A method to recognize universal patterns in genome structure using Hi-C

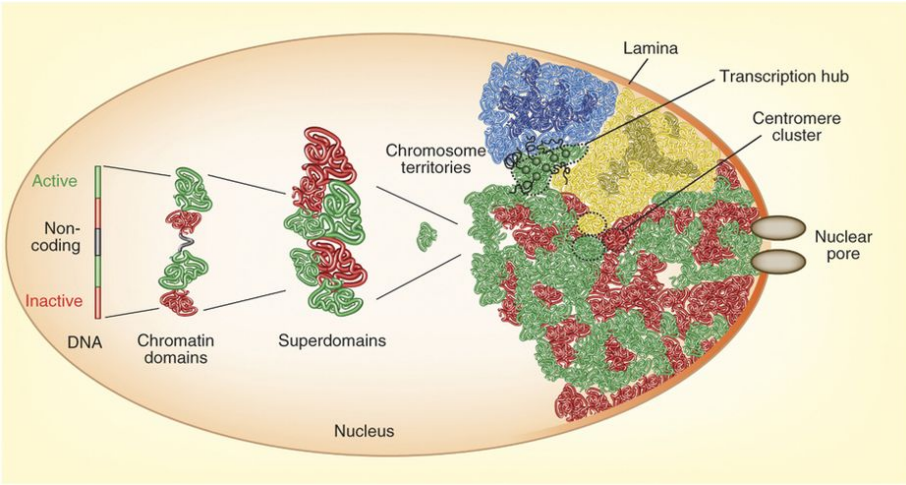
Neil Chowdhury

Phillips Exeter Academy

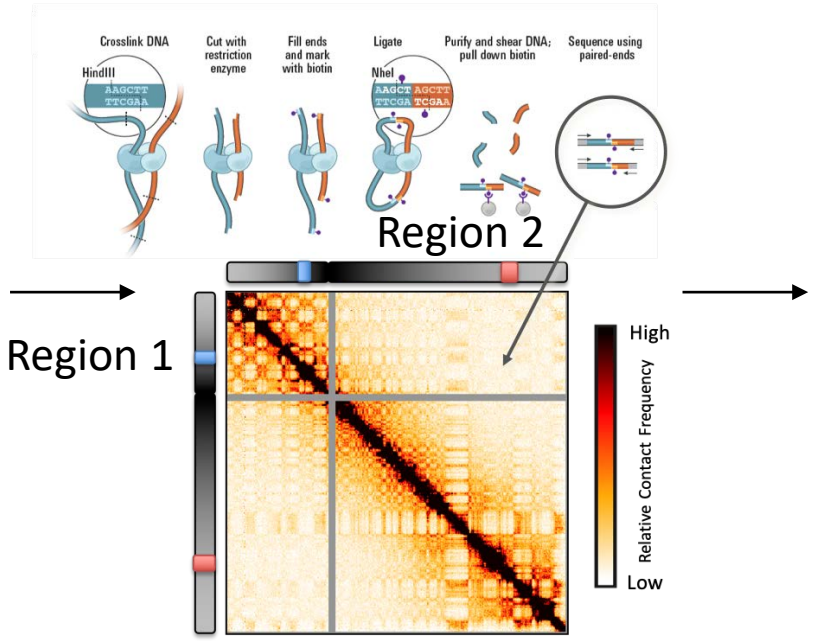
Mentor: Sameer Abraham

Massachusetts Institute of Technology

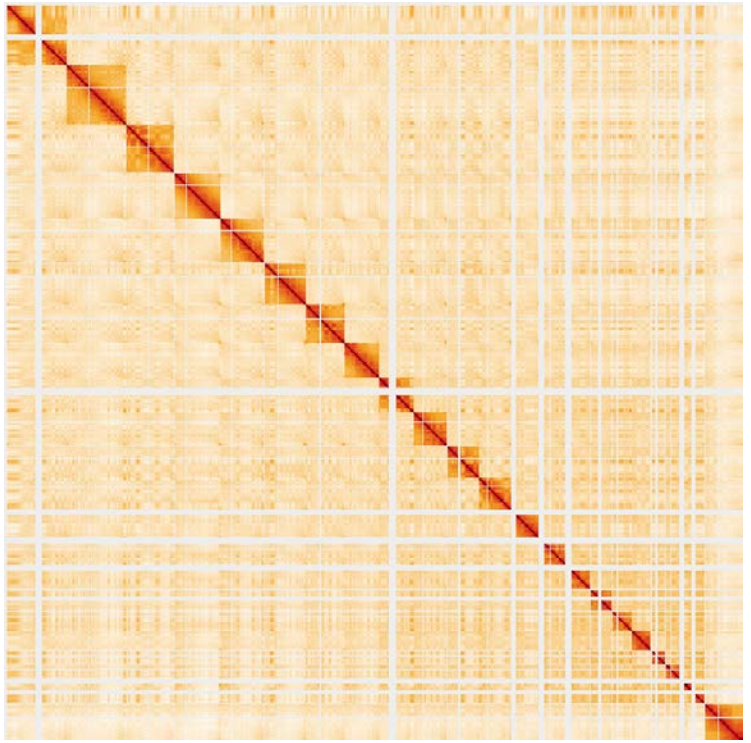
Introduction to Hi-C



Chromatin structure in the cell nucleus



Hi-C process



Hi-C Matrix of
Rao et al. (2014) GM12878 genome

- Hi-C: Process of finding contact probabilities between every pair of regions in a DNA strand
- Graphed as a square adjacency matrix (rows/cols are regions, cells are probabilities)
- Contact probabilities measures the interaction frequency of two regions

Patterns in Hi-C

Large scale

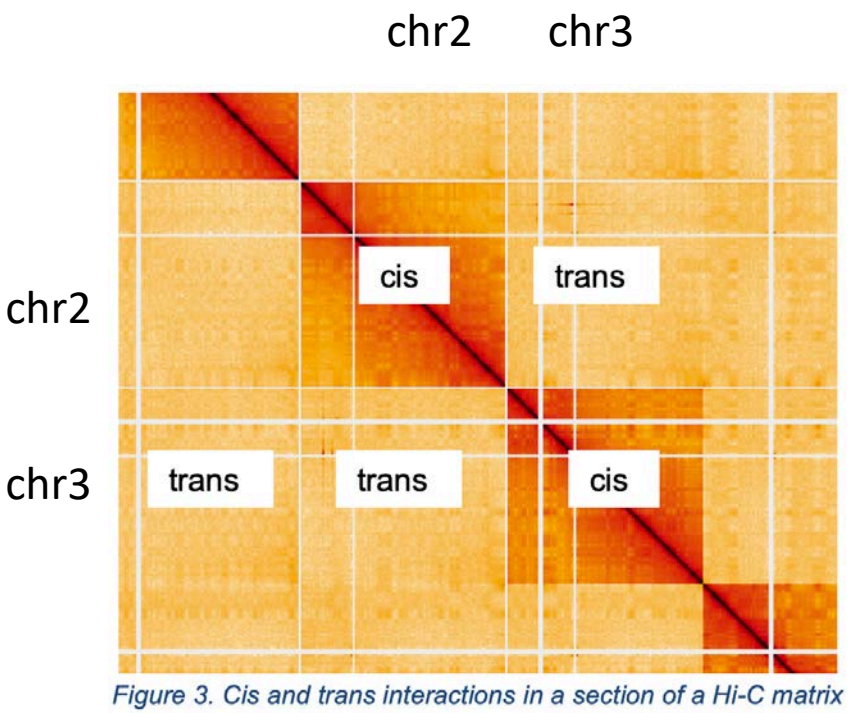
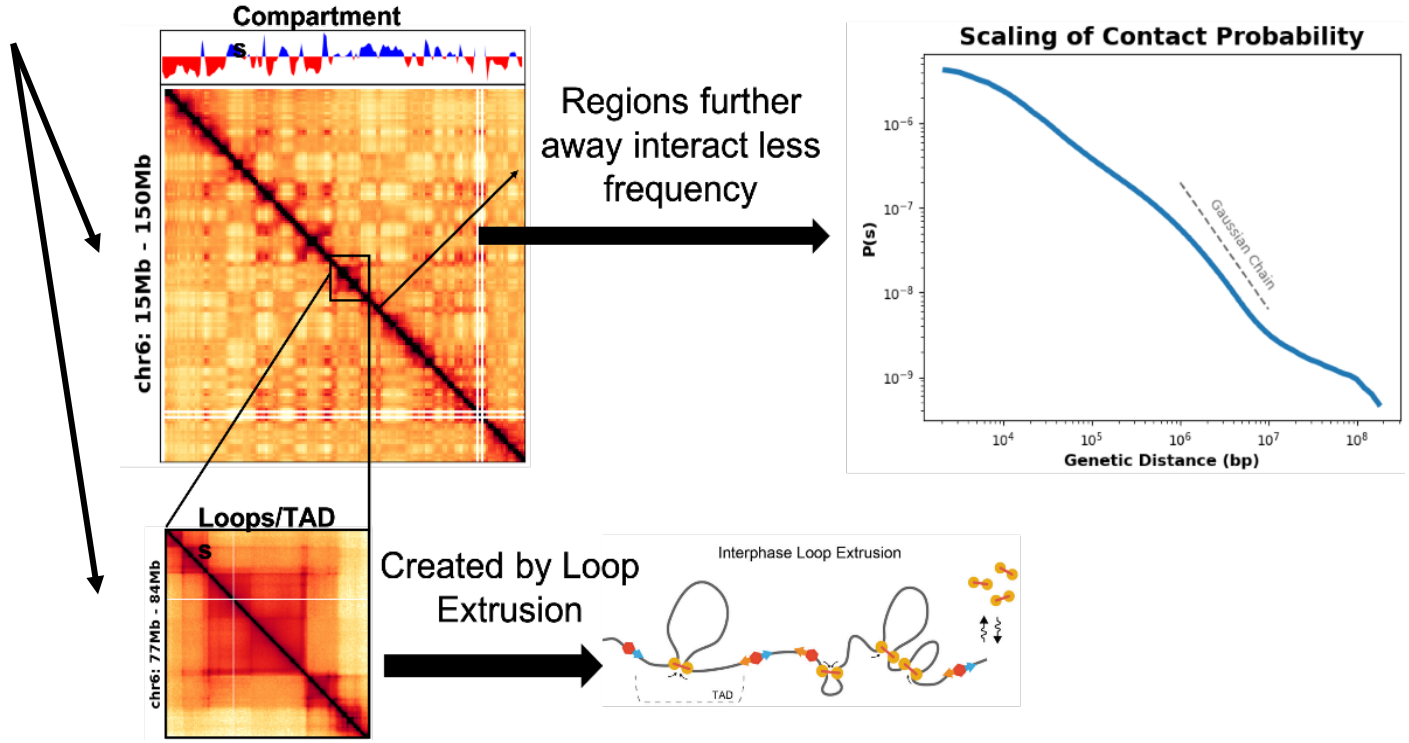


Figure 3. Cis and trans interactions in a section of a Hi-C matrix

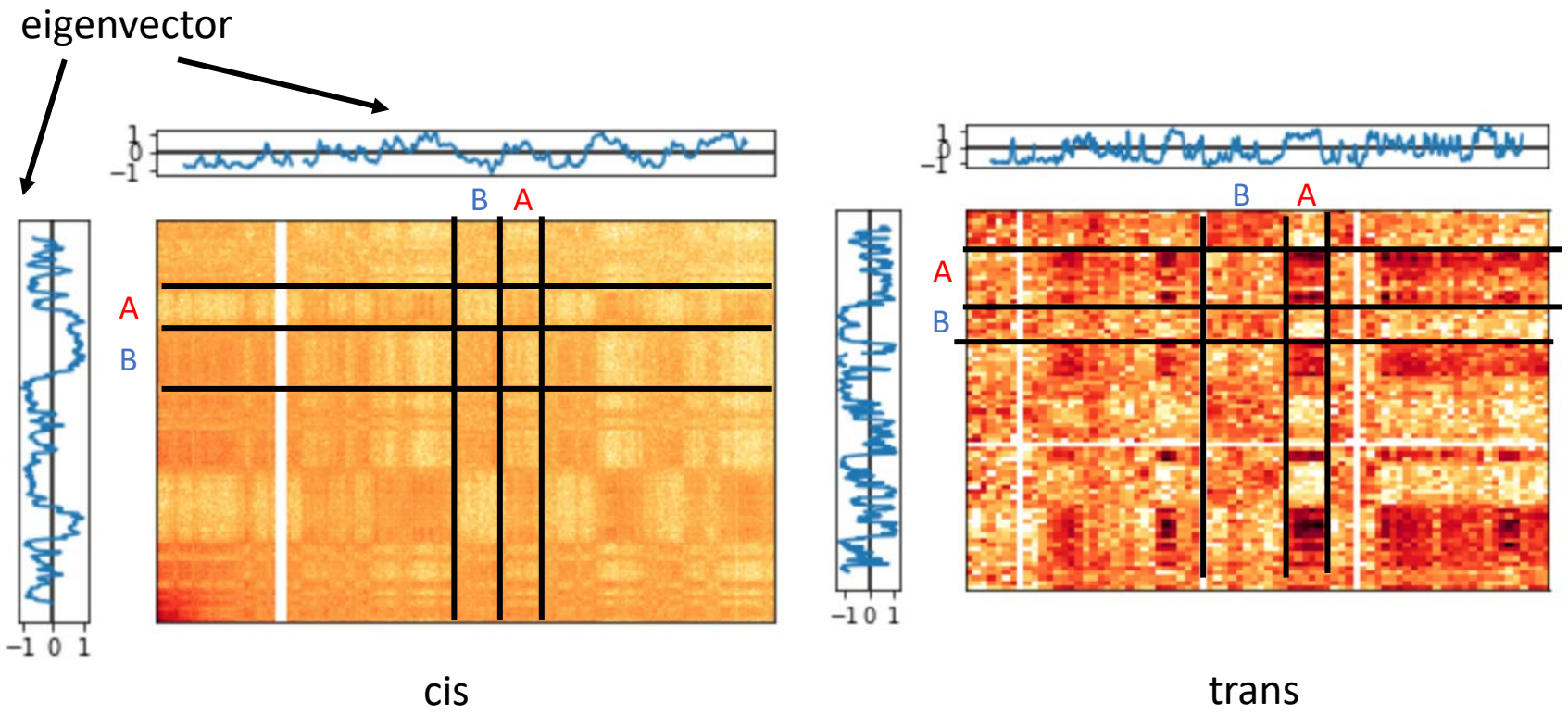
Cis interactions



Datasets used: Schwarzer et al. (2017) NIPBL mutant, Rao et al. (2014) GM12878

Compartmentalization

- Checkerboarding pattern
- Seen in both cis and trans interactions
- Pattern captured by sign of eigenvector
- Clear that there are subcompartments within A/B

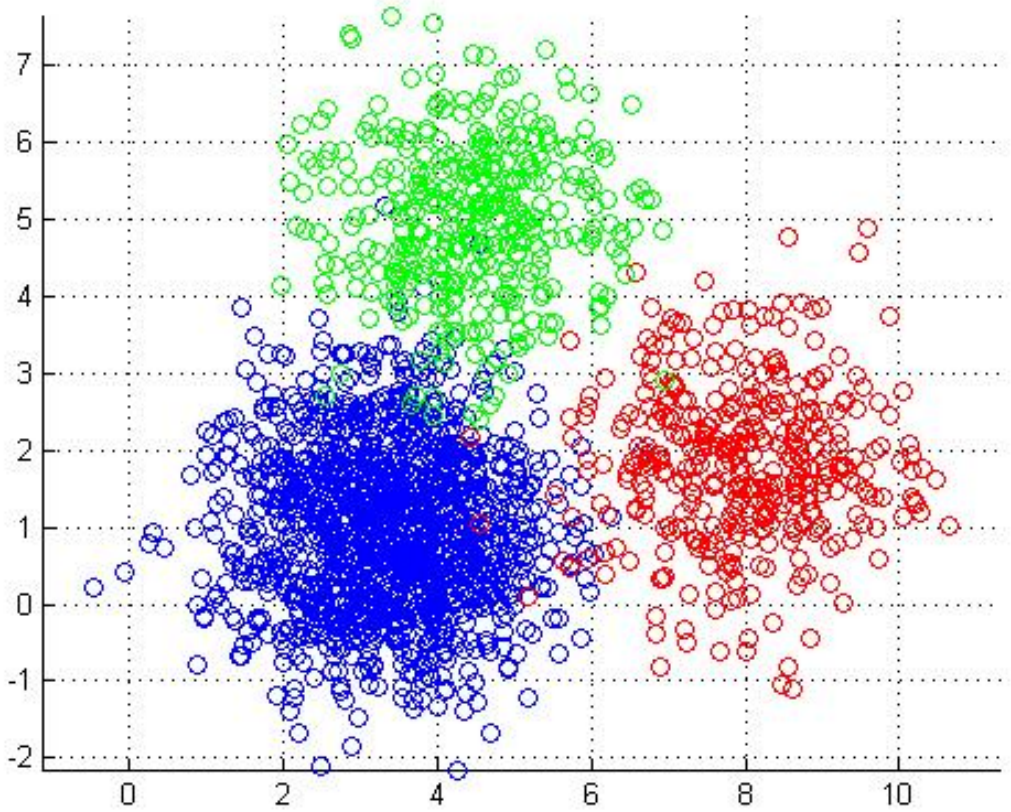


Methodology and Challenges

- Methodology: Use existing clustering techniques from data science to find subcompartments in Hi-C

- Challenges:
  - Process Hi-C data into a form amenable to existing clustering algorithms
  - Assess the quality of clusters
  - Assess the number of clusters

What is clustering?

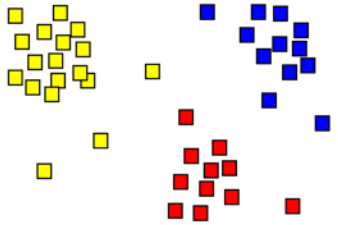


(MathWorks)

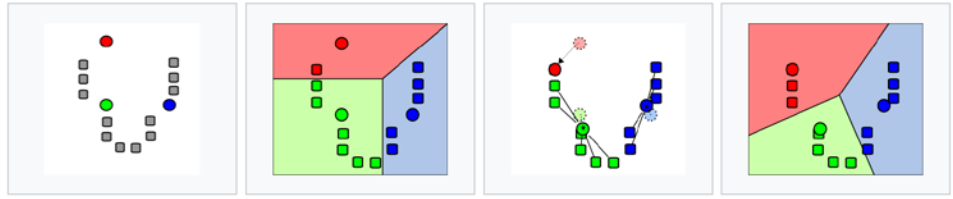
- Clusters are tightly packed groups of points in space
- Clustering is the process of algorithmically finding these groups of points

Clustering algorithms

(finding k clusters in a set of n points in space)

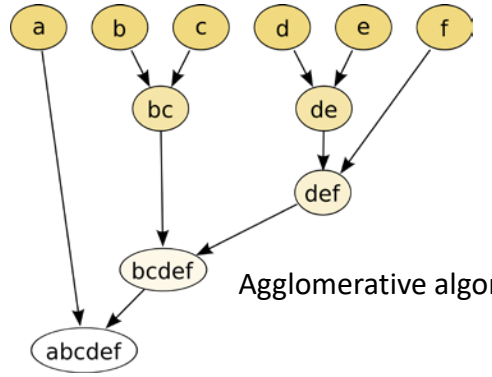


Example of Clusters (Wikipedia)



K-means algorithm

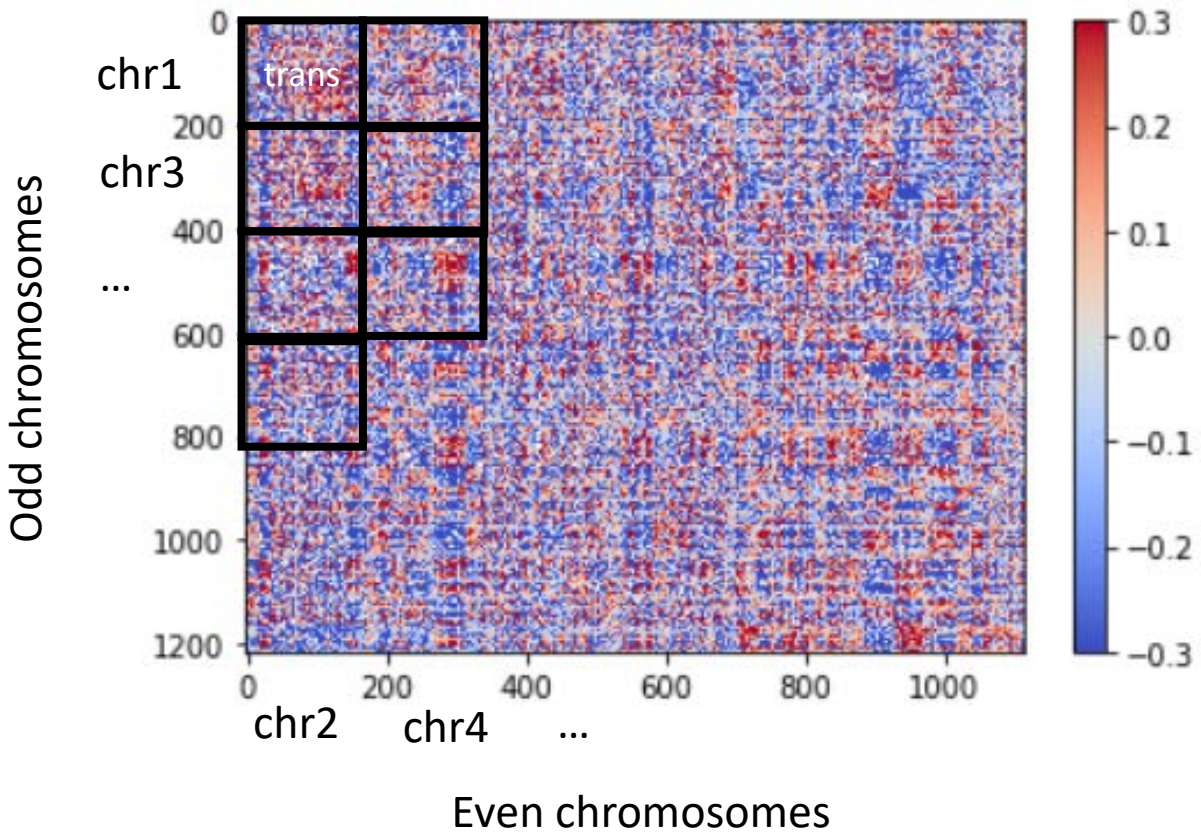
- K-means
  - k centroids
  - each point is in the cluster with the nearest centroid (means)
  - minimize variance (squared Euclidean distance) within each cluster
- Agglomerative
  - start with each point in its own cluster
  - repeatedly merge a pair of clusters by some linkage criterion (single, ward, average) until k clusters reached
- Spectral
  - create an affinity matrix, compute eigenvectors, use k-means to cluster eigenvectors



Agglomerative algorithm

(images from Wikipedia)

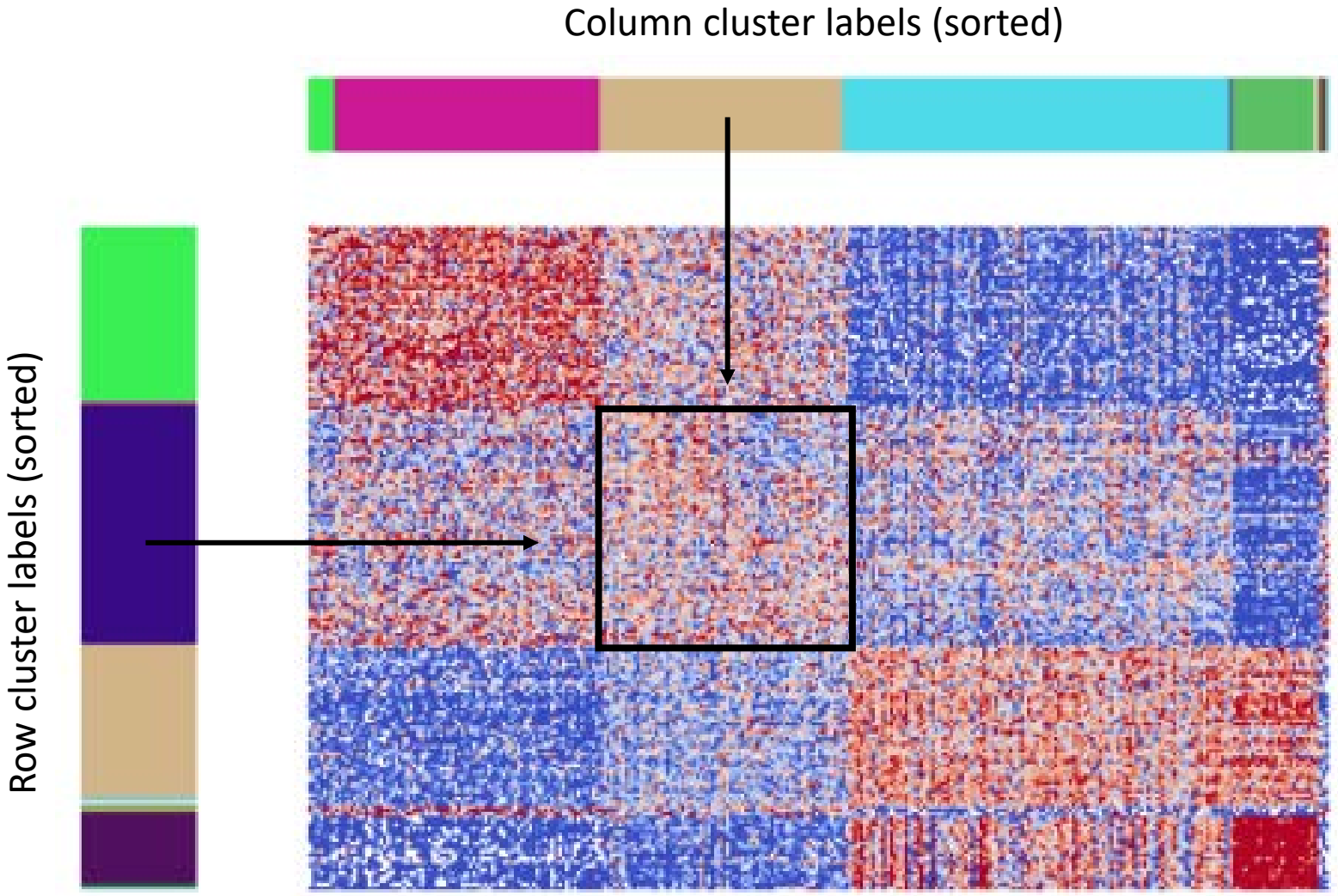
Creating a matrix for clustering



"odd-even matrix"

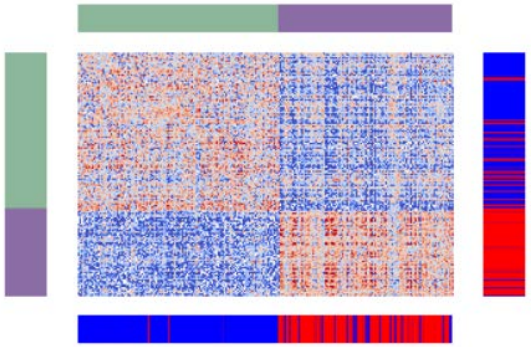
- Problem with cis interactions: scaling of contact probability and TADs interfere with compartmentalization
- Construct a matrix with all odd chromosomes vs. all even chromosomes (all **trans interactions**). Technique used in Rao et al. (2014).
- Clustering rows: treat rows as points in thousand-dimensional space and columns as dimensions
- Clustering columns: treat columns as points and rows as dimensions

Visualization of cluster labels

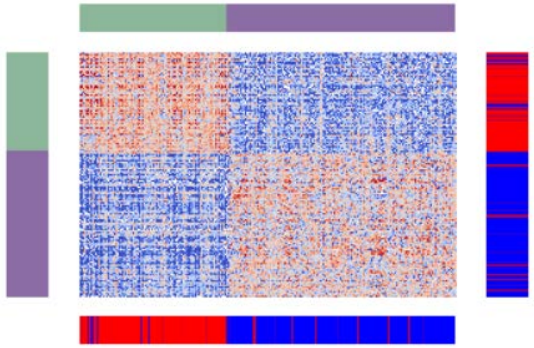


1. Run clustering procedure on rows and columns
2. Sort matrix by row and column according to cluster label
3. Plot matrix
4. Add clustering labels to the top and left

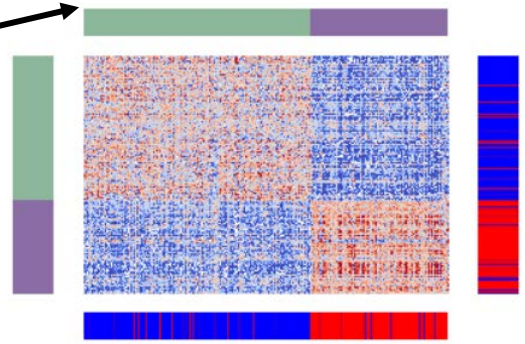
Eigendecomposition vs. clustering (k=2)



Agglomerative (average linkage)



K-means



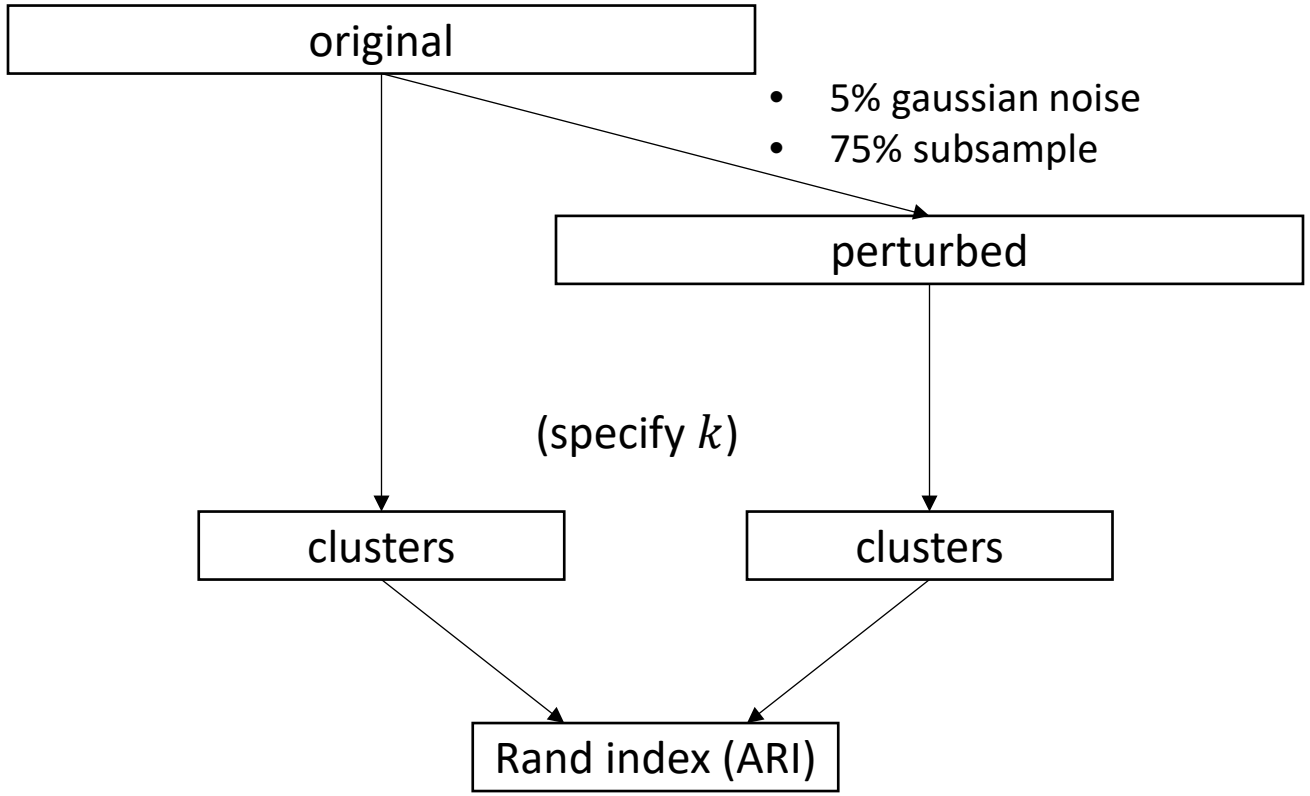
Spectral

Labels obtained from clustering

Labels obtained from eigenvector sign

- Sort by cluster label, but show eigenvector sign on bottom and right (A: +/red, B: -/blue)
- Each clustering method is able to find the two compartments

Stability



- 5% gaussian noise
- 75% subsample

(specify k)

Given: a set S of data points, a clustering algorithm $\mathcal{A}$ that takes the number k of clusters as input

(1) For $k = 2, \ldots, k_{\max}$

(a) Generate perturbed versions S_b ($b = 1, \ldots, b_{\max}$) of the original data set (for example by subsampling or adding noise, see below)

(b) For $b = 1, \ldots, b_{\max}$:
Cluster the data set S_b with algorithm $\mathcal{A}$ into k clusters to obtain clustering C_b

(c) For $b, b' = 1, \ldots, b_{\max}$:
Compute pairwise distances $d(C_b, C_{b'})$ between these clusterings (using one of the distance functions described below)

(d) Compute instability as the mean distance between clusterings C_b:

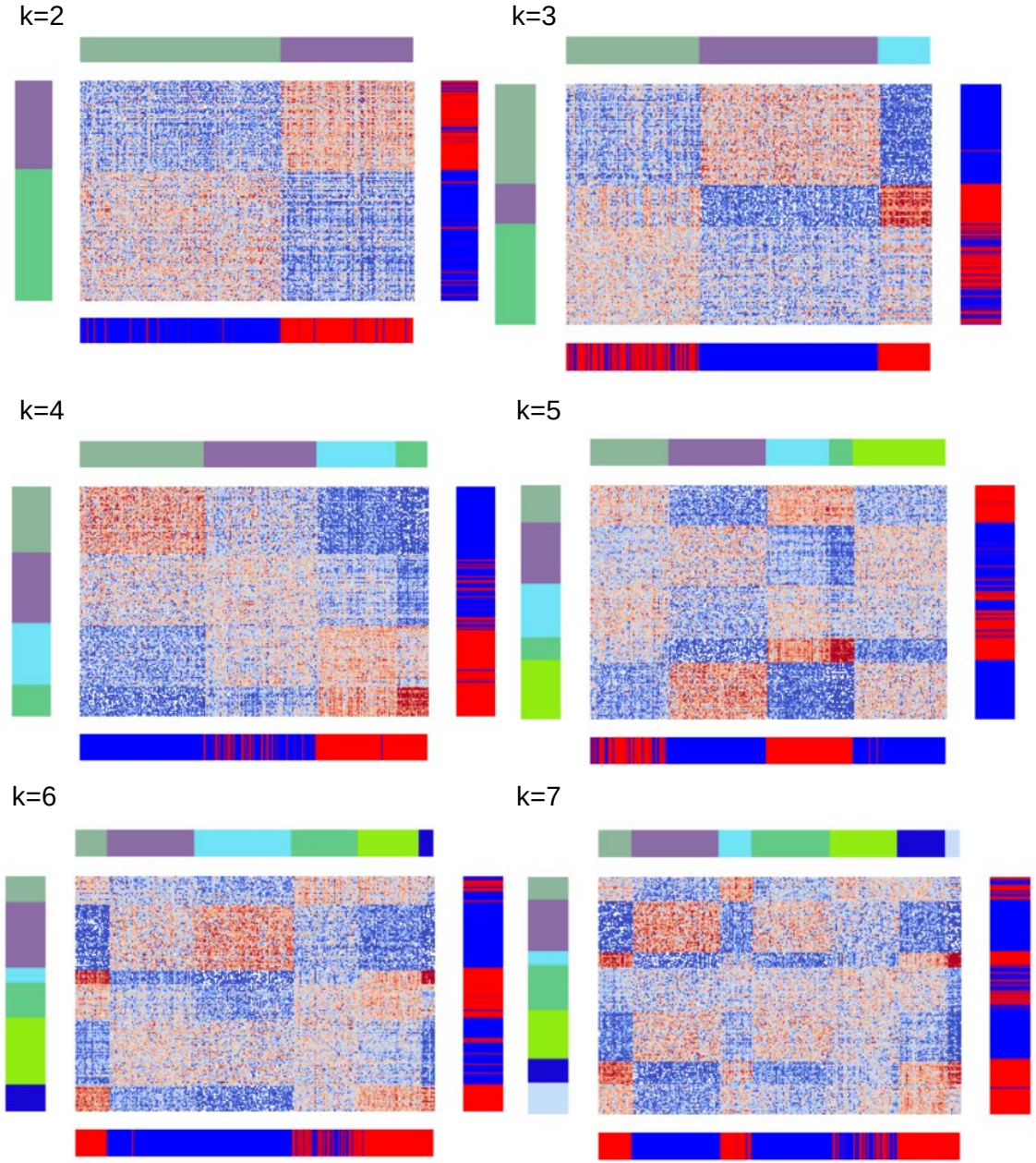
$$\widehat{\text{Instab}}(k, n) = \frac{1}{b_{\max}^2} \sum_{b,b'=1}^{b_{max}} d(C_b, C_{b'})$$

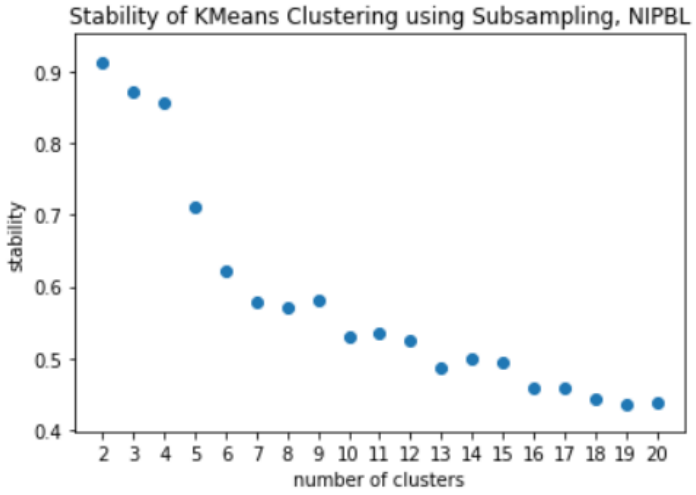
(2) Choose the parameter k that gives the best stability, in the simplest case as follows:

$$K := \operatorname*{argmin}_k \widehat{\text{Instab}}(k, n)$$

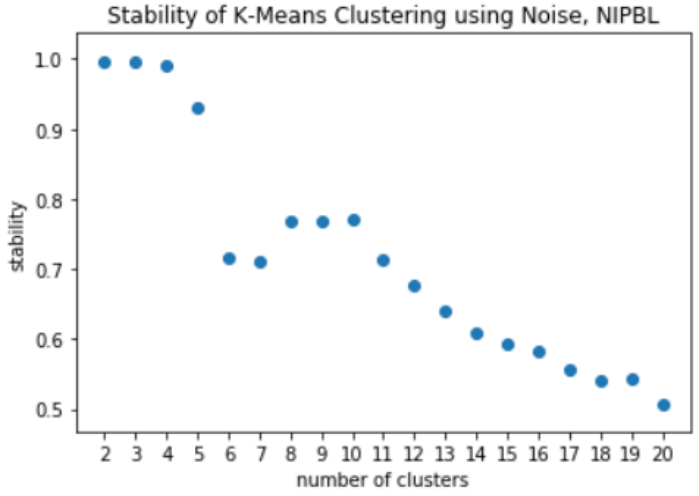
Stability metric from Luxburg (2010).

K-means





subsampling

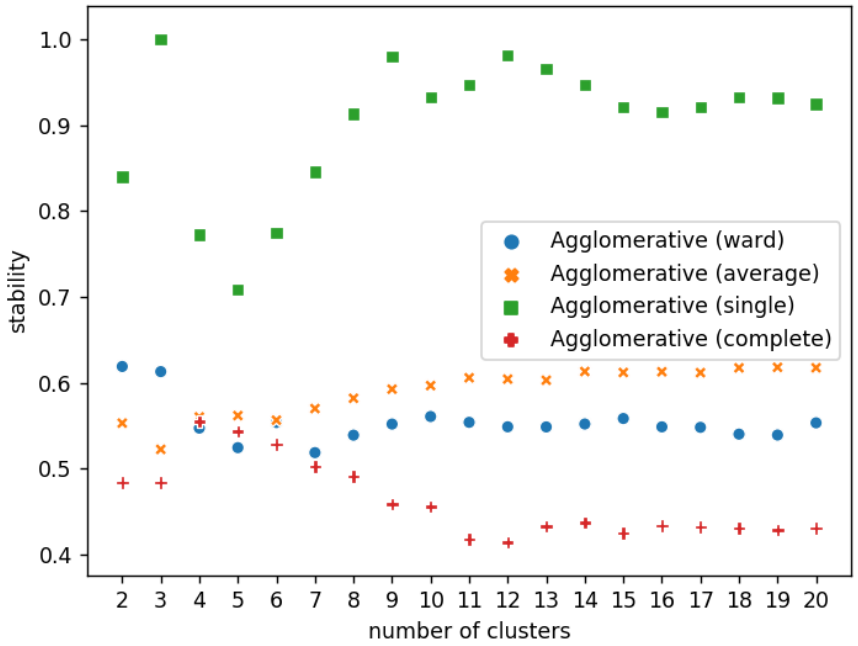


noise

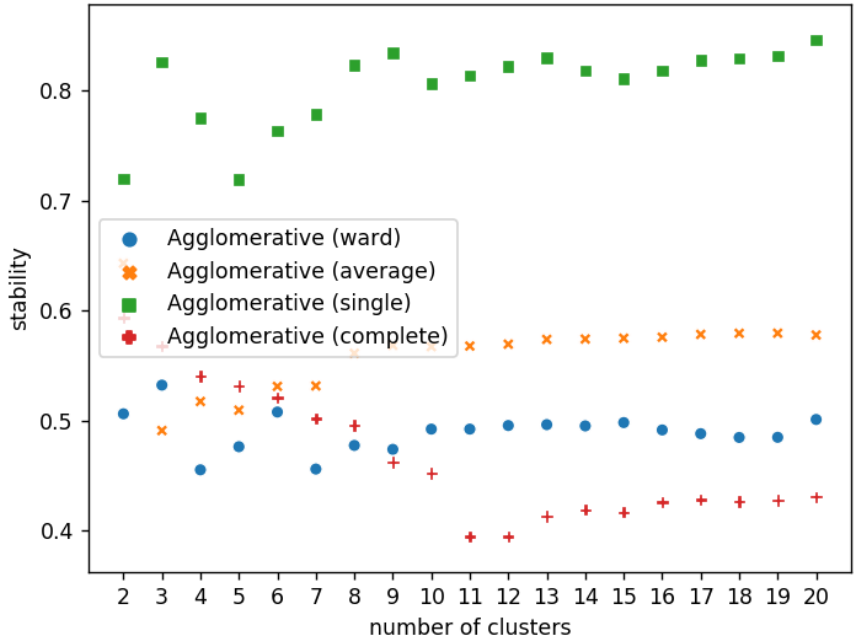
- k<=4 is most stable
- k=6, k=7 clusters look very similar

Agglomerative

- Linkage measures the distance between clusters
- Average linkage most stable
- Stability increases as k gets larger



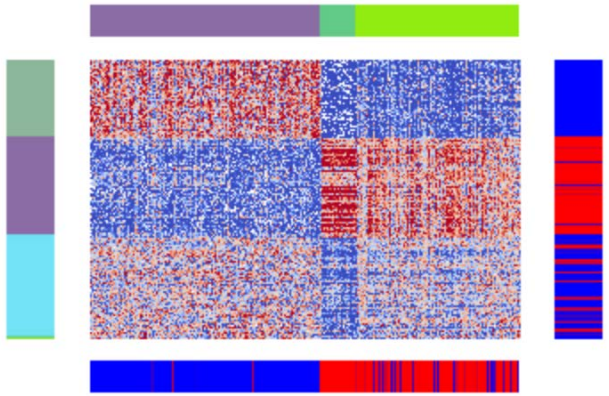
subsampling



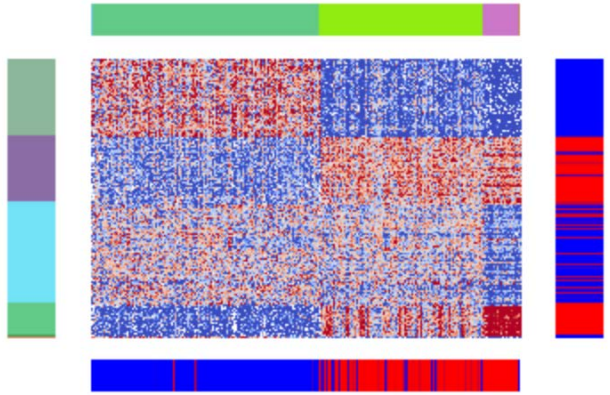
noise

Agglomerative (average linkage)

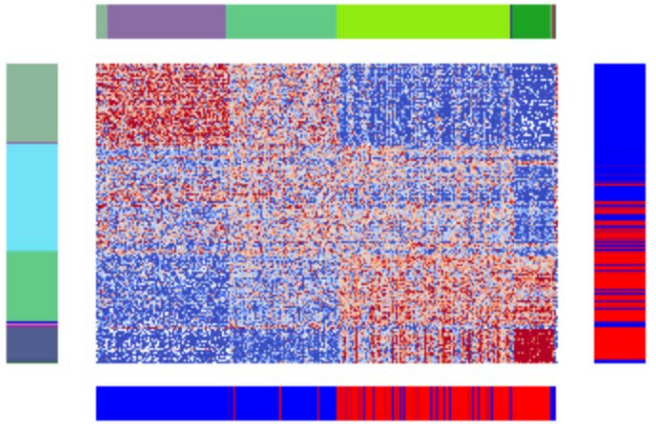
k=5



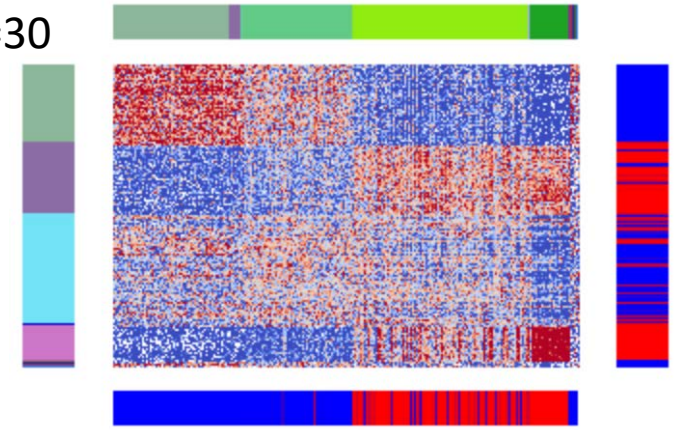
k=10



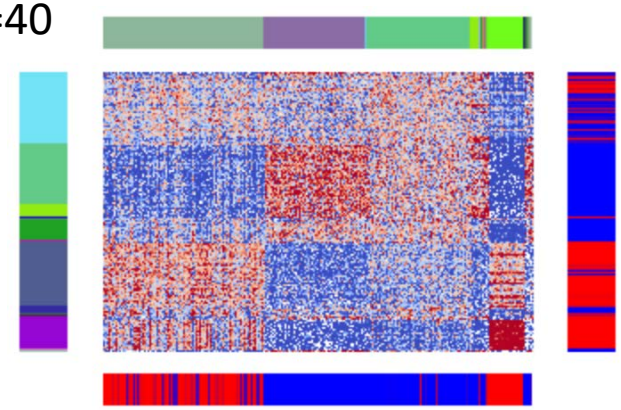
k=20



k=30



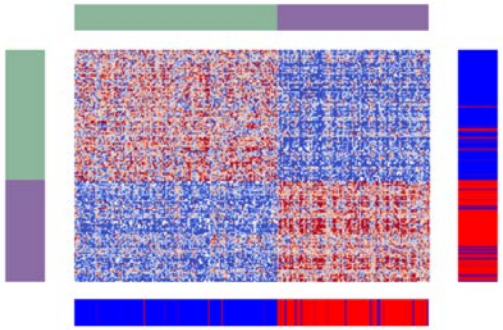
k=40



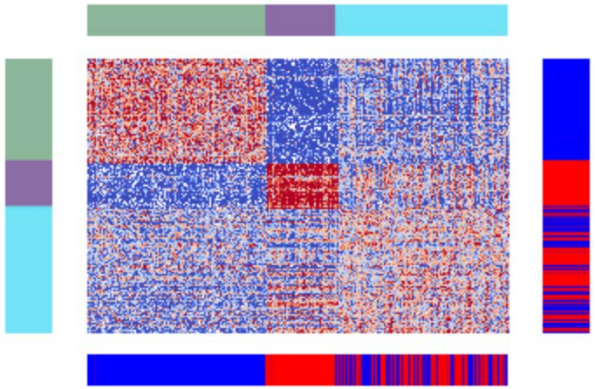
- k >= 20: four clusters found, similar to K-means with k=4

Spectral

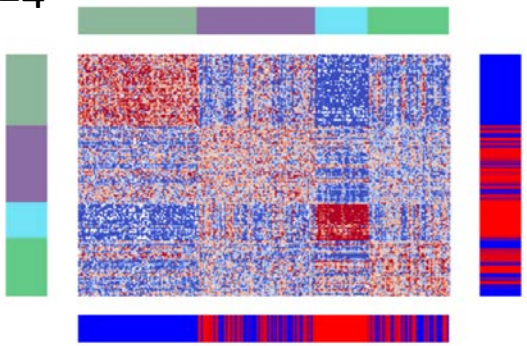
k=2



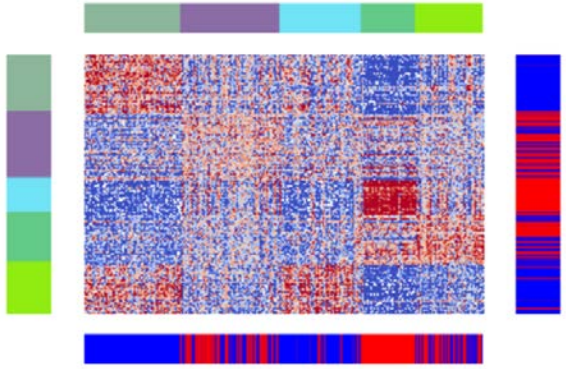
k=3



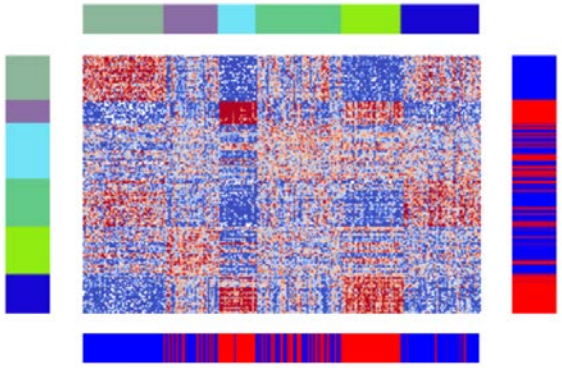
k=4



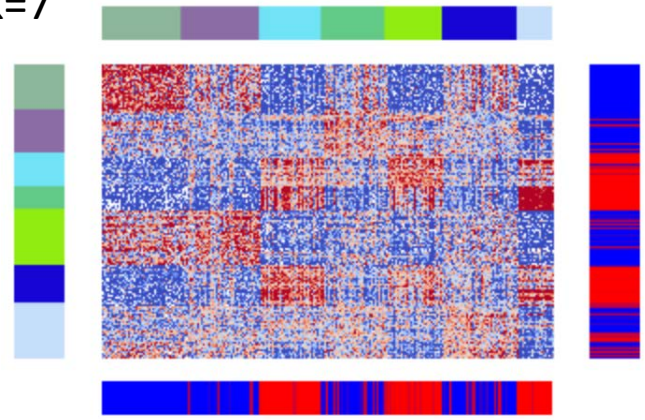
k=5

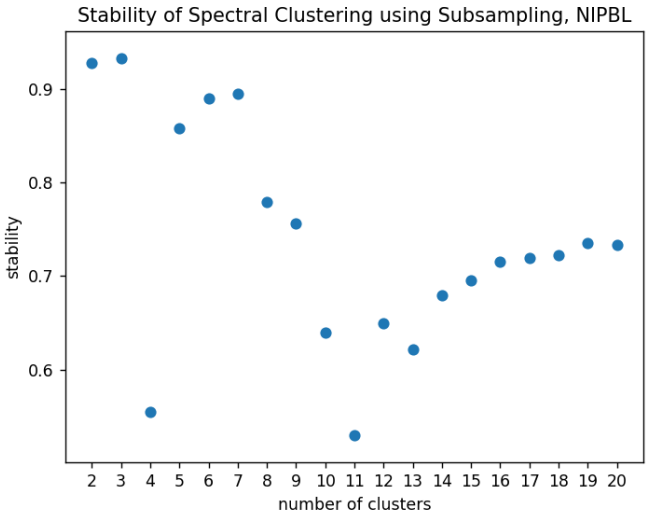


k=6

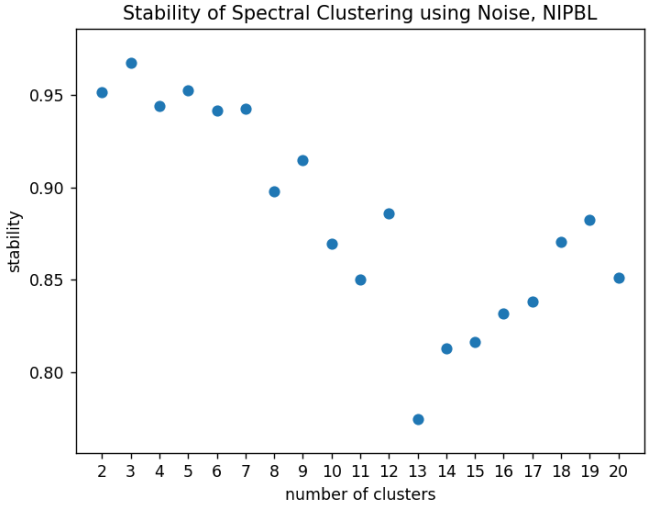


k=7





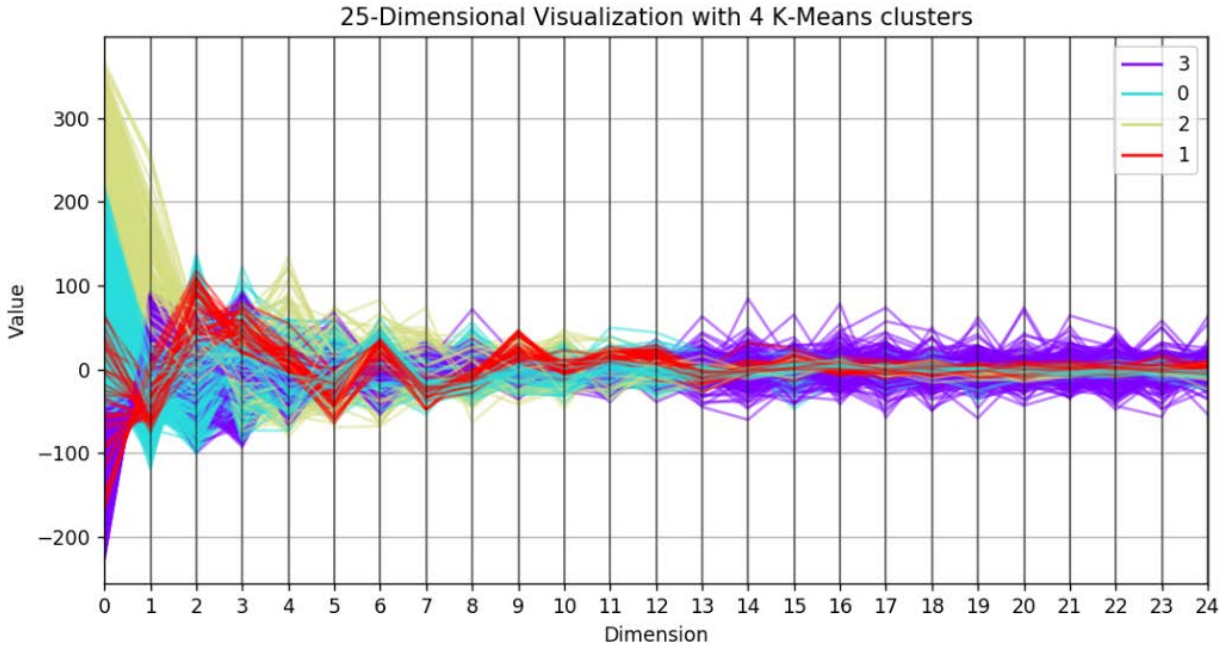
subsampling



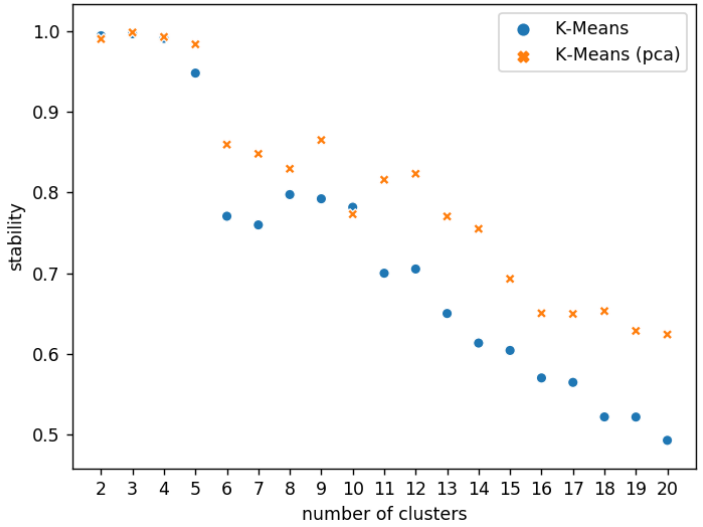
noise

- K=2 to 7 most stable
- A/B compartments not replicated

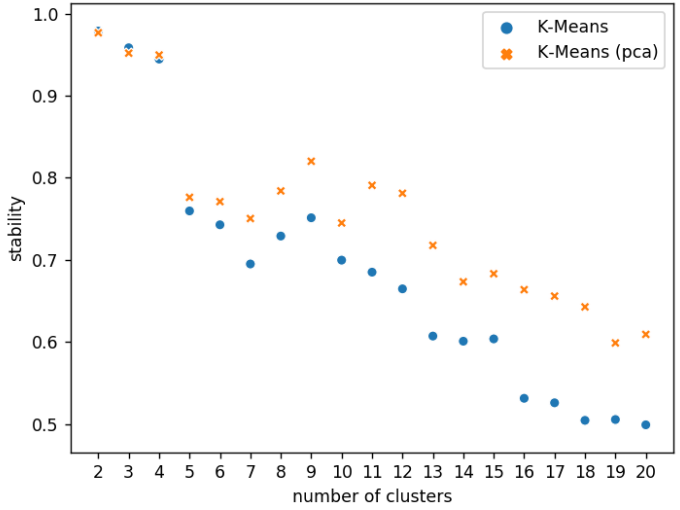
Dimensionality reduction



- Parallel coordinates plot: 4 clusters have distinct positions in 25-dimensional space (each zig-zag line represents a point)



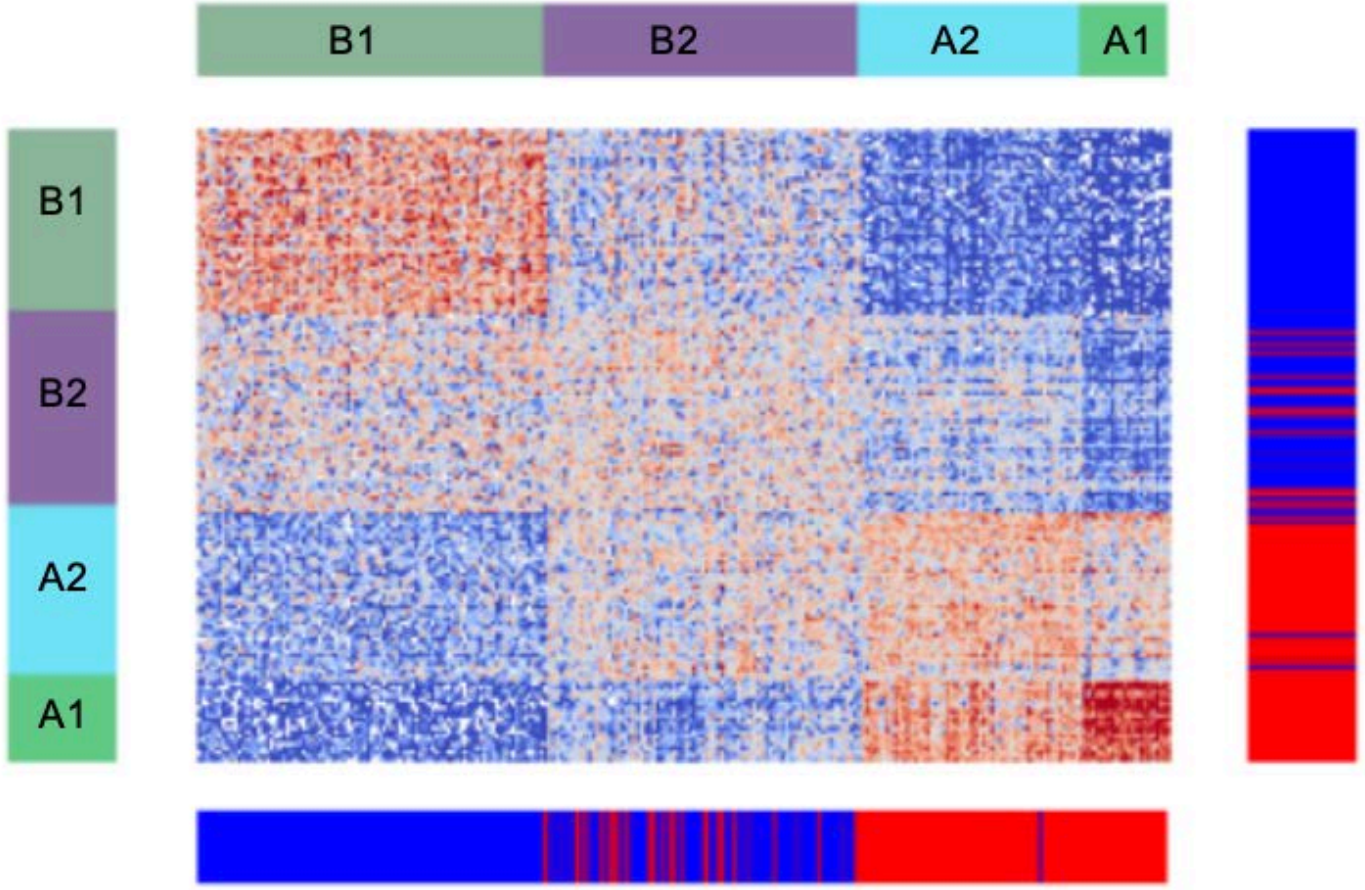
subsampling



noise

- Dimensionality reduction using principal component analysis (PCA) improves stability of K-means

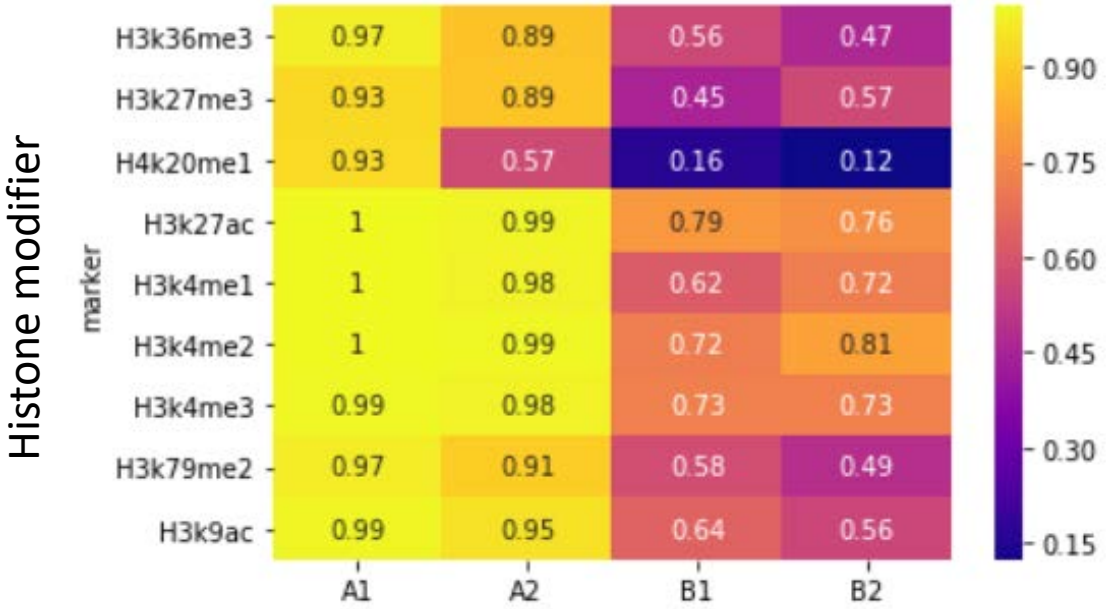
K-means (4)

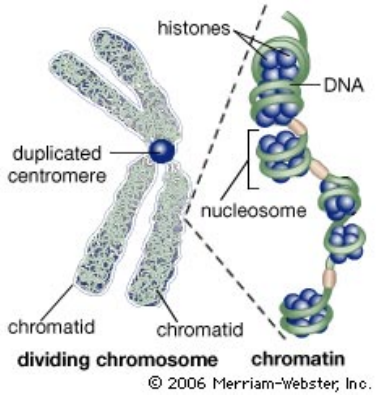


- Supported by stability analysis
- Replicates the results of agglomerative clustering

Eigenvector: Negative (B) Positive (A)

ChIP-seq

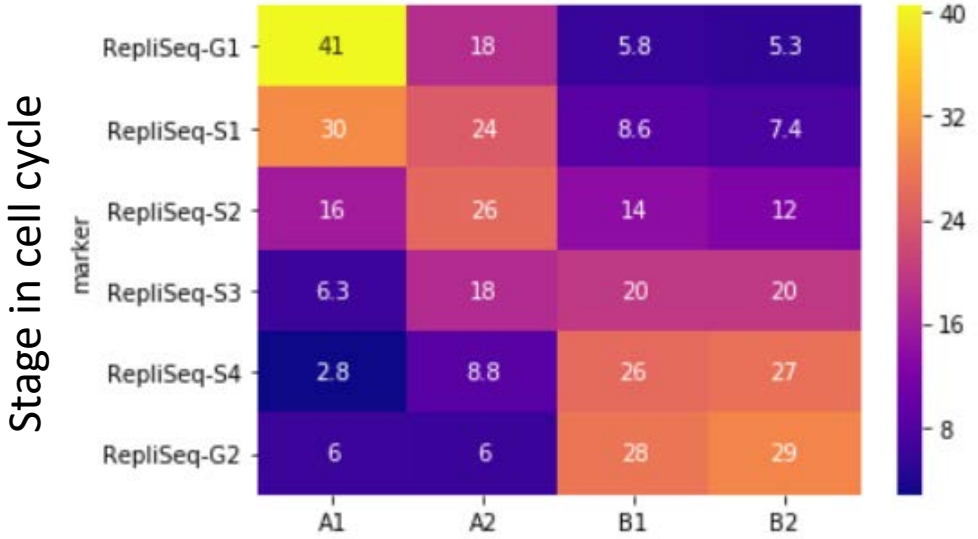




Histones (Merriam-Webster)

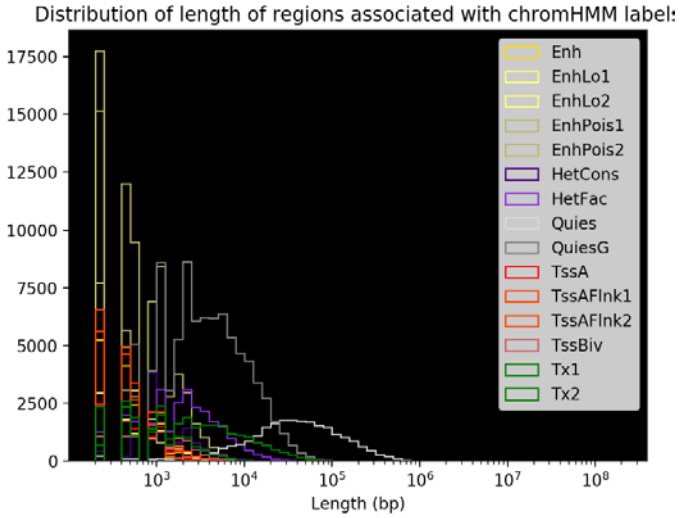
- A2, B1, B2 deficient in H4k20me1
- B regions generally deficient in all proteins

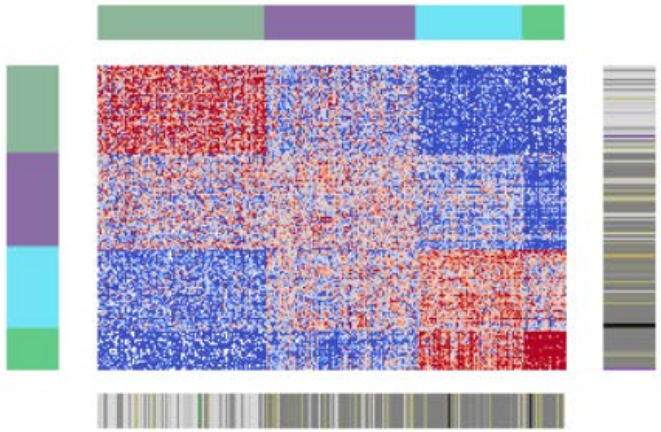
Repli-seq



- Replication order: A1, A2, B1 = B2

ChromHMM states





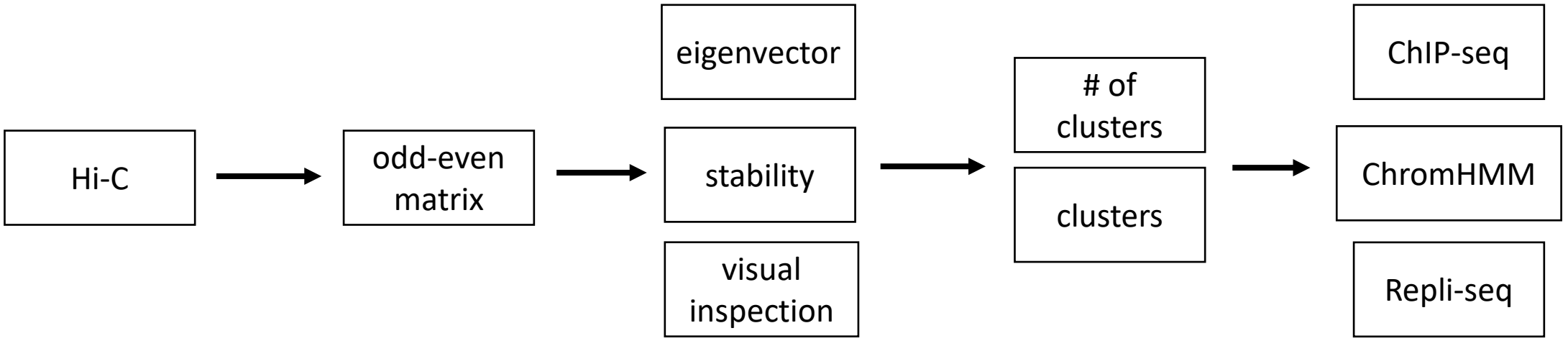
ChromHMM states

	A1	A2	B1	B2
Enh	1.1	0.56	0.077	0.31
EnhLo1	1.5	0.77	0.12	0.43
EnhLo2	0.63	0.47	0.082	0.29
EnhPois1	0.89	0.55	0.11	0.35
EnhPois2	5.9	3.7	0.53	2.1
HetCons	0.66	0.71	1.3	0.57
HetFac	10	4.8	0.58	1.9
Quies	19	43	90	69
QuiesG	37	31	5.2	19
TssA	0.77	0.36	0.066	0.24
TssAFlnk1	2.1	0.78	0.092	0.37
TssAFlnk2	0.84	0.44	0.085	0.29
TssBiv	1.4	0.66	0.16	0.37
Tx1	2.8	1.1	0.11	0.51
Tx2	15	8	1.5	4.8

- ChromHMM (hidden Markov model) running on ChIP-seq classifies DNA into 15 states (promotors, enhancers, quiescent, transcription start sites, etc.)
- B1 mainly made up of Quies state (light gray)

Conclusion

- Demonstrated that there are four nuclear subcompartments with distinctive features
- Framework for clustering Hi-C data



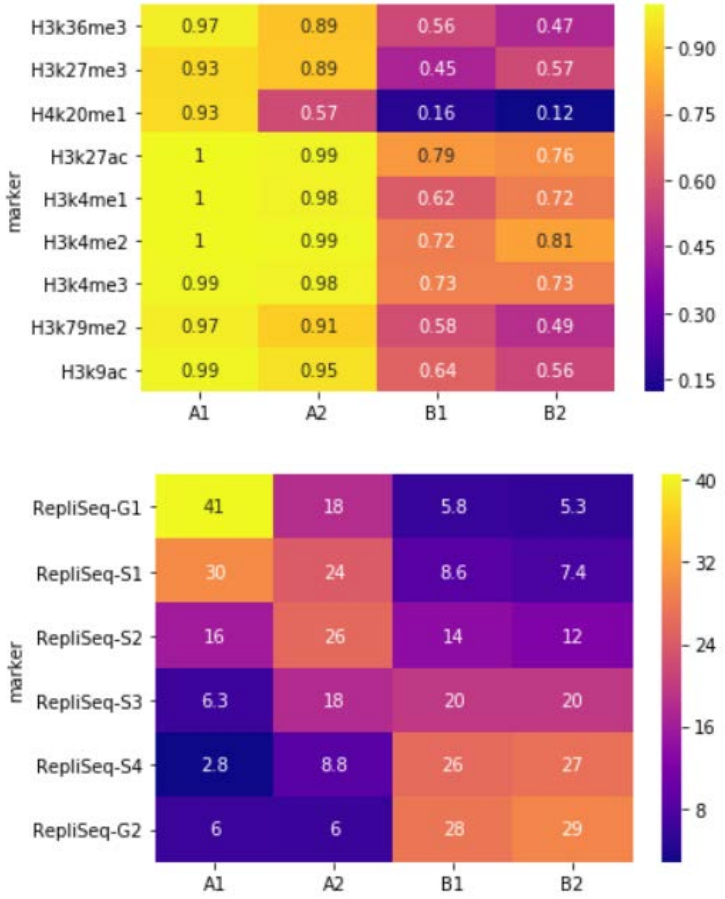
Future Work

- Compare to physics models of chromatin
- Write software pipeline to find the compartments (A1, A2, B1, B2) given any Hi-C matrix
- More sophisticated graph-based clustering techniques (simulated annealing)
- More fine-grained analysis with resolution higher than 1mb

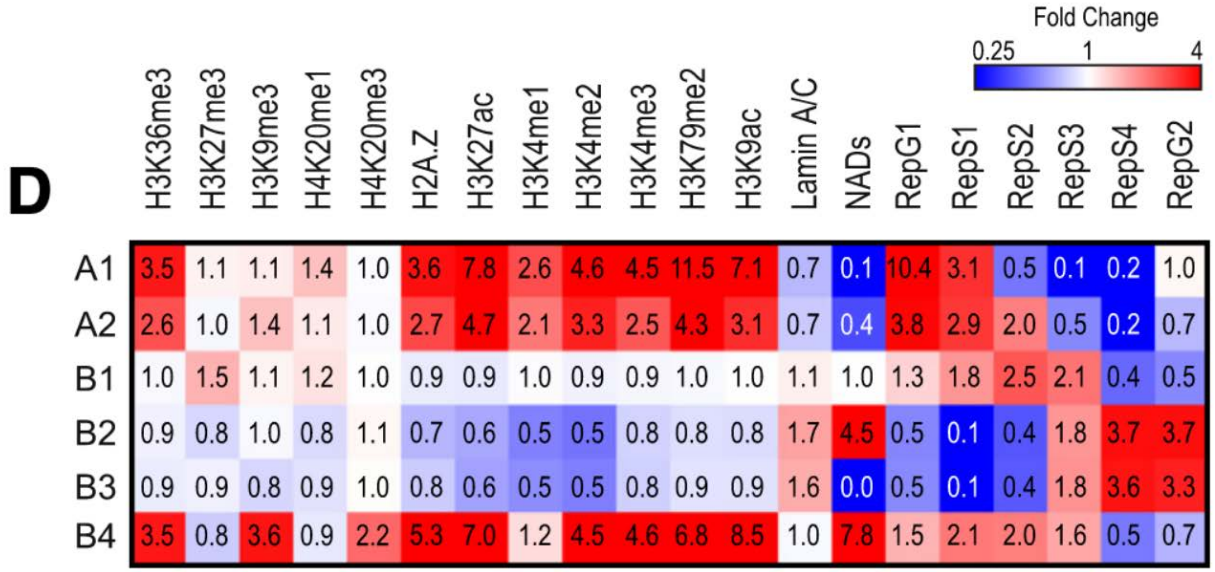
Acknowledgements

- ENCODE project
  - Hi-C matrices, ChIP-seq, Repli-seq, ChromHMM
- Sameer Abraham for mentoring me
- Martin Falk and Prof. Leonid Mirny for inviting me to MIT MirnyLab and providing me with computational resources
- Dr. Slava Gerovitch and Prof. Srini Devadas for inviting me to MIT PRIMES-CS

ChIP-seq/Repli-seq enrichments compared to Rao et al. clusters

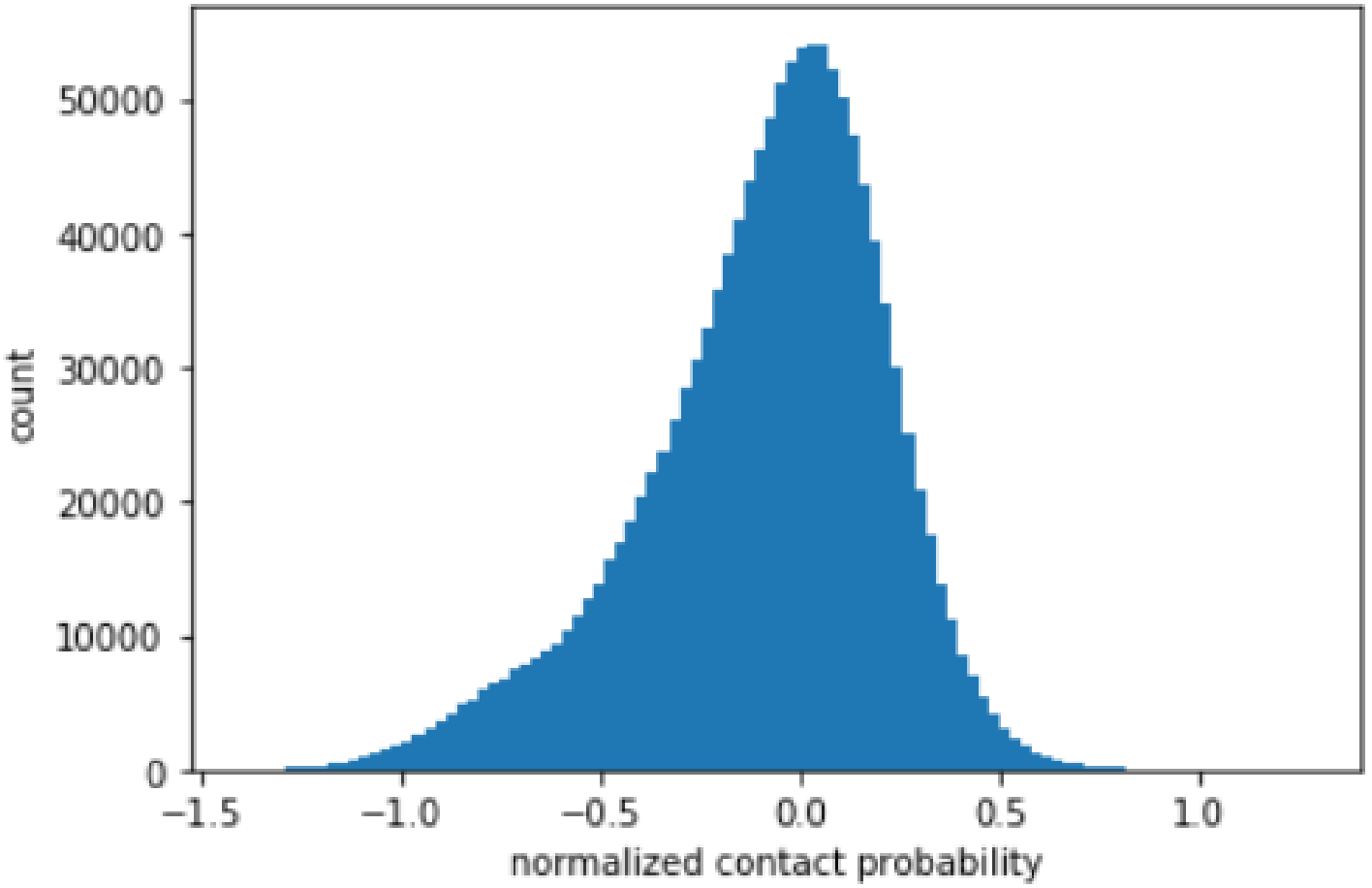


Our subcompartments



Rao et al. (2014)

Distribution of logged trans contact probabilities



Adjusted Rand Index (ARI)

- X and Y are two clusterings of the same set. X_i and Y_i each represent a cluster.
- ARI measures similarity between the two clusterings

$X \backslash Y$	Y_1	Y_2	$\dots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Sums	b_1	b_2	$\dots$	b_s	

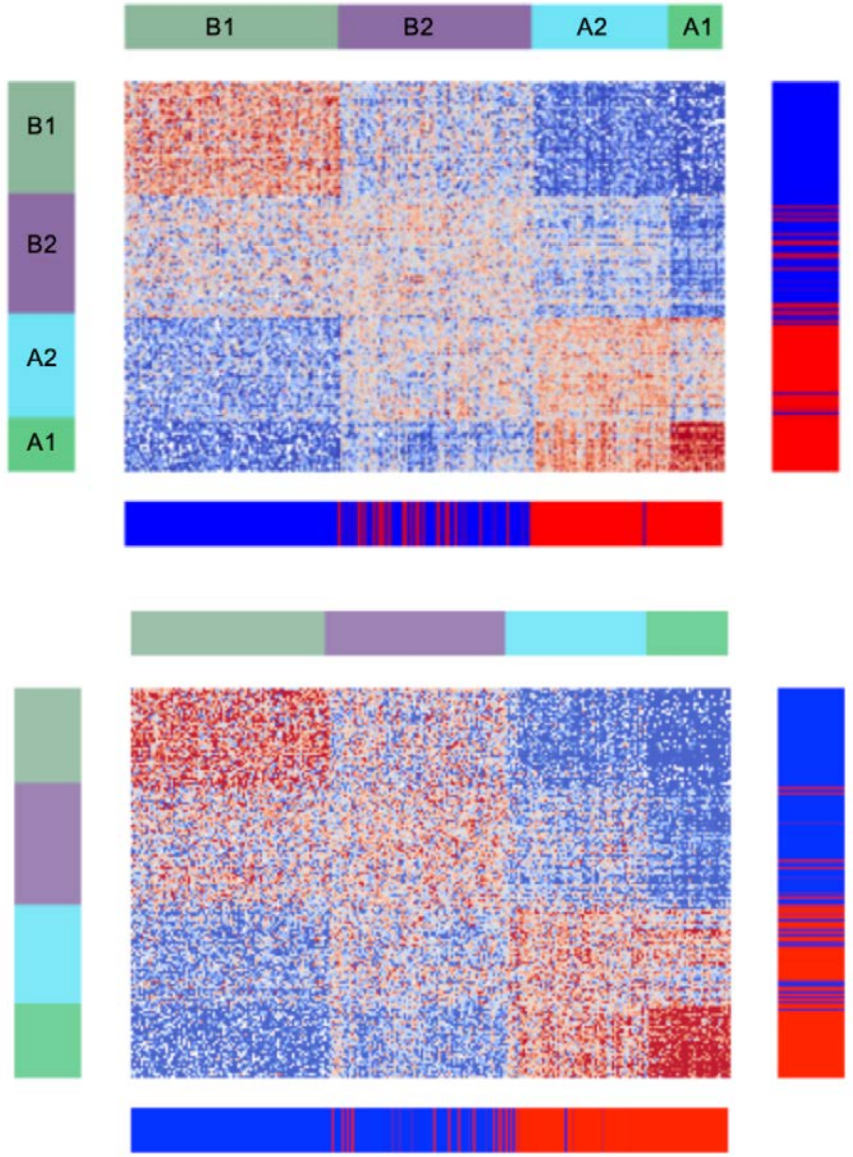
Contingency table (Wikipedia)

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

ARI formula (Wikipedia)

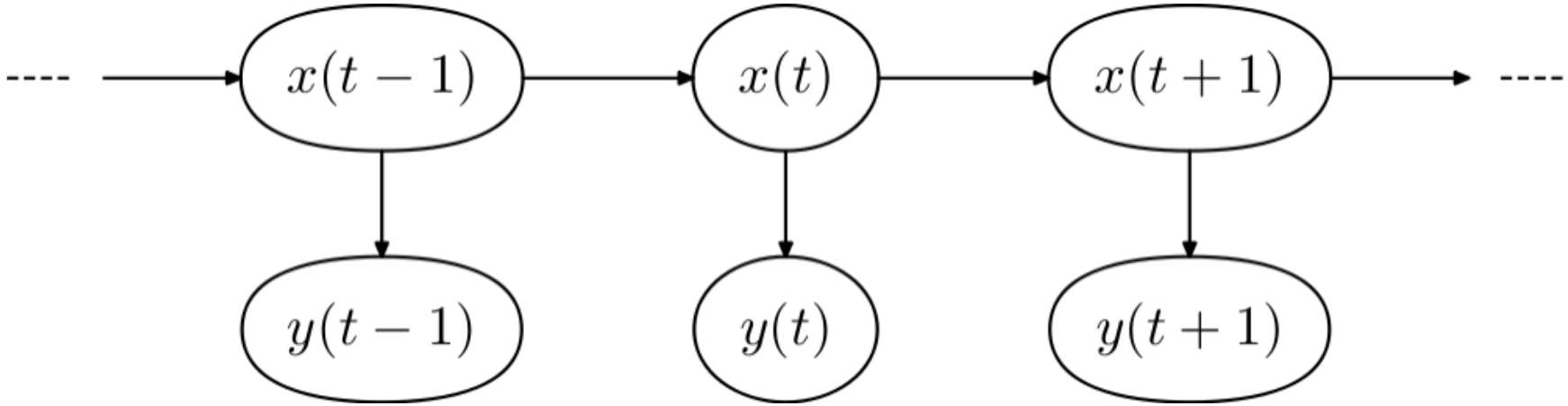
NIPBL vs. Untreated (UNTR)

- Mouse liver cells
- NIPBL is a cohesion loading protein thought to be responsible for loop extrusion



ChromHMM

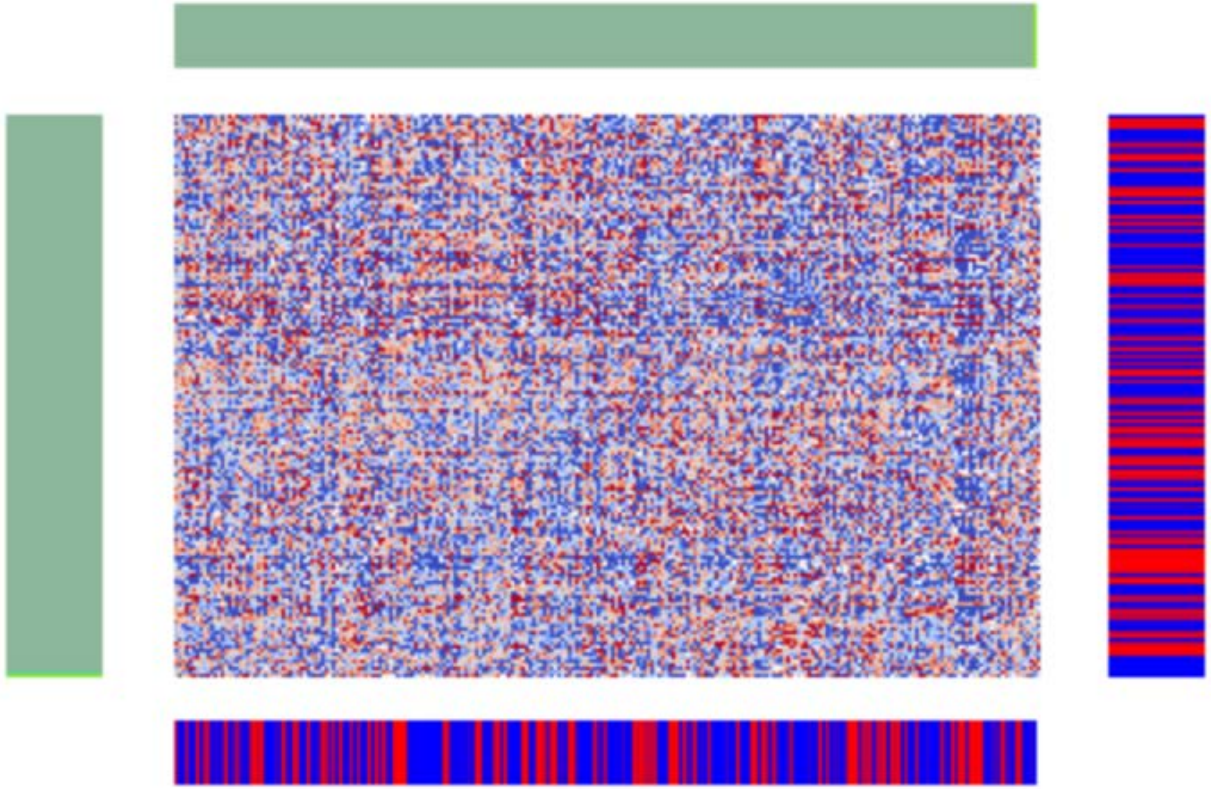
Specify number of chromatin states



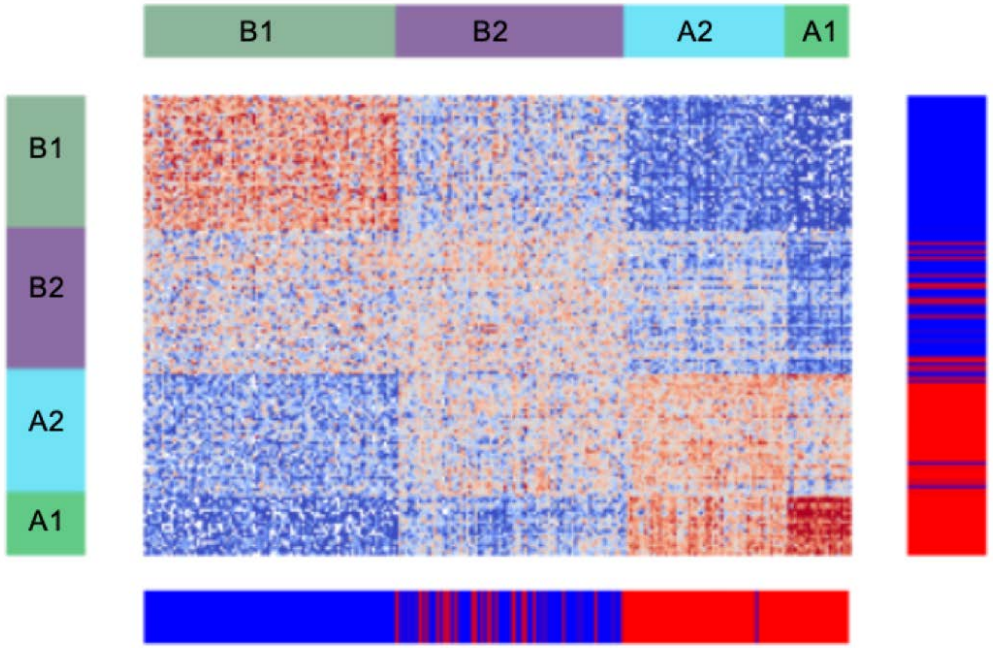
Observations: ChIP-seq tracks

(Wikipedia)

Agglomerative (single linkage, k=5)

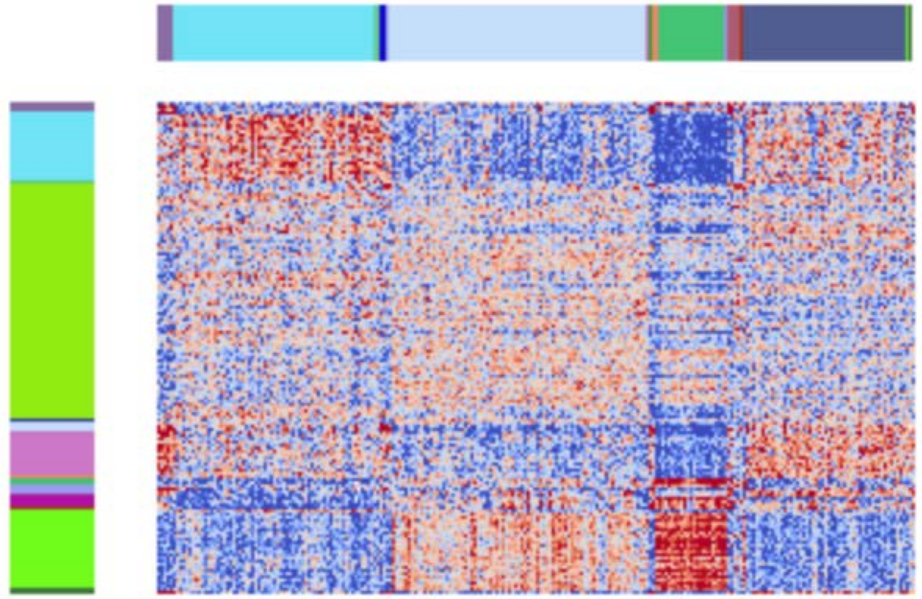


Comparison with human cells



NIPBL factor removed

Mouse liver



GM12878 (Human cell line)

Effect of Dimensionality Reduction