect of CNNs and simple networks such as those used for CIFAR10 and MNIST seemed to be a good starting point for applying neural networks to this classification problem.

We used Keras [1] to implement the neural network experiments. We tested on the CIFAR [8] and MNIST [15] networks, based on implementations provided in Keras examples, which are relatively simple networks that can be trained quickly. CIFAR had better results for both classes of labels so we only report results from that network. Our images were 100 x 100 float matrices normalized to have all values between 0 and 1. We had 1,117 labeled examples. 10% were randomly held our for testing. Of the remaining images, 10% were used for validation. Both networks were modified slightly for tuning. We increased the number of filters and sizes of filters. Details of each network can be found in the code so we will omit all tuned parameters here. Full parameters can be seen in `https://github.com/nlhuong/HiCshapes`. Both networks were run for 20 epochs, solved with adadelta and adam for MNIST and CIFAR respectively.

After human labeling was completed, a random 100 examples were relabeled to test human labeling accuracy (the human was not trained for 20 epochs so we didn't expect overfitting). The human accuracy for relabeling was 70%. Using the CIFAR network we achieved 61% for the loop and domain based classes (red) and 67% accuracy for the negative space classes (blue) each having 6 class labels. Better accuracy could potentially be reached with more parameter tuning, more preprocessing of the inputs, and more accurate human labeling to increase our baseline gold standard. Nevertheless, these simple networks show potential for effective recognition of the shapes defined and could potentially be used to machine label the rest of the genome and may likely be applied to different resolutions and normalization types (which would substantially grow the number of images in the dataset). Below we show examples of mislabeled images and some explanation of why these may have been hard for the network to learn.

Supervised Classification Errors

To better understand where the supervised model was failing, we looked at specific examples in both the Loop-Domain and Negative Space based classification schemes. The following were simply spot-checked and are likely not representative of the entire reason for errors.

Loop-Domain Based Classification Errors

Many inconsistencies between human and machine labeling with the Loop-Based classification system could be credited to the human inability to observe subtle differences in images or having trouble choosing between highly similar categories. This is an interesting result as the classes were indeed designed by the human, but were also chosen to reflect currently understood chromosome patterns.

The first example of differential labeling is shown in Figure 13. The tensor flow model labeled this image as "an information loss," while the human called it "a domain" pattern. At the first glance, the human is correct, but a closer inspection reveals that regions of data loss, which support the machine's assignment. It is likely that the learner is acutely sensitive to lost information as this is simply any value of zero in the bedpe file, and is thus a simple trait to detect.

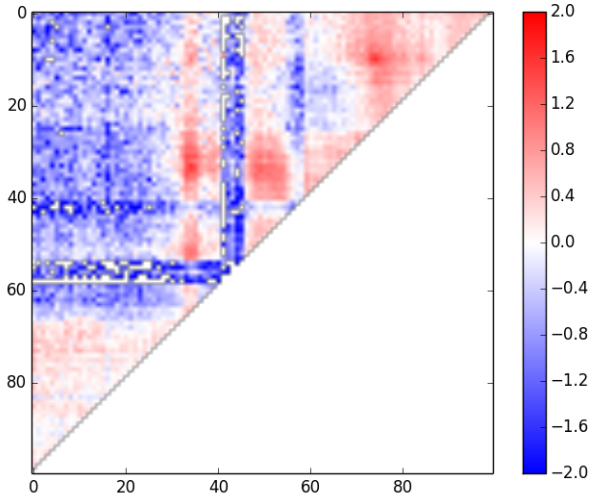


Figure 13: Example frame that the supervised learner identified as "information loss," while human classification claimed a domain pattern.

Another interesting case where the human and machine often did not agree was in the definition of strong versus weak loops. Figures 14 and 15 show loops called strong and weak by the human, respectively, while the classifier disagreed. A closer examination that the frame in Fig. 15 definitely has stronger loops than that in Fig. 14. This brings up an interesting point that the machine is likely better suited to fine classification systems, where accurate pixel values are important. Human classifiers also suffer from relative intensities when ranking multiple images in a row, perhaps a frame with incredibly weak peaks was shown before the frame in Figure 14, causing the human to consider Figure 15 strong. This result is encouraging for the use of standard classification machine learning techniques in consistently identifying pre-specified patterns.

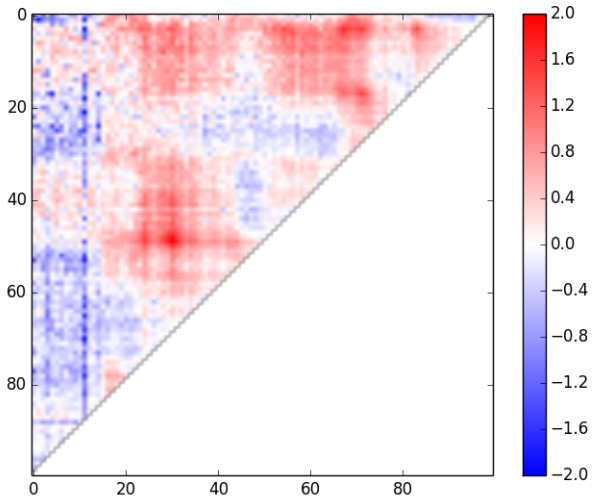


Figure 14: Example frame that the supervised learner identified as "weak loops," while human classification claimed "strong loops".

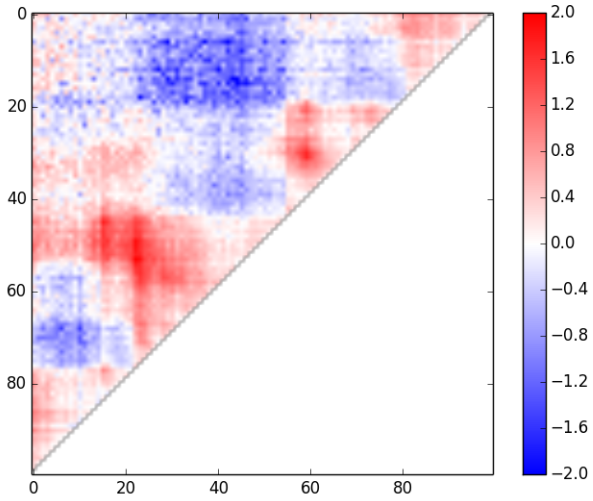


Figure 15: Example frame that the supervised learner identified as "strong loops," while human classification claimed "weak loops".

Negative Space Based Classification Errors

Although the classifier performed better on the Negative Space based classification system, the discrepancies between human and machine labeling are not clearly caused by the human error. This can be seen in Fig. 16, where the supervised learner decided the frame fit the "America" category, even though the large blue contiguous portion was not square and not in the upper left of the image. This is arguably a case where the machine under performed, but would potentially improve with a larger training set.

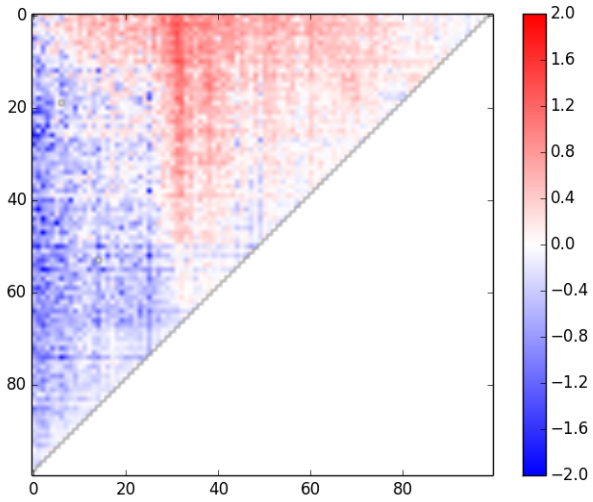


Figure 16: Example frame that the supervised learner classified as "Blue Pockets" while a person classified the frame as an "America" pattern.

An example of where the image perhaps does not fit well into the classification system created is shown in Figure 17. The human classifier noticed the prominent blue pocket pattern of the image while the machine saw the strong America blue in the corner. Both classifications have value which brings up the question of how to weight the different classes, or determine whether an image should be put in one group over another when it is at the verge of both. It would be interesting to try and parse out the decision making process of the machine on this example

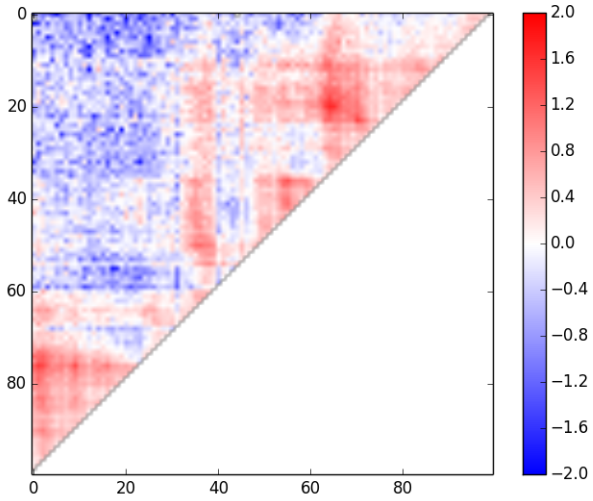


Figure 17: Example frame that the supervised learner classified as "America" while a human classified the frame as a "blue pockets" pattern.

Finally, there appears to be another situation where machine labeling may be preferred over human labeling as depicted in Figure 18. The human likely observed the blue pockets, but decided that they were too large and faint, while the machine observed the pockets and was content in its classification. Upon closer a inspection, we are inclined to agree with the machine choice in this particular instance. The multiple case studies described here, where a

discrepancy exists, but the human is at fault falsely decrease the accuracy of the model. If the human had become better trained analyzing the images, or perhaps simply labeled more frames, it could be expected that fewer incorrect labels would be present in the test set.

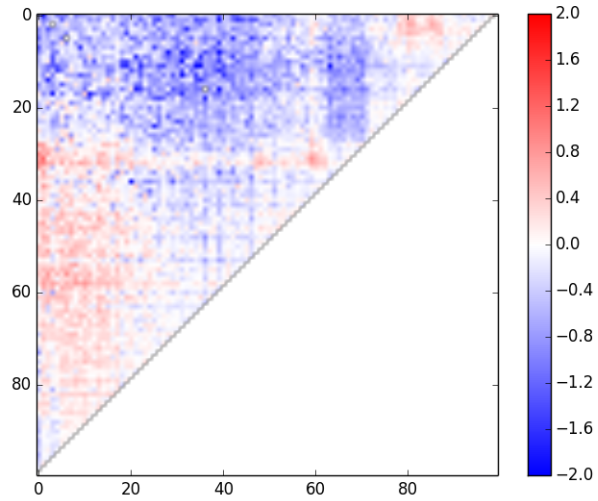


Figure 18: Example frame that the supervised learner classified as "blue pockets" while a human classified the frame as a "diffuse" pattern.

4 Unsupervised Learning

Another approach to detect interesting shapes in the HiC data is to apply unsupervised methods. In this project we decided to take an approach common in computer vision. From the set of available "frames" of HiC data, which were described earlier in section2 we learn to detect automatically the more interesting shapes. In this project we explore two approaches. The first one is based on clustering the interaction frames based on the features detected by the scale-invariant feature transform (or SIFT) algorithm. The second is the autoencoder convolutional neural network method for discovering a lower-dimensional structures in the interaction windows.

Scale-Invariant Feature Transform

We treat the 1Mb frames cut out from the diagonal of the interaction matrix as pictures, and use the number of interactions as 'pixel intensity'. When analyzing the matrix iterations we are interested in detecting shapes that are invariant to translation, scaling, and rotation. SIFT is a method that achieves exactly this goal. As stated in the original paper,on SIFT by David Lowe [11] the algorithm goes through the following stages to extract features:

1. *Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.*

2. *Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.*

3. *Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image*

data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. *Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination*

We used the SIFT implementation provided by OpenCV library. The detailed tutorial can be found at `http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html`.

Fig.19 displays ten examples of the 1Mb frames of HiC data and the features detected by SIFT. We see that the algorithm mostly picks up on high intensity interactions, which corresponds to loops. This could be useful, even as an automatic way of finding loops in the chromosomes. Apart from that though, SIFT also pick upon wider range interactions, which potentially could correspond to large domains.

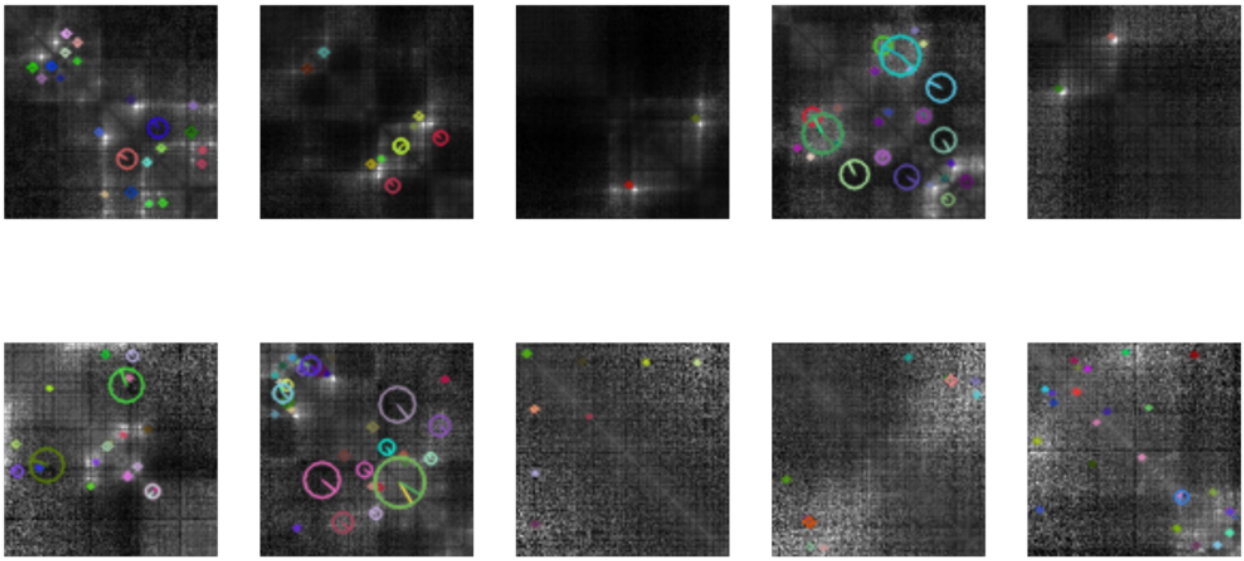


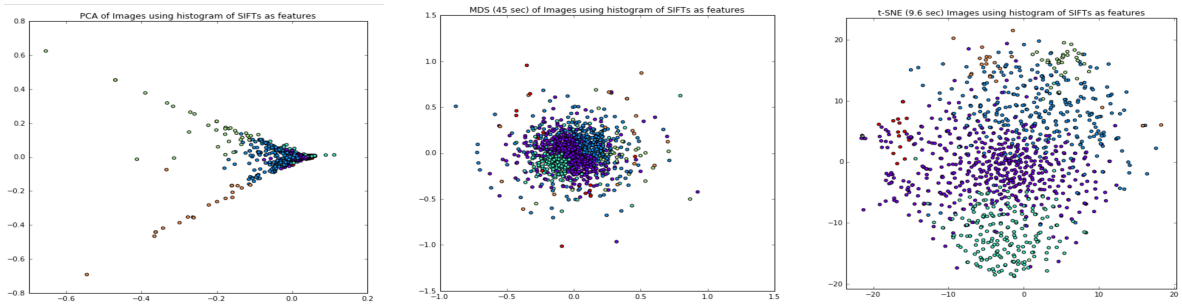
Figure 19: SIFT features detected for a sample or randomly chosen frames.

Since we are interested in detecting interesting shapes besides simple loops and domains, the next step for us was to somehow cluster the available interaction frames based on the SIFT features detected. That is we are interested whether the features detected are organized in any interesting more complicated patterns. In other words, we were interested whether the 1Mb interaction windows, described by the SIFT features, lie in some lower dimensional manifold. For this purpose we used a common approach called *the bag of words*[2], [10]. We first aggregate all the feature descriptors detected in all available 1Mb windows. Note that we only use the descriptors, as the location and the orientation of the keypoints is not relevant to us. Then we cluster the descriptors into classes. We used k-means algorithm with 200 clusters, but other choices of clustering methods and the number of classes are equally valid.

After the classification of SIFT features into 200 separate groups, we can iterate over the interaction frames and compute the frequency of each of the feature groups. That is we find a histogram vector of length 200 over the feature groups found in the previous step. In cases of the images where SIFT found no keypoints, an all-zero vector is assigned. At the end of this process we have a resulting $N \times 200$ matrix, where N is the number of considered interaction

frames, and the entries are the counts of the features in that frame assigned to a corresponding SIFT feature class. The matrix is further normalized so that each row sum is equal to one.

It is hard to comprehend the 200-dimensional space of the feature class count table. Dimensional reduction techniques can be used to visualized the data in 2D. We used a simple PCA, a metric MDS on the Euclidean distance, and t-SNE to analyze the data. Fig. 20 shows the resulting 2D maps.

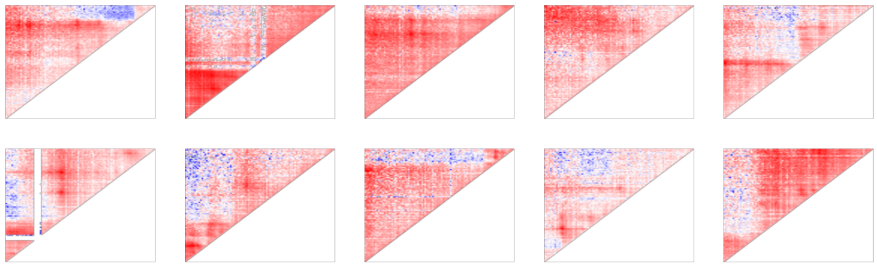


(a) PCA (b) MDS (c) t-SNE

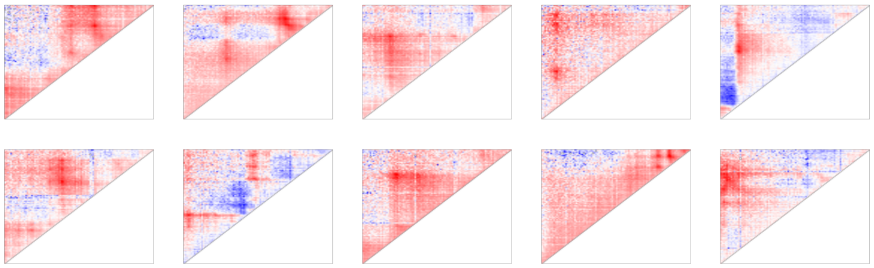
Figure 20: SIFT clusters count data for the images in a lower dimensional representation. Colors correspond to labels assigned to the frames by the k-means algorithm with $k = 6$.

The histogram data table can be used to cluster the interaction frames. We chose $k = 6$ clusters and ran the second stage k-means algorithm. To classify the interaction frames. The choice of the number of clusters could be improved by cross validation or by computing the Gap statistics. The cluster assignment of the frames is displayed in the plots in Fig. 20. Sample frames with different cluster assignments are shown in Fig. 21.

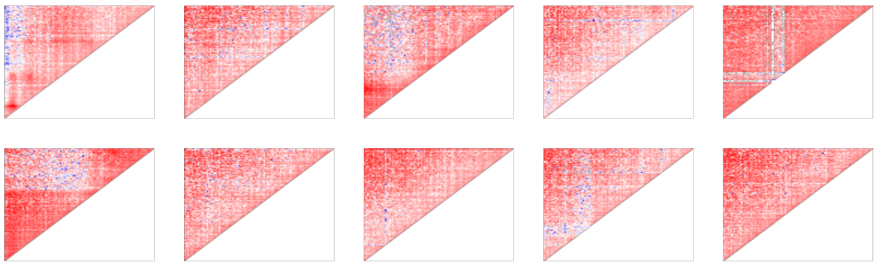
Figure 21: Frames classification based on k-means over SIFT feature groups histogram data. Notice how class 5 contains mostly frames with 0 interactions (depicted here in white).



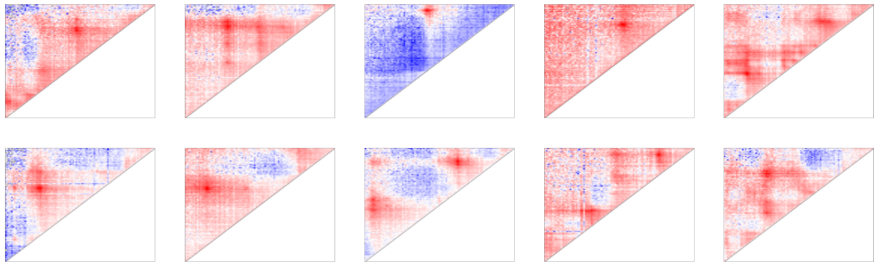
(a) Class 0



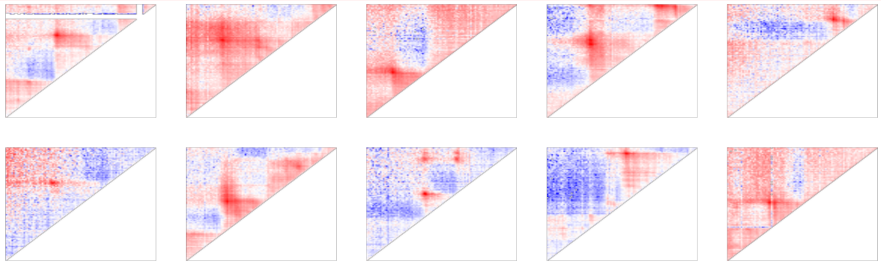
(b) Class 1



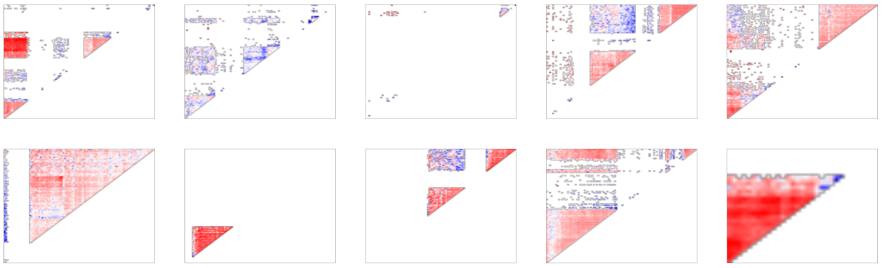
(c) Class 2



(d) Class 3



(e) Class 4



(f) Class 5

Autoencoders

While we could try to cluster the raw data/images, interpreting the data as a single flat vector would miss the two dimensional qualities of the data and would not be translation invariant. In other words, if two images have similar loops, but the loops appear at different distances from the diagonal, clustering on the raw data would likely not pick up on this similarity. Therefore, we need to compress the raw data into a lower dimensional dimensional space, but the mapping needs to also be to invariant to scaling, translation and rotation transformations. In the previous section we described SIFT as a method that attains this goal. Here, we discuss how to use convolutional neural autoencoders for this task. The autoencoder is composed of an encoding and decoding phase. The encoding phase uses convolution and pooling layers to convert the image to a lower dimensional representation and the decoding phase undoes all the encoding layers to rebuild the original representation. We use binary cross-entropy as the loss function. The network learns to rebuild the original input and in the process, learns a good vector representation of the images.

Then using these vector representations of the images we can apply unsupervised clustering techniques similar to the ones applied to the SIFT feature vectors. The main goal of using unsupervised learning is to see if we can automatically identify new classes that we didn't consider looking at the raw data. Also, this method doesn't require human time to label examples, which will be more scalable for larger datasets.

Our CNN composed of a set of encoding and decoding layers each composed of three convolution layers interspersed with three max pooling layers. We used 50 filters in the first convolution layer and 25 filters for subsequent layers. The final compressed representation was a $(25 \times 12 \times 12)$ representation of the data (3600-dimensional) compared to a (100×100) raw image input, i.e. a ~ 3 fold dimensional reduction. This is not a high compression rate, but the hope is that this lower-dimensional representation if good for denoising and reconstruction the original input. In fact the main incentive is to find a are representation that captures the most of the structure present in the input data. The implementation of the CNN was done using Keras and TensorFlow deep learning library. We based our followed closely the methods described in the tutorial available at `http://blog.keras.io/building-autoencoders-in-keras.html`.

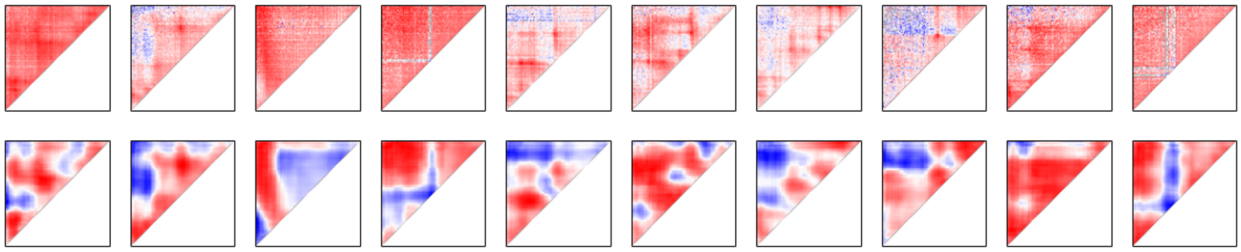


Figure 22: Autorencoder model 1 results. Example of randomly chosen frames. Input raw frames (top row) and decoded output (bottom) row.

Fig. 22 shows the reconstruction of the original frames as they are passed through the CNN. We can see that the reconstructed images remove a large portion of the noise present in the original data. The shape detection task should be easier on this set of reconstructed/denoised frames rather than on the original files. We can pass the reconstructed images throught the SIFT framework as described in the previous section. An alternative is to try clustering the encoded lower-dimensional representation of the images.

We trained another, simpler CNN which was more geared towards denoising task. This model includes only two encoding and decoding convolutional layers interspersed with max pooling layers. However, the convolutional layers each are composed of 50 filters. In this model the encoded image is 50 filters by (25×25) which is more than (100×100) in the original image, but the over representation is structured, and can remove unnecessary noise. The example of decoded images for this model are shown in Fig. 23

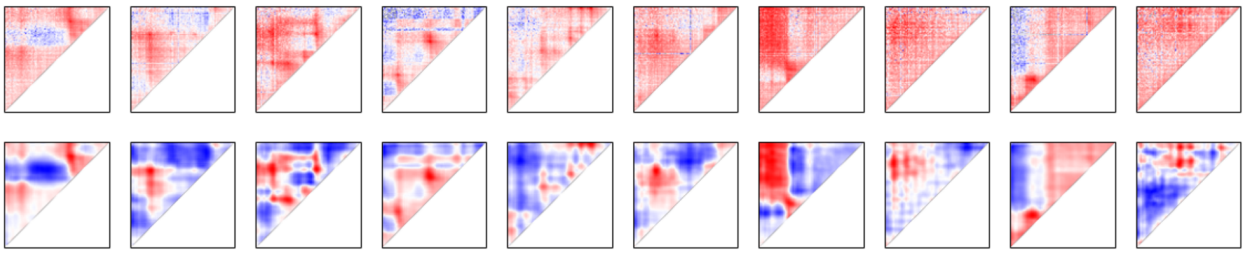


Figure 23: Autorencoder model 2 results. Example of randomly chosen frames. Input raw frames (top row) and decoded output (bottom) row.

While the image reconstructions showed that we were able to capture a high quality representation of the data, testing on a few sizes of clusters, we failed to see truly consistent clustering based on the feature representations from both SIFT and the autoencoder. One of the main challenges was evaluating the quality of the clustering. Mostly, we hoped to see a consistent set of feature in the images. From browsing through separate groups of clusters we couldn't identify any obvious themes. Perhaps exploring other clustering methods besides KMeans and further tuning could provide better results.

5 Future Work

In this paper we explored a number of techniques for categorizing different shapes that can be observed from our dataset of gene segment interactions. We hope that many of these initial attempts can be leveraged for providing some initial areas for future explorations.

We attempted to classify our data using two simple neural networks. Future work may consider attempting to use deeper networks such as AlexNet or VGG and using pretrained weights. While even these small networks can easily overfit the data, larger networks with pretrained weights may be better at detecting particular shapes from the transfer learning. One challenge is that our data has 1 channel (depth) whereas AlexNet and VGG both have 3, so reusing the weights my not be trivial.

We've provided some motivation for using SIFT features to extract relevant areas of the visualized data. Future models can include key points information and build on similar image clustering techniques for finding clusters of originally unrecognized classes.

We have also shown that deep neural autoencoders can capture some structural information about the data and encode it in lower dimensional space. Deeper inspection of the encoded representations may provide more insight into better classification schemes that may be less obvious than a cursory visual analysis of the data.

6 Tools Contributed

As part of our contribution, we developed a set of tools and scripts that can be leveraged in future work. All these tools are being checked into a github repository for Oana Ursu and will be accessible for any future work.

- hand labeled 1,117 examples from the human genome with our definitions for potential classes
- SIFT feature unsupervised clustering notebook
- Autoencoder notebook
- Keras implementation of CIFAR and MNIST like CNNs for supervised classification

All code can be found at https://github.com/nlhuong/HiCshapes.

7 Discussion and Conclusion

The results of this work support future efforts to investigate novel patterns in chromosome conformation capture data. Domains and loops were characterized since the advent of 3C, and predicted even before. While these features have obvious biological meaning and are relatively easy to find in the data, far more interesting chromosome shapes likely remain undiscovered. The supervised learning using the negative space based classifications shows that nearly arbitrary patterns can be consistently identified. The next exciting step in this work would be to find meaning in the identified novel shapes. A simple method could be finding associations with the presence of a certain shape to DNA regions that are known to share function or form.

8 Individual Contributions

8.1 Dan

Implemented and tuned the CIFAR and MNIST networks for the supervised classification tasks. He also helped with an initial attempt at building bag of word image features from the SIFT features. He also provided general guidance and intuition about using neural networks for various tasks. He worked with Oana to preprocess raw HiC bedpe data into numpy arrays for easy processing.

8.2 Lan

Focused on the unsupervised part of the project including the SIFT based approach and the autoencoders. Lan wrote scripts to find the SIFT features and perform dimensional reduction (PCA, MDS and t-SNE), clustered the frames based on the Bag of Words features. She also wrote the ConvolutionNetwork.ipynb file for training the CNNs.

8.3 Rob

Created classification categories for the supervised learning and hand-labeled the roughly 1000 frames into these different categories. Gave biological background into the problem and standard chromosome conformation capture techniques.

References

[1] François Chollet. keras. *GitHub repository*, 2015.

[2] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[3] Job Dekker, Karsten Rippe, Martijn Dekker, and Nancy Kleckner. Capturing chromosome conformation. *Science (New York, N.Y.)*, 295(5558):1306–11, 2002.

[4] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012.

[5] Jos??e Dostie, Todd A. Richmond, Ramy A. Arnaout, Rebecca R. Selzer, William L. Lee, Tracey A. Honan, Eric D. Rubio, Anton Krumm, Justin Lamb, Chad Nusbaum, Roland D. Green, and Job Dekker. Chromosome Conformation Capture Carbon Copy (5C): A massively parallel solution for mapping interactions between genomic elements. *Genome Research*, 16(10):1299–1309, 2006.

[6] Peter Fraser and Wendy Bickmore. Nuclear organization of the genome and the potential for gene regulation. *Nature*, 447(7143):413–417, 2007.

[7] H Hagege, P Klous, C Braem, E Splinter, J Dekker, G Cathala, W de Laat, and T Forne. Quantitative analysis of chromosome conformation capture assays (3C-qPCR). *Nat Protoc*, 2(7):1722–1733, 2007.

[8] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[9] Erez Lieberman-aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, Richard Sandstrom, Bradley Bernstein, M A Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, and Leonid A Mirny. of the Human Genome. 33292(October):289–294, 2009.

[10] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[11] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] Erik Splinter, Elzo de Wit, Elph??ge P. Nora, Petra Klous, Harmen J G van de Werken, Yun Zhu, Lucas J T Kaaij, Wilfred van Ijcken, Joost Gribnau, Edith Heard, and Wouter de Laat. The inactive X chromosome adopts a unique three-dimensional conformation that is dependent on Xist RNA. *Genes and Development*, 25(13):1371–1383, 2011.

[13] Iain Williamson, Soizik Berlivet, Ragnhild Eskeland, Shelagh Boyle, Robert S. Illingworth, Denis Paquette, Jos??ee Dostie, and Wendy A. Bickmore. Spatial genome organization: Contrasting views from chromosome conformation capture and fluorescence in situ hybridization. *Genes and Development*, 28(24):2778–2791, 2014.

[14] Elzo De Wit and Wouter De Laat. a decade of 3C technologies-insights into nuclear organization. pages 11–24, 2012.

[15] Corinna Cortes Yann Lecun. The mnist database of handwritten digits.