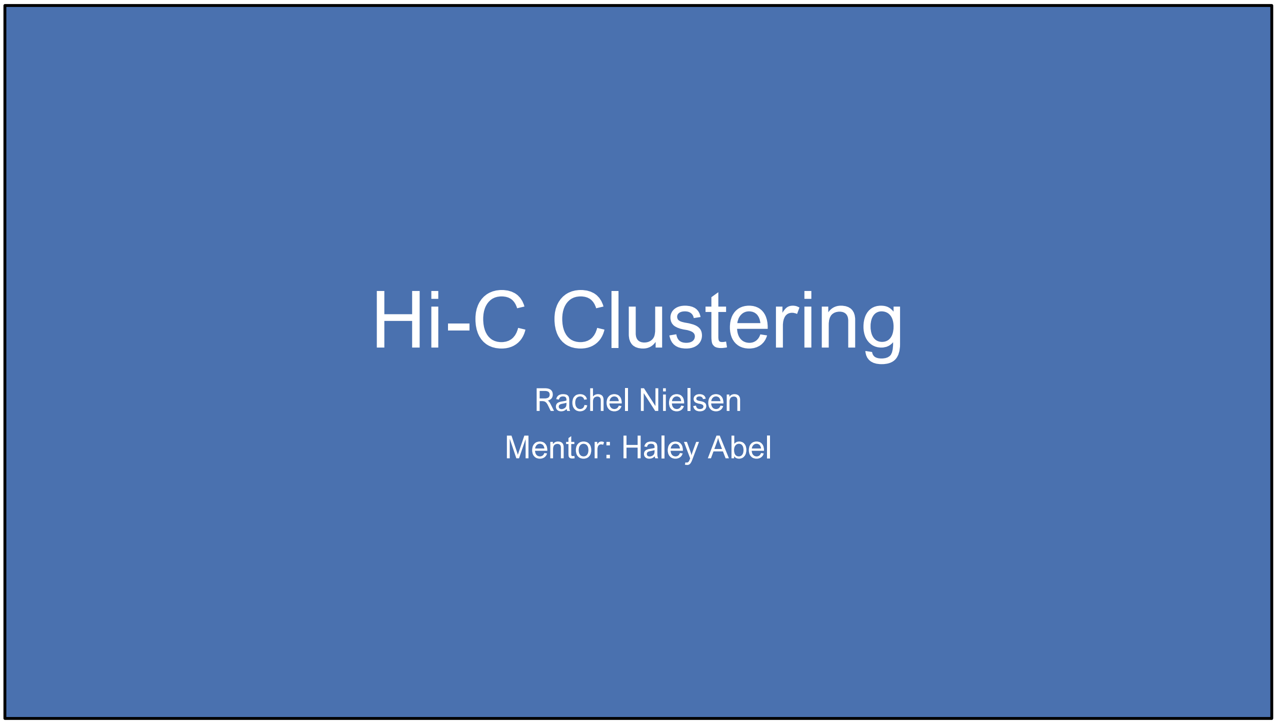
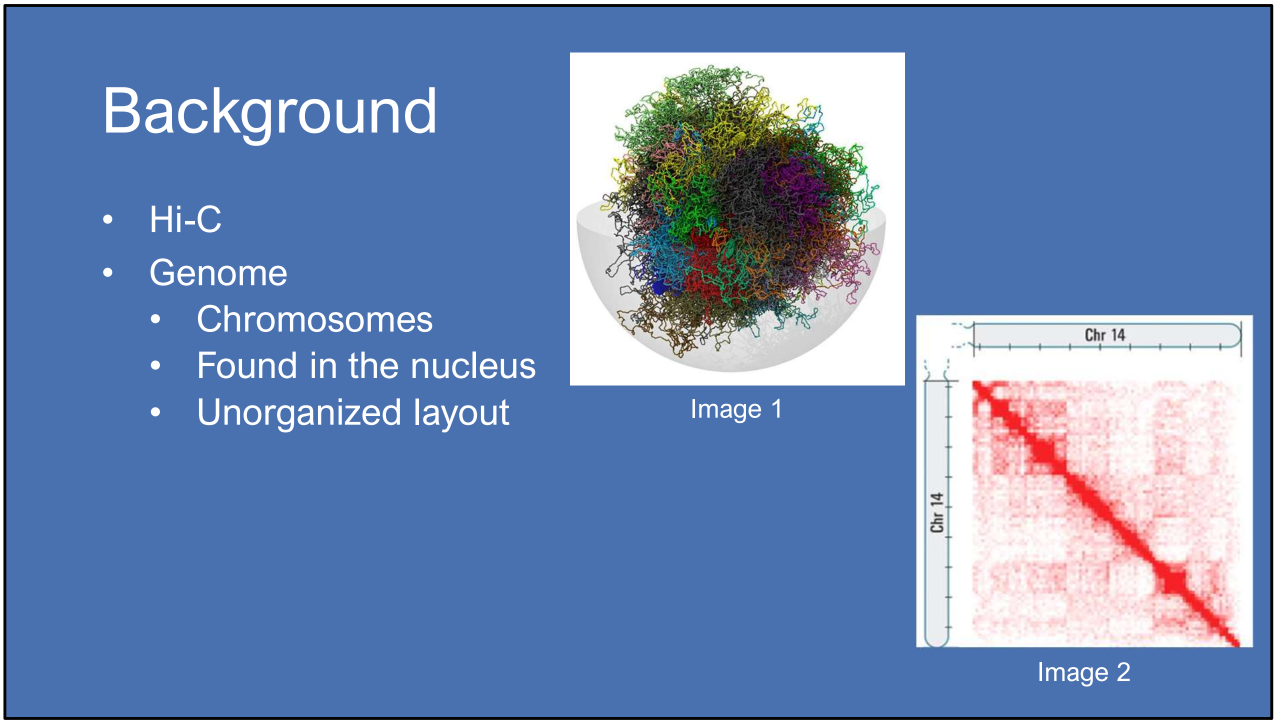


Hi-C Clustering

Rachel Nielsen

Mentor: Haley Abel

- Hi, my name is Rachel Nielsen and this summer I have been working under Haley Abel on Hi-C clustering
- Today I will be walking you through my summer project which was creating a program for finding and visualizing clusters in genomes



Background

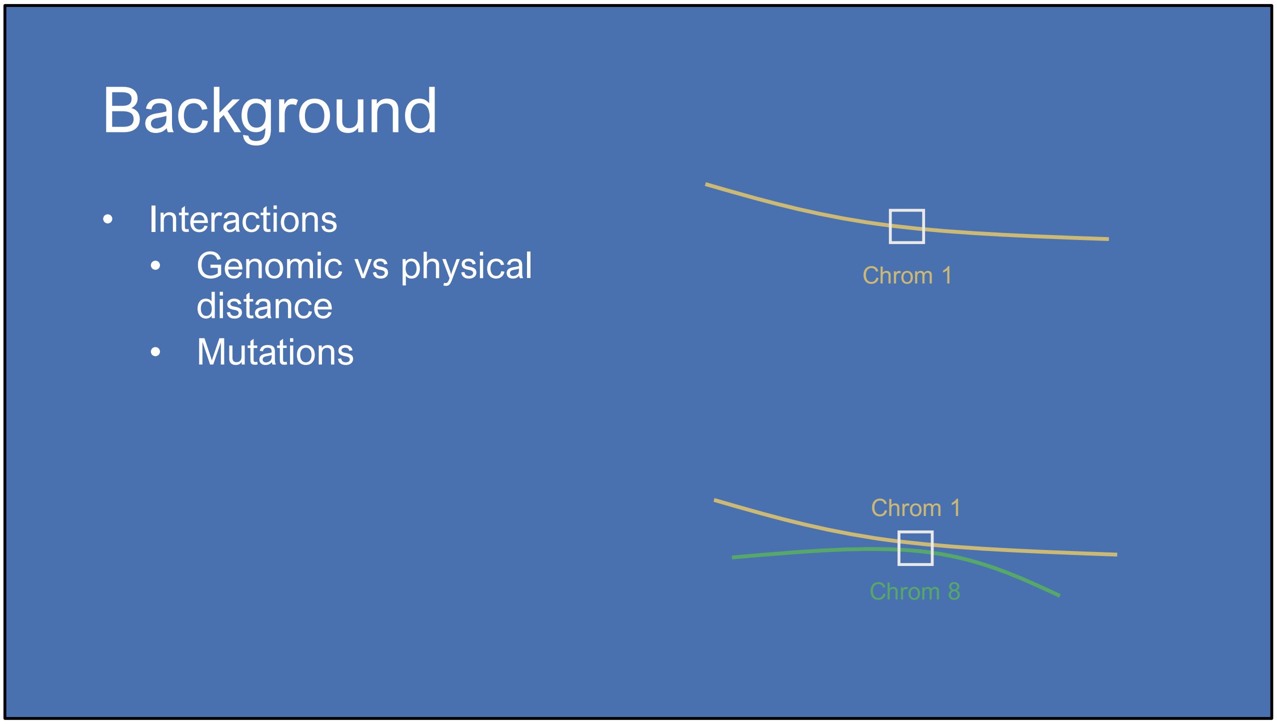
- Hi-C
- Genome
  - Chromosomes
  - Found in the nucleus
  - Unorganized layout

Image 1

Chr 14
Chr 14

Image 2

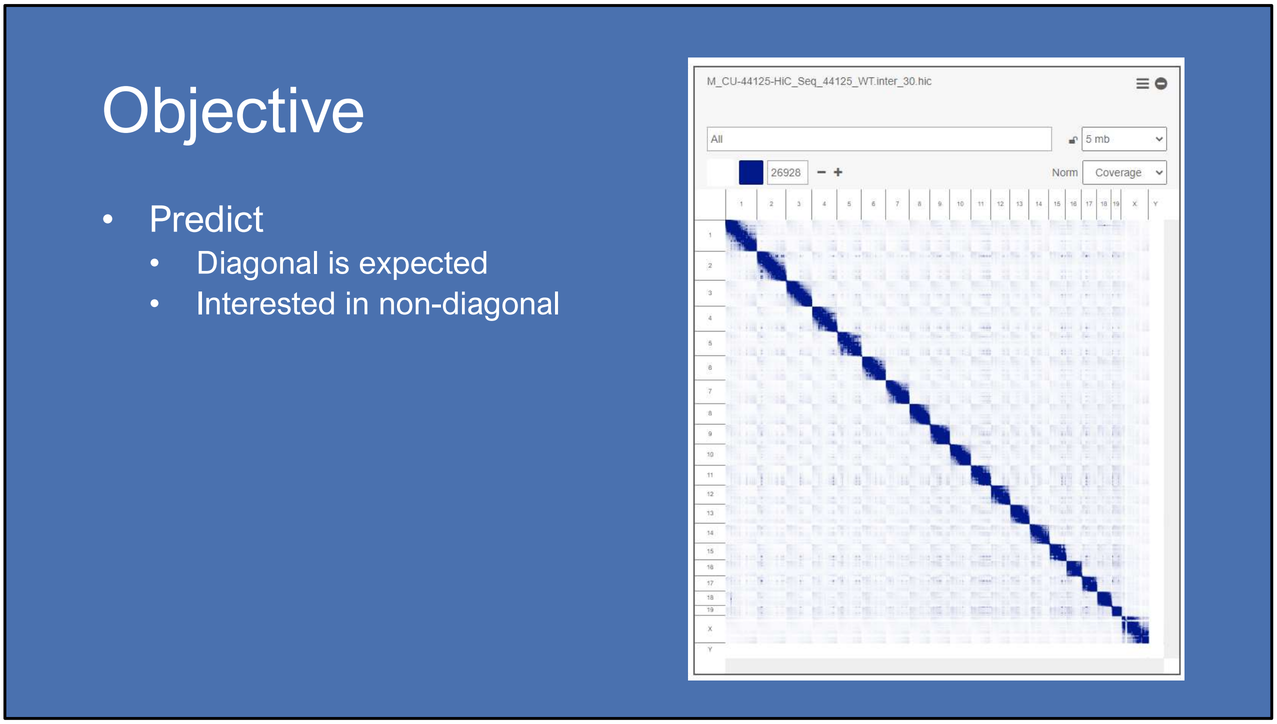
- Hi-C is a way to see how different regions of the genome interact in 3D space
- It refers to the method of looking at the genome as a whole when searching for interactions between chromosomes
- Chromosomes are found in chromatin in the nucleus of a cell and are wound up appearing like a disorganized ball of yarn as in image 1
- There are complex patterns for how these chromosomes are organized, but currently Hi-C is just beginning to uncover them
- Image 2 shows an example of Hi-C output
- It counts how often different parts of the genome interact and in this case specifically, chromosome 14
- The diagonal is very bright, because those are the parts of the chromosome that are the closest together along the strand leading to the most interactions



Background

- Interactions
  - Genomic vs physical distance
  - Mutations

- We know that connections are normally made along parts of the strand that are close together, however, sometimes since the chromatin is all bunched up, parts of the strand come into contact that aren't expected
- To visualize this, if you look at the top figure, connections along a contiguous section of chromosome 1 are expected since they are very close together in genomic space, but as shown by the figure on the bottom, sometimes connections are made because the strands are close in physical space even if they are far apart in genomic space
- These unexpected connections may lead to mutations and that is where the problem lies



Objective

- Predict
  - Diagonal is expected
  - Interested in non-diagonal

M_CU-44125-HiC_Seq_44125_WT.inter_30.hic

All

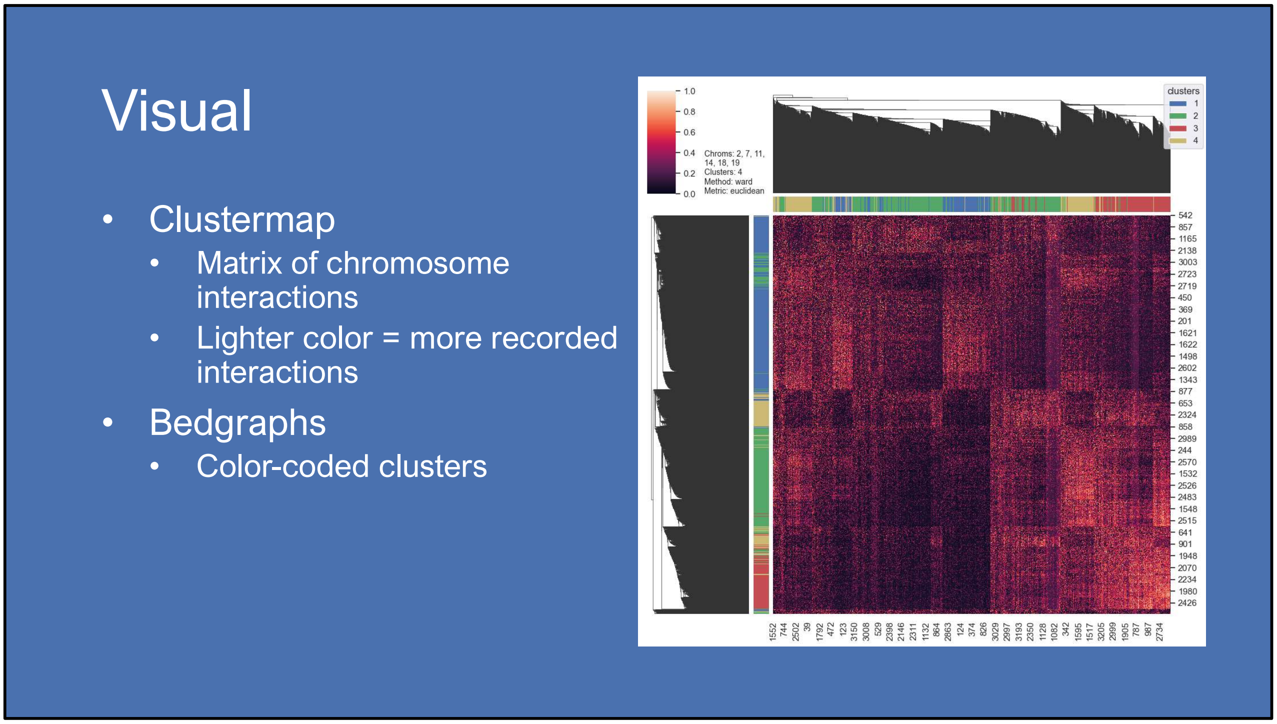
5 mb

26928

Norm Coverage

− +
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 X Y
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 X Y

- There is no easy way to predict when and where these types of mutations may occur, so the objective of my project this summer was to create a tool that aides in making these predictions by finding patterns
- The diagram on the right shows a big-picture plotting of Hi-C data for an entire genome
- Similar to the previous diagram looking at chromosome 14, the plot as a whole also tends to follow the diagonal, with there being some, but not many interactions made between chromosomes far apart in genomic space
- Because of this observation, we are interested in finding patterns for these interactions not along the diagonal



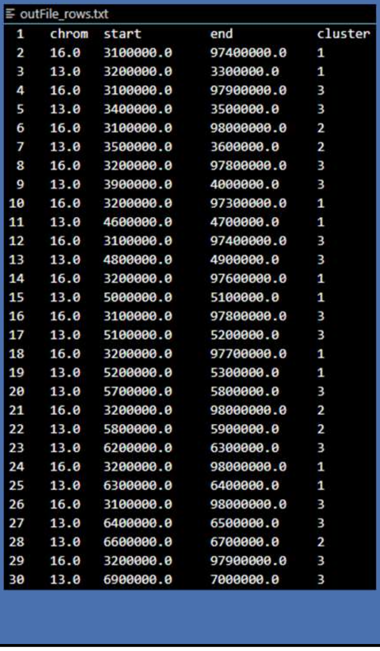
Visual

- Clustermap
  - Matrix of chromosome interactions
  - Lighter color = more recorded interactions
- Bedgraphs
  - Color-coded clusters

- To start, I was given a segment of Python code written by Haley for visualizing matrices of chromosomes and developed it further to create and output clusters of the plotted matrix
- The first part of this task is visualizing the interactions
- This diagram on the right is part of the output for my program and is called a Clustermap
- It's used for plotting a matrix of chromosome interactions and in this case, using arbitrary chromosomes 2, 7, 11, 14, 18, and 19 just to get a good spread without mapping the entire genome
- Lighter color indicates more interactions in this case
- The ribbons of color along the top and left side of the Clustermap are known as Bedgraphs
- And Bedgraphs are used to visualize the clusters created by the program as well
- For example, this Clustermap is divided into 4 clusters, each represented by a color
- In this case, the program tried to find the best way to group the rows and columns each into 4 groups
- You can see that the clusters appear to start and stop at distinguishable lines in the Clustermap
- The plot also includes all variable information at the upper left and right of the Clustermap so the user can quickly and easily identify what the plot represents

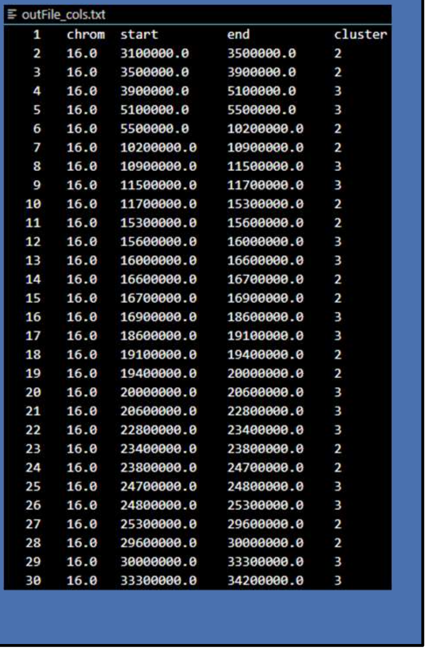
Results

- Output files
  - Chromosome
  - Start position
  - End position
  - Cluster number



outFile_rows.txt

```
chrom  start        end          cluster
16.0   3100000.0    97400000.0   1
13.0   3200000.0    3300000.0    1
16.0   3100000.0    97900000.0   3
13.0   3400000.0    3500000.0    3
16.0   3100000.0    98000000.0   2
13.0   3500000.0    3600000.0    2
16.0   3200000.0    97800000.0   3
13.0   3900000.0    4000000.0    3
16.0   3200000.0    97300000.0   1
13.0   4600000.0    4700000.0    1
16.0   3100000.0    97400000.0   3
13.0   4800000.0    4900000.0    3
16.0   3200000.0    97600000.0   1
13.0   5000000.0    5100000.0    1
16.0   3100000.0    97800000.0   3
13.0   5100000.0    5200000.0    3
16.0   3200000.0    97700000.0   1
13.0   5200000.0    5300000.0    1
13.0   5700000.0    5800000.0    3
16.0   3200000.0    98000000.0   2
13.0   5800000.0    5900000.0    2
13.0   6200000.0    6300000.0    3
16.0   3200000.0    98000000.0   1
13.0   6300000.0    6400000.0    1
16.0   3100000.0    98000000.0   3
13.0   6400000.0    6500000.0    3
13.0   6600000.0    6700000.0    2
16.0   3200000.0    97900000.0   3
13.0   6900000.0    7000000.0    3
```



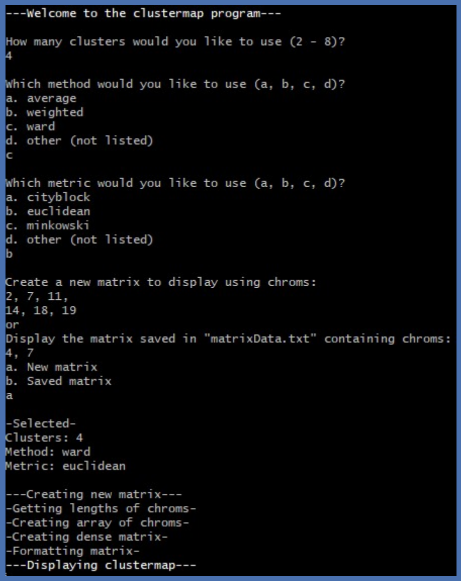
outFile_cols.txt

```
chrom  start        end          cluster
16.0   3100000.0    3500000.0    2
16.0   3500000.0    3900000.0    2
16.0   3900000.0    5100000.0    3
16.0   5100000.0    5500000.0    3
16.0   5500000.0    10200000.0   2
16.0   10200000.0   10900000.0   2
16.0   10900000.0   11500000.0   3
16.0   11500000.0   11700000.0   3
16.0   11700000.0   15300000.0   2
16.0   15300000.0   15600000.0   2
16.0   15600000.0   16000000.0   3
16.0   16000000.0   16600000.0   3
16.0   16600000.0   16700000.0   2
16.0   16700000.0   16900000.0   2
16.0   16900000.0   18600000.0   3
16.0   18600000.0   19100000.0   3
16.0   19100000.0   19400000.0   2
16.0   19400000.0   20000000.0   2
16.0   20000000.0   20600000.0   3
16.0   20600000.0   22800000.0   3
16.0   22800000.0   23400000.0   3
16.0   23400000.0   23800000.0   2
16.0   23800000.0   24700000.0   2
16.0   24700000.0   24800000.0   3
16.0   24800000.0   25300000.0   3
16.0   25300000.0   29600000.0   2
16.0   29600000.0   30000000.0   2
16.0   30000000.0   33300000.0   3
16.0   33300000.0   34200000.0   3
```

- The second part of this process was getting the program to output these clusters as data
- Since it groups the rows and columns separately, there is an output file for each which include columns for: the chromosome, the start position on the chromosome, the end position, and the cluster number that includes that segment

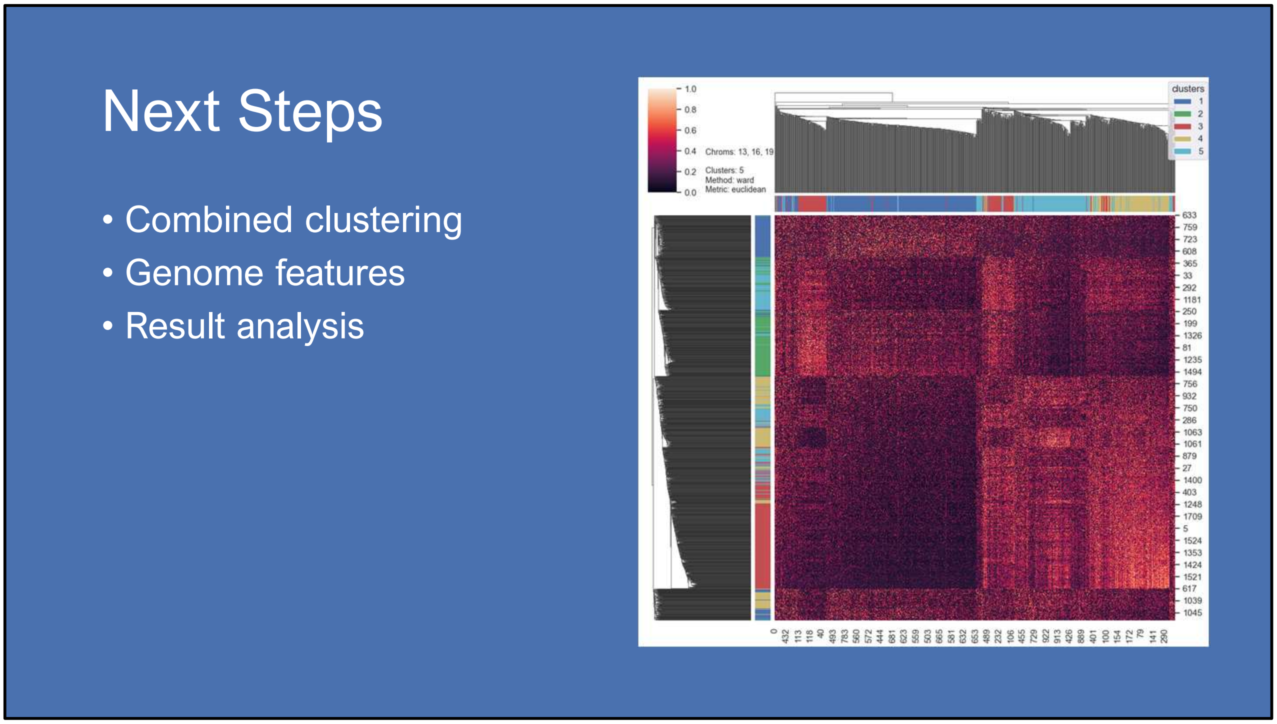
Functionality

- Command line interface
  - Clusters
  - Method
  - Metric
  - New or saved matrix



```
---Welcome to the clustermap program---

How many clusters would you like to use (2 - 8)?
4

Which method would you like to use (a, b, c, d)?
a. average
b. weighted
c. ward
d. other (not listed)
c

Which metric would you like to use (a, b, c, d)?
a. cityblock
b. euclidean
c. minkowski
d. other (not listed)
b

Create a new matrix to display using chroms:
2, 7, 11,
14, 18, 19
or
Display the matrix saved in "matrixData.txt" containing chroms:
4, 7
a. New matrix
b. Saved matrix
a

-Selected-
Clusters: 4
Method: ward
Metric: euclidean

---Creating new matrix---
-Getting lengths of chroms-
-Creating array of chroms-
-Creating dense matrix-
-Formatting matrix-
---Displaying clustermap---
```

- The first line of the code itself allows for the user to indicate which chromosomes they would like to use
- It then uses data from a mouse Hi-C experiment to create the Clustermap matrix
- Here on the right is what the program looks like running
- The program's command line interface allows for the user to give input
- It allows the user to input the number of clusters, clustering method, and clustering metric
- Also, since the program takes some time to run, it saves all information pertinent to the last matrix it displayed and the user has the option of displaying the last matrix or creating a new matrix to display using the chromosomes listed at the top of the code



Next Steps

- Combined clustering
- Genome features
- Result analysis

- The next steps in this project would be to first tie together the row and column clusters
- Currently, the rows and columns are clustered separately, though they both include the same numbers and colors
- The results could be more specific if these clusters were related to one another
- Another step would be to find connections between these clusters and features of the genome
- For example, maybe cluster 1 represents regions that are transcriptionally active, meaning making a lot of RNA and proteins whereas cluster 2 may represent regions that are not active
- Finally, observing how these results compare to results from other genomes could help to better identify the patterns in interchromosomal interactions

Acknowledgements

- Haley Abel
- Chris Miller
- Timothy Ley

Thank you

Sources

- https://www.genengnews.com/topics/omics/human-genome-modeled-in-3d-via-proximity-pair-analysis/
  - Image 1
- https://www.youtube.com/watch?v=-MxEw3IXUWU
  - How it Works: Proximo Hi-C Genome Scaffolding
- https://youtu.be/Hk5ixO7Tb24
  - 2020 STAT115 Lect15.1 HiC Introduction
- https://science.sciencemag.org/content/326/5950/289.figures-only
  - Image 2