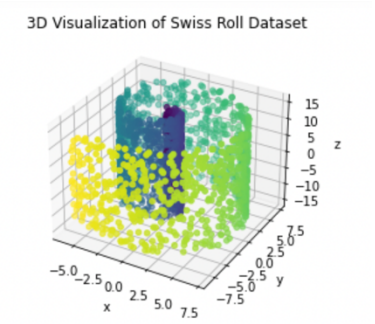
CPSC 453 Problem Set 1: Dimensionality Reduction via PCA and Diffusion Maps

Author: Wenxin Xu

The `xu_wenxin_ps1.zip` file includes 3 files:

- `xu_wenxin_ps1_report.pdf` : A detailed report.
- `/code`
  - `ps1_functions.py`: contains 5 pre-defined and 1 new-defined functions
  - `xu_wenxin_ps1.jpynb`: A Jupyter Notebook contains all the code.
- `/figures`: contains all 27 figures used in the report.

2 Understanding the Data Set



- Swiss Roll Dataset has 2000 samples with 3 features
- `swiss_roll_points.json` has 3 features for 2000 samples
- `swiss_roll_labels.json` has labels ∈[0,1] for 2000 samples
- 3D Visualization of Swiss Roll Dataset

Q 2.1

- *What does this visualization of the swiss roll data set look like?*

  It looks like a coiled roll.

- *What properties do you notice about the data set?*

  The data set has a geometrically curly shape.

- *Are the provided labels meaningful, and if so, in what way?*

  Yes, There is a color gradient from the inside axis (x = 0, y = 0) of Swiss Roll to the outside of Swiss Roll corresponding to the density of data points.

- *How do you expect a "good" dimensionality reduction technique to look for the swiss roll data set?*

  It should uncoil the roll to a surface. Maybe a nonlinear method.

2.1 Visualizing Data with PCA

1. Run PCA on data, obtaining principal components, projections, singular values

Principal components are contained in a 2000 x 2000 matrix, some of its rows are:

```
[[ 1.03328966e-02 -1.56763009e-02 -3.89834583e-02 ...  4.11357786e-03
   6.19227758e-03 -1.82873106e-03]
 [-2.67650480e-02  2.98899809e-03 -4.33774440e-02 ... -5.63836492e-03
  -3.65167227e-02 -5.58247322e-03]
 [ 3.89500923e-02 -1.59988783e-02 -3.73832481e-02 ...  2.55927892e-02
  -5.27132138e-03 -3.48888370e-02]
 ...
 [-2.58745484e-02  4.71207684e-03  3.52396051e-03 ...  9.99330181e-01
  -7.93872361e-05  8.31781644e-04]
 [ 3.38248084e-03  3.72323073e-02  2.07696868e-03 ... -8.00242655e-05
   9.98595765e-01 -3.98397666e-04]
 [ 3.45815521e-02  7.17130648e-03 -2.52673134e-03 ...  8.32141078e-04
  -4.02014774e-04  9.98786424e-01]]
```

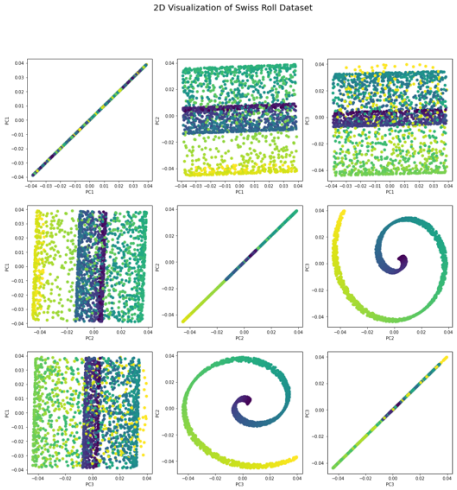
Projections are contained in a 3 x 3 matrix:

```
[[   5.2167858    -8.82683981 -165.36563411]
 [  -7.84488874  193.80358133   -7.57464212]
 [ 388.49038021    4.03205676    2.06763129]]
```

We have 3 singular values: `[388.60459679 194.04638291 165.55193519]`

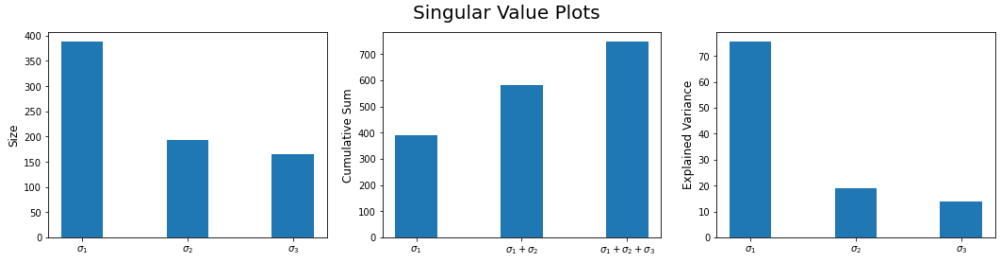
2. Plot data in 2 dimensions using principal components, colored by labels.

Here I try three different combinations of 1st and 2nd principal components, 1st and 3rd principal components, 2nd and 3rd principal components, respectively.



3. Plot singular values.

   Here I plot singular values, cumulative sum of singular values, and explained variance.



Q 2.2

- *As a dimensionality reduction technique, to what extent does PCA retain properties of swiss roll dataset?*

  PCA performs a bad job of uncoiling the swiss roll, i.e. explaining the density gradient. A plot of PC1 VS. PC2 or a plot of PC1 VS. PC3 tried to uncoil the swiss roll but different bands overlapped. While a plot of PC2 VS. PC3 failed to uncoil the swiss roll.

- *Can you explain why the visualizations look like this, given how the algorithm works?*

  Because PCA is a linear transformation. A plot of PC1 VS. PC2 is a projection of swiss roll in $\mathbb{R}^3$ onto yz-plane. A plot of PC1 VS. PC3 is a projection of swiss roll in $\mathbb{R}^3$ onto xz-plane. A plot of PC2 VS. PC3 is a projection of swiss roll in $\mathbb{R}^3$ onto xy-plane. So there are many overlapped colored points in the first two plots. While the third plot is a cross section removing the z-dimension.

- *What can you learn about the intrinsic dimensionality from the singular values?*

  By looking at the 3 plots of singular values, we can see the first singular value explained the most variance of data, in that way the properties (meaningful part) of data is only located in one dimension, it makes sense because after we uncoil the swiss roll to a rectangular band, we could project the band onto xy-plane, then the projected band is a line of color gradient, which is in 1D.

3 Implementing Diffusion Maps

All six functions for implementing diffusion maps are in Python script `ps1_functions.py` :

`load_json_files()`: loading JSON data into Numpy Arrays

`compute_distances()`: creating Euclidean distance matrix from dataset

`compute_affinity_matrix()`: creating an affinity matrix from a distance matrix via a Gaussian kernel

`diff_map_info()`: creating info for easily create diffusion map at t

`get_diff_map()`: creating a diffusion map at t from eigenvalues and eigenvectors of Markov Matrix

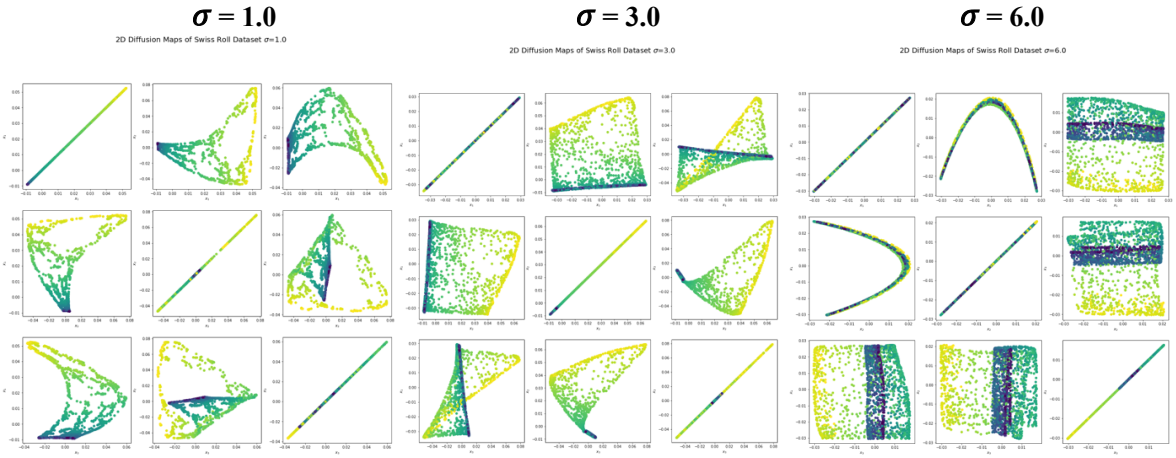
Note: I further create a new function `diff_map()` to wrap up all the four functions above. Computing diffusion maps of data, return diffusion maps, nontrivial eigenvectors and eigenvalues of Markov matrix.

4 Experiment 1: Swiss Roll Dataset

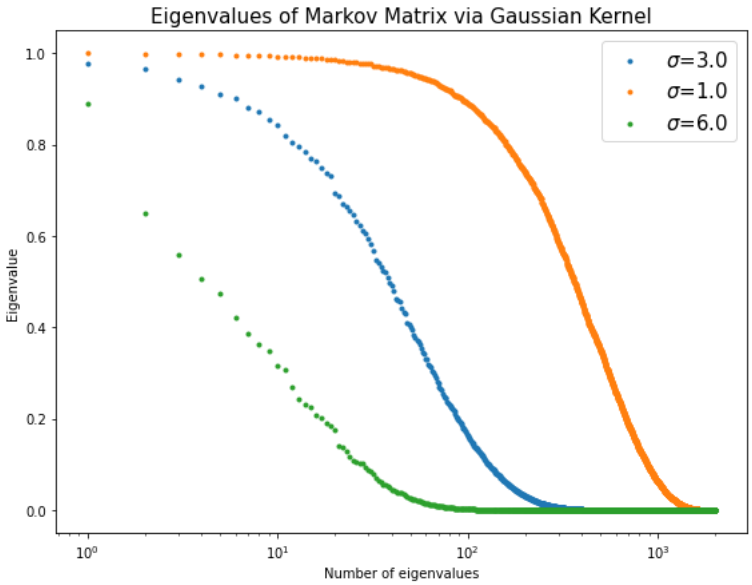
4.1 Visualizing Data with Diffusion Maps

For σ = 1.0, 3.0, 6.0

1. Create 2D scatterplots of diffusion mapping using different coordinates. Here I try three different combinations of 1st and 2nd coordinates, 1st and 3rd coordinates, 2nd and 3rd coordinates, respectively.



2. plot eigenvalues λi of Markov matrix M



Q 4.1

- *As a dimensionality reduction technique, to what extent does diffusion mapping retain properties of swiss roll dataset?*

  Diffusion mapping perform a better job than PCA to uncoil the swiss roll because plot of $x1$ VS $x2$ almost made it. It retains the non-linear, manifold shape of swiss roll.

- *Can you explain why visualizations look like this, given how algo works?*

  The first non-trivial eigenvector DM1 tracks the most prominent non-linear path. So the 2D visualizations can uncoil the swiss roll to different extent. plots of $x1$ VS $x2$ give a 1D manifold while plot of $x1$ VS $x3$ or plot of $x2$ VS $x3$ still give a 2D manifold.

- *What can you learn about intrinsic dimensionality of dataset from eigenvalues of M?*

  From the plot of eigenvalues of M given different σ, we can see there is a sharp decrease of eigenvalues when the number of eigenvalues approaches to 100.

- *How does choice of Gaussian kernel width σ change embedding and why?*

  Among different choices of σ = 1.0, 3.0, 6.0, σ = 3.0 performed the best job of uncoiling swiss roll. Because σ controls distance between two points in diffusion maps. When σ is too small or too large, both diffusion mapping can't capture neighbors of a point.

5 Understanding the First Eigenvector of Markov Matrix

1. Construct affinity matrix of swiss roll dataset using Euclidean distance and Gaussian kernel with width σ=1.0

   Affinity matrix is a 2000 x 2000 square matrix:

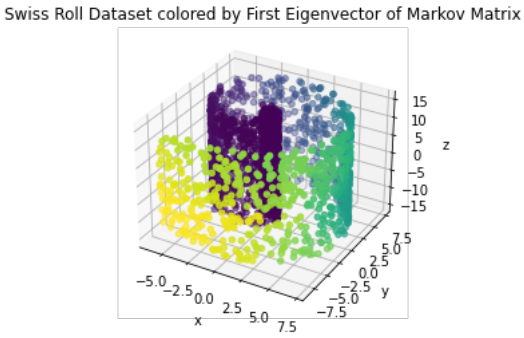
```
[[1.00000000e+000, 6.48625588e-097, 1.81101234e-054, ...,
  5.19391203e-115, 9.70181706e-070, 1.20248473e-063],
 [6.48625588e-097, 1.00000000e+000, 2.83005554e-290, ...,
  5.20153433e-027, 4.20719732e-104, 1.07190655e-267],
 [1.81101234e-054, 2.83005554e-290, 1.00000000e+000, ...,
  2.92713538e-303, 1.44914199e-148, 3.21692731e-025],
 ...,
 [5.19391203e-115, 5.20153433e-027, 2.92713538e-303, ...,
  1.00000000e+000, 3.48589386e-074, 5.73425846e-241],
 [9.70181706e-070, 4.20719732e-104, 1.44914199e-148, ...,
  3.48589386e-074, 1.00000000e+000, 1.35382140e-079],
 [1.20248473e-063, 1.07190655e-267, 3.21692731e-025, ...,
  5.73425846e-241, 1.35382140e-079, 1.00000000e+000]]
```

2. Compute largest left eigenvector $\phi 1$ of M=D^{-1}W, maybe obtained as $\phi 1$=$v 1 D^{1/2}$

   Largest left eigenvector is a vector ∈ R^{2000}

```
[0.03508958,  0.02482031,  0.03590893, ..., -0.00898086,
0.00565618, -0.00875206]
```

3. Plot swiss roll in 3D using original coordinates and color the points using corresponding values in $\phi 1$



Q 5.1

- *How do values in $\phi 1$ correspond to structure of swiss roll ?*

  The gradient of values in $\phi 1$ is perfectly consistent with structure of swiss roll.

- *Can you explain what you are seeing in terms of diffusion?*

  The points are diffusing from the central axis of swiss roll like a swirl, as they spread farther and farther, the density of them also becomes smaller and smaller. First non-trivial eigenvector DM1 tracks the most prominent non-linear path

5.1 Using an Adaptive Gaussian Kernel

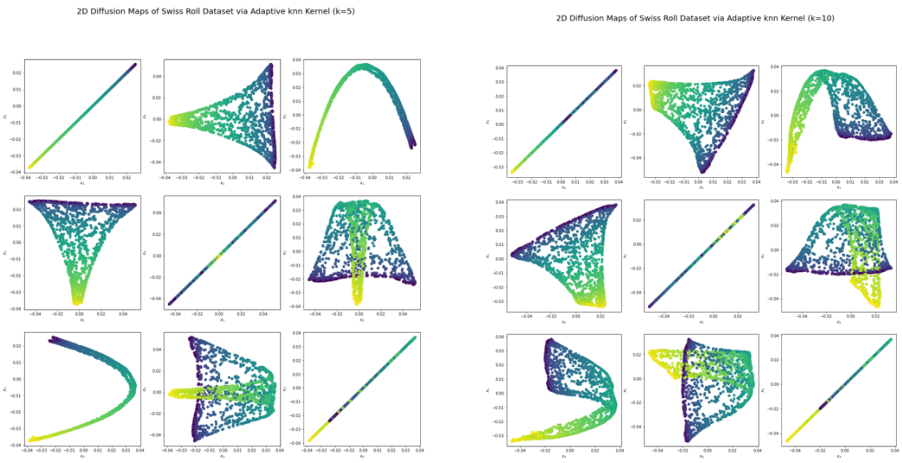
Fixing diffusion param t=1, do experiments for nearest neighbor parameter k=5 and 10

1. Create diffusion map Ψt of swiss roll dataset using Euclidean distance, adaptive k-nearest neighbor Gaussian kernel.

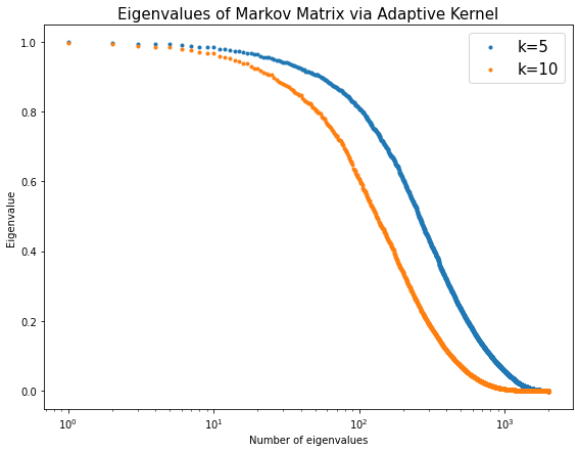
2. Create 2D scatter plots of diffusion mapping using different coordinates.

Here I try three different combinations of 1st and 2nd coordinates, 1st and 3rd coordinates, 2nd and 3rd coordinates, respectively.

$k = 5$ $k = 10$



3. Plot eigenvalues λ_i of Markov Matrix M



Q 5.2

- *What are differences between fixed and adaptive choices of kernel parameters σ?*

  Differences are in density calculations.

- *Explain the difference.*

  Given fixed kernel width σ, diffusion mapping estimates density globally by controlling the bandwidth of stretch of points. Given adaptive kernel width σ controlled by number of nearest neighbors k, diffusion mapping estimates density locally.

- *What can you learn about intrinsic dimensionality of dataset from eigenvalues of M ?*

  The intrinsic dimensionality of the data set is greater than 5.

- *How does choice of nearest parameter k change diffusion map?*

  k = 10 performs better than k = 5 on uncoiling the swiss roll

- *Which kernel method would you recommend using for swiss roll dataset and why?*

  I recommend fixed Gaussian kernel. Because from the plots $x1$ VS. $x2$ we can see, fixed Gaussian kernel performs better than adaptive Gaussian kernel, the former uncoiled the swiss roll to a band with nearly even width while the latter uncoiled the swiss roll to a band with triangle shape (i.e., uneven width) which means the former can capture global structure of swiss roll dataset better.

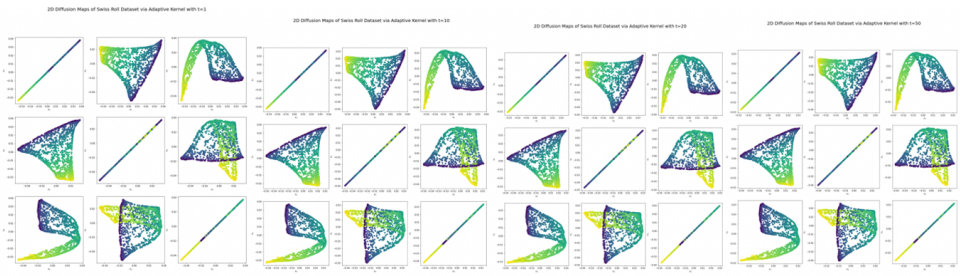
5.2 Changing the Diffusion Parameter t

Fixing k=10, do experiments for diffusion param t=1, 10, 20, and 50

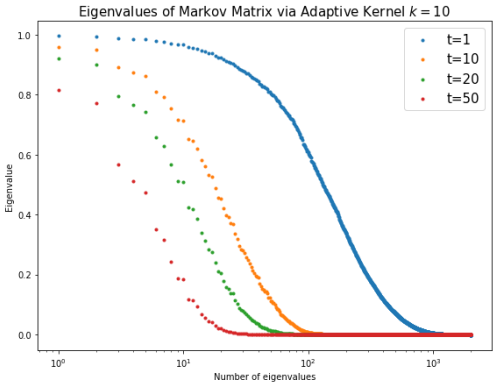
1. Create diffusion map Ψt of swiss roll dataset using Euclidean distance, adaptive k-nearest neighbor Gaussian kernel.
2. Create 2D scatter plots of diffusion mapping using different coordinates.

Here I try three different combinations of 1st and 2nd coordinates, 1st and 3rd coordinates, 2nd and 3rd coordinates, respectively.

t =1	t = 10	t = 20	t = 50



3. Plot eigenvalues λ_i of Markov matrix M.



Q 5.3

- *In terms of diffusion processes, what's interpretation of increasing t?*

  t is the number of random walk steps, as t increases, the diffusion area of will increase random walk will be around focus in a specific area and restrict more embedding mapping of the observations.

- How do diffusion embeddings visually change as t increases?

  I don't see significant difference between embeddings of different t. Because eigenvectors won't change as t increases.

- *How do eigenvalues change and how does this help explain what you see in embeddings?*

  As t increases, since eigenvalues are between 0 and 1, they decrease.

- *What can you learn about intrinsic dimensionality of dataset from eigenvalues of M as t increases?*

  As t increases, the number of significant eigenvalues decreases.

- *Is there a specific value of t that you find most informative for Swiss roll dataset?*

  t = 50 is most informative since there is one significant eigenvalue (first eigenvalue).

5.3 Final Thoughts

Q 5.4

- *Describe 2 other datasets that you may generate to test the diffusion mappings*
- *What you expect to see from these tests*

Dataset 1: a dataset in high dimensionality with multiple clusters

Test result: 2D visualization using components of diffusion map can separate these clusters clearly without overlapping.

Dataset 2: a dataset in 3D with a sphere shape

Test result: 2D using components of diffusion map can spread the surface of this sphere on a flat surface without twisting its correct structure.

6 Experiment 2: iPSC Reprogramming Dataset

- Dataset: `ipsc_data_set.json` contains 2005 time points subsampled from 220,450 time points
- 34 Features:
  - 33 columns of 33 protein markers, e.g., 'pplk1', 'h3k9ac', all names are in `channel_names.json`
  - `timepoint` column: simply row number, e.g. 1st timepoint is row 1, 2nd timepoint is row 2, etc.
- Many pre-processing steps have already done.

6.1 Visualizing Data with PCA

1. Run PCA on processed iPSC dataset, obtain principal components (PC), projections, singular values

   Principal components are contained in a 2005 x 2005 matrix, some of its rows are:

```
[[ 3.66284018e-02 -4.60637956e-02  1.55770037e-02 ... -5.82534793e-02
   2.76635706e-03 -8.50447978e-03]
 [ 9.63523167e-03  4.44370753e-02 -7.02414636e-03 ... -3.06995398e-02
   3.76177965e-02  2.51073565e-02]
...
[-6.43914654e-03 -3.22120729e-03  3.72198705e-02 ...  2.64447810e-03
   9.80155759e-01 -3.60600869e-03]
 [-7.21205615e-04  1.52367877e-02  5.30278951e-02 ... -1.68270648e-03
  -3.64045847e-03  9.81695776e-01]]
```

Projections are contained in a 33 x 33 matrix, some of its rows are:

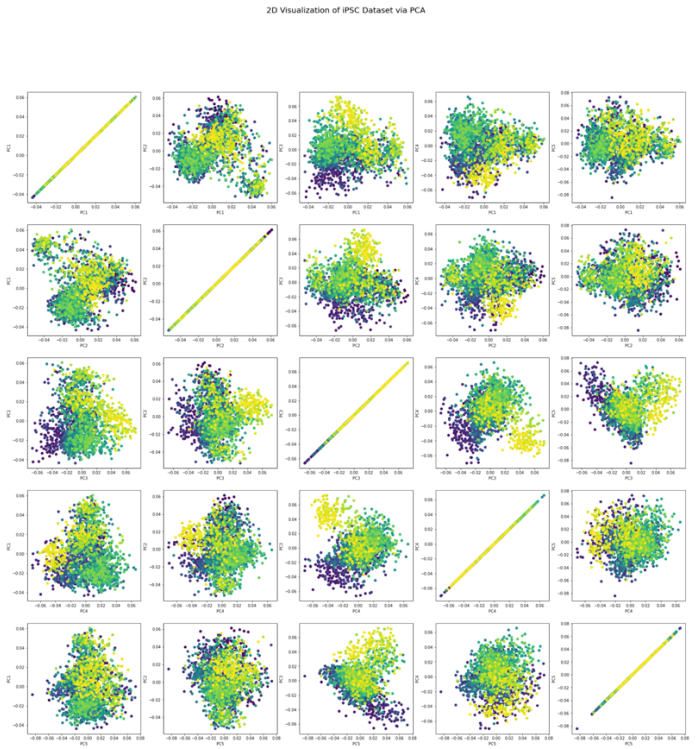
```
[[-3.47656757e+01  3.52995807e+01 -2.74929444e+00 ...  1.44209204e+0
1
   9.86727956e-01 -7.39080958e-01]
 [-5.78769943e+01  2.13784811e+01  1.81774436e+01 ... -1.03097549e+0
1
   2.11501482e-01 -6.04168513e-02]
...
[-5.97610543e+01 -5.30164487e+00 -4.20145592e+00 ... -6.83872388e-02
  -5.27872556e-01  3.56639486e-02]
 [-3.63129331e+01 -3.34960198e+00 -1.28038131e+01 ... -1.28226506e+0
0
   8.01265986e-01  5.16401529e-01]]
```

We have 33 singular values:

```
[187.95852218 162.50346932 111.87200507 101.87682359  85.92705467
  74.80877622  70.04470068  65.45496377  60.96943886  60.18312875
  58.78540851  54.58352307  51.82273753  50.60310829  49.15702757
  46.52660644  45.8710361   44.71614819  43.86064479  42.32729047
  41.41062519  38.63179425  37.83809446  36.3311266   34.01644835
  33.25348949  32.00294362  30.75039668  29.23333934  26.8116555
  24.49787972  24.26539302  17.71273959]
```

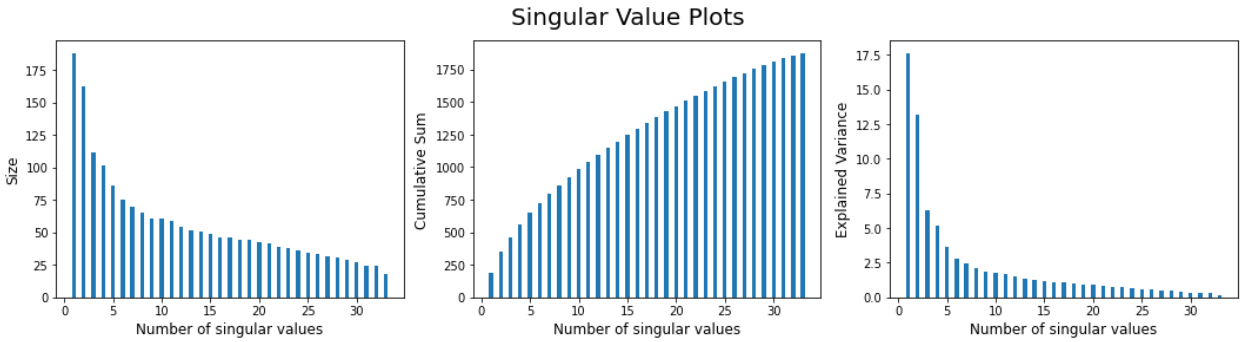
2. Plot iPSC dataset in 2D using PC, colored by time steps.

Here I try different combinations of top k=5 PC.



3. Plot singular values.

Here I plot size of singular values, cumulative sum of singular values, and explained variance.



Q 6.1

- *How do PCA visualizations look?*
- *Are they capturing time progression of dataset?*
- *Do you see any clusters forming?*

  All the plots form several clusters of 4 different colors: yellow, light green, dark green and purple. So, to some extent, PCA captures time progression of dataset. But I can't clearly distinguish between them because they have overlap.

- *What is intrinsic dimensionality given by PCA?*

  From the 1st plot (size) and 3rd plot (explained variance) of singular values, we can see there are 2 significant singular values, so the intrinsic dimensionality is 2.

- What are top few channels in first and second principal directions?

  In other words, these are top 5 proteins who first expressed during cell reprogramming.

  PC1: 'oct4', 'ki67', 'klf4', 'h3k9ac', 'prb'

  PC2: 'bcatenin', 'ikba', 'thy1', 'ps6', 'pstat3-727'

6.2 Visualization Data with Diffusion Maps

Note：I combine the experiment of k=2 and t=1 with experiments in **6.3 Changing Parameters** together. The results are all in 6.3 Section.

Q 6.2.

- *How do the Diffusion Map visualizations look?*

  2D plot looks like an arrow while 3D plot looks like a vertebral body.

- *Are they capturing the time progression of the data set?*

  Yes, They captured the time progression of dataset.

- *Do you see any clusters forming?*

  Yes, I can see clusters of different colors: purple, dark green and yellow.

- *What is the intrinsic dimensionality given by Diffusion Maps?*

  From the plot of explained variance and number of eigenvalues, I can see 4 significant dimensions.

- *Which channels are most highly correlated with diffusion components? Any guesses to the biological interpretation?*

  1st diffusion components: ['h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk']
  2nd diffusion components: ['bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4']
  3rd diffusion components: ['lin28', 'ssea1', 'epcam', 'cd44', 'gfp']

  Guess: These channels interact much with each other and play an important role during cell reprogramming.

- *What are the main differences between Diffusion Maps and PCA methods on the iPSC dataset?*

  PCA captures some sort of clusters but they have overlap and data points are spread out while diffusion maps not only captures clusters but also condense close data points to each other so the contour of clusters are clear and not overlapped which is good for interpretation of time progression.

6.3 Changing Parameters

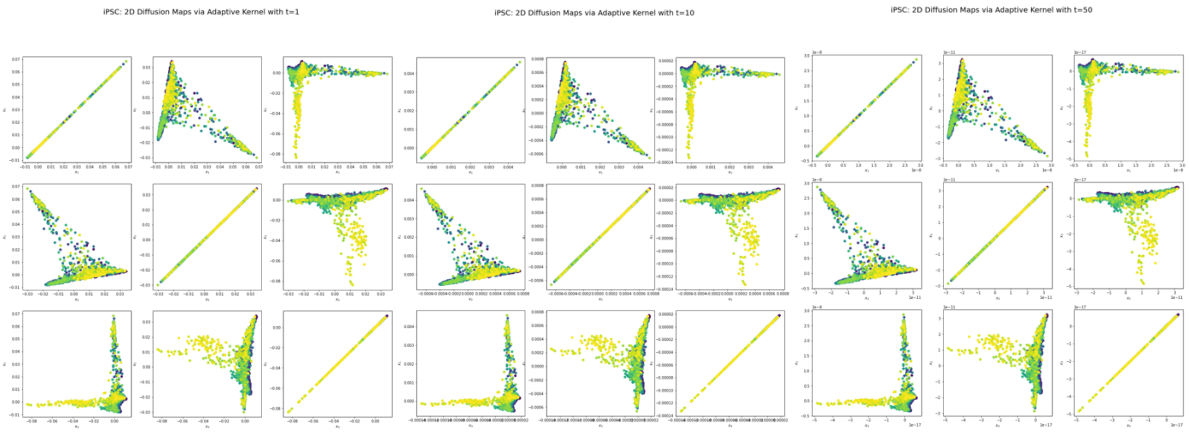
6.3.1 Changing Parameter $t = 1, 10, 50$

1. Construct affinity matrix of iPSC data set using Euclidean distance and adaptive KNN Gaussian kernel with k = 2 and diffusion parameter t.

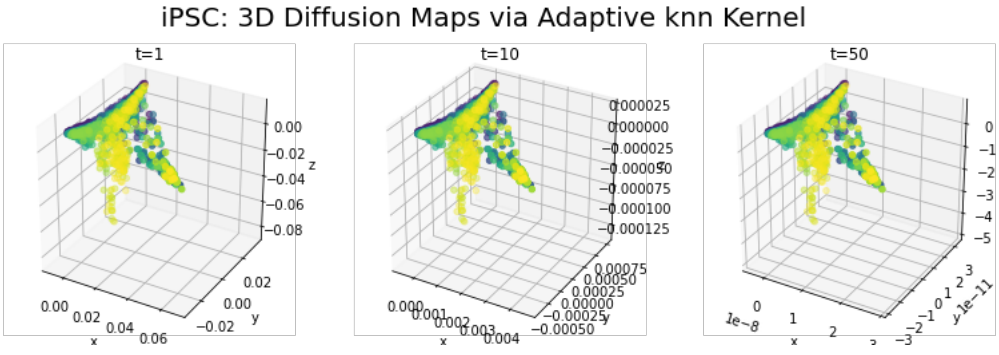
2. Create 2D scatter plots of diffusion mapping using different coordinates.

   Here I try three different combinations of 1st and 2nd coordinates, 1st and 3rd coordinates, 2nd and 3rd coordinates, respectively.

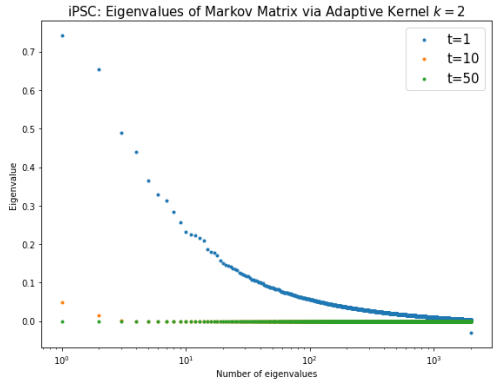
$t = 1$ **$t = 10$** **$t = 50$**



3. Create 3D scatter plot of diffusion mapping using 1st 3 coordinates for plotting. Color the points with their corresponding timepoint values.



4. Plot eigenvalues of Markov matrix.



5. Compute top 5 channels that have **highest absolute correlation** with 1st, 2nd, 3rd diffusion components.

t	1st Diffusion Components	2nd Diffusion Components	3rd Diffusion Components
1	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'ssea1', 'epcam', 'cd44', 'gfp'
10	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'epcam', 'ssea1', 'cd44', 'gfp'
50	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'epcam', 'ssea1', 'sox2', 'thy1'

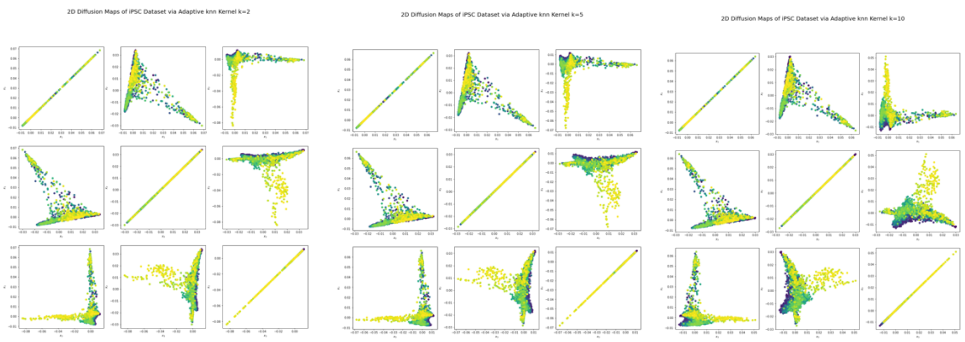
6.3.2 Changing Parameter k = 2, 5, 10

1. Construct affinity matrix of iPSC data set using Euclidean distance and adaptive KNN Gaussian kernel with k and diffusion parameter t=1.

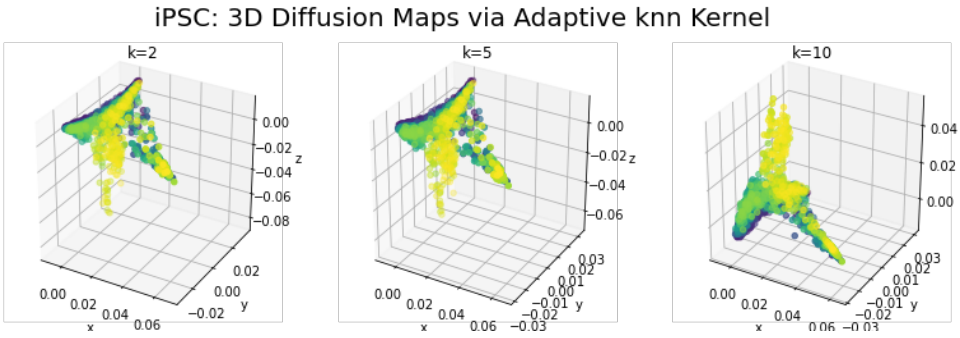
2. Create 2D scatter plots of diffusion mapping using different coordinates.

Here I try three different combinations of 1st and 2nd coordinates, 1st and 3rd coordinates, 2nd and 3rd coordinates, respectively.

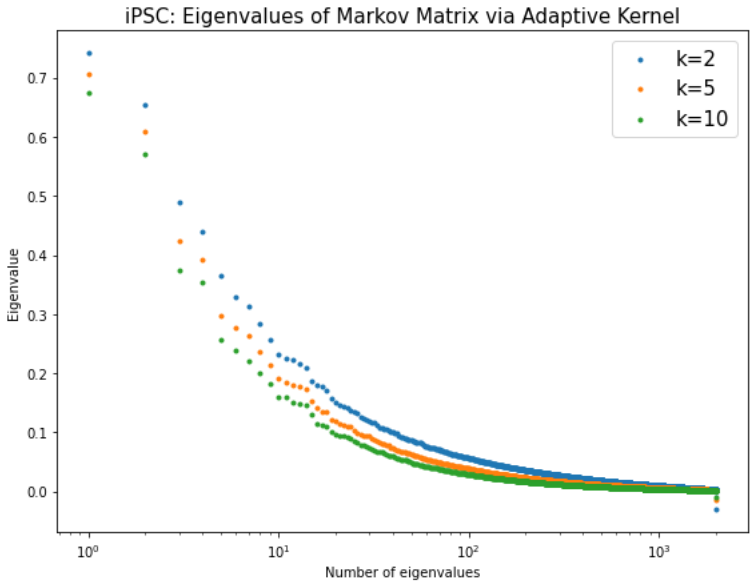
k = 2 **k = 10** **k = 50**



3. Create 3D scatter plot of diffusion mapping using 1st 3 coordinates for plotting. Color the points with their corresponding timepoint values.



4. Plot eigenvalues of Markov matrix.



5. Compute top 5 channels that have **highest absolute correlation** with 1st, 2nd, 3rd diffusion components.

k	1st Diffusion Components	2nd Diffusion Components	3rd Diffusion Components
2	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'ssea1', 'epcam', 'cd44', 'gfp'
5	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'epcam', 'ssea1', 'cd44', 'gfp'
10	'h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'	'bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'	'lin28', 'epcam', 'ssea1', 'sox2', 'thy1'

Q 6.3

- *Did choice of t affect embeddings and how?*

  My choice of t is 1, 10, 50. I don't see significant difference in 2D and 3D plots of embeddings.

- *Do corresponding eigenvalues support your finding?*

  In theory, as t increases, eigenvalues will decrease, which is supported by the plot of eigenvalues.

- *Did choice of k affect embeddings and how?*

  My choice of k is 2, 5, 10.
  When k=2 and k=5, the 2D plots or 3D plots of embeddings are very similar. But when k=10, both 2D plots and 3D plots of embeddings are different from former.
  For 2D plot, it differs in terms of x1 VS. x3 and x2 VS. x3.
  For 3D plot, it is turned upside down and occurs in bottom of plot.

- *Did previously observed trends (clusters, time progression) in diffusion dimensions change dramatically?*

  The previously observed trends aren't change dramatically.

- *Did correlated channels change with σ? How would you interpret this in terms of the data?*

  With different combinations of hyperparameters t and k, the correlated channels don't change too much with adaptive σ.

  1st diffusion components: ['h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'] (same)

  2nd diffusion components: ['bcatenin', 'cd140a', 'ikba', 'mefsk4', 'oct4'] (same)

  3rd diffusion components: ['lin28', 'ssea1', 'epcam', 'cd44', 'gfp'] or ['lin28', 'ssea1', 'epcam', 'sox2', 'thy1']

  This indicates these proteins ['h3k9ac', 'prb', 'pplk1', 'pstat3-727', 'pampk'] may interact much with each other and play an important role during cell reprogramming