1. #==============================================================================
2. #====== SLDM hw6, PROBLEM 1 ===================================================
3. #==============================================================================


7. # Need to pip install: "phate", "scprep" on the command line before running
8. # the rest of this code in an IDE.


12. #==============================================================================
13. #==== Loading the 10X data ====================================================
14. #==============================================================================
15. # PART A from the homework
17. # Import the data (code given on the tutorial website at:
18. # https://github.com/KrishnaswamyLab/PHATE/blob/master/Python
19. #        /tutorial/EmbryoidBody.ipynb)
20. import os
21. import zipfile
22. from urllib.request import urlopen
23. download\_path = os.path.expanduser("~")
24. print(download\_path)
26. **if** not os.path.isdir(os.path.join(download\_path, "scRNAseq", "T0\_1A")):
27. **if** not os.path.isdir(download\_path):
28. os.mkdir(download\_path)
29. zip\_data = os.path.join(download\_path, "scRNAseq.zip")
30. **if** not os.path.isfile(zip\_data):
31. with urlopen("https://data.mendeley.com/datasets/v6n743h5ng"
32. "/1/files/7489a88f-9ef6-4dff-a8f8-1381d046afe3"
33. "/scRNAseq.zip?dl=1") as url:
34. print("Downloading data file...")
35. # Open our local file for writing
36. with open(zip\_data, "wb") as handle:
37. handle.write(url.read())
38. print("Unzipping...")
39. with zipfile.ZipFile(zip\_data, 'r') as handle:
40. handle.extractall(download\_path)
41. print("Done.")
43. # Need these libraries
44. import pandas as pd
45. import numpy as np
46. import phate
47. import scprep
49. # Now use scprep to import data into a Pandas dataframe, using the
50. # scprep.io.load\_10x function.
51. sparse=True
52. T1 = scprep.io.load\_10X(os.path.join(download\_path,"scRNAseq", "T0\_1A"),
53. sparse=sparse,
54. gene\_labels='both')
55. T2 = scprep.io.load\_10X(os.path.join(download\_path, "scRNAseq", "T2\_3B"),
56. sparse=sparse,
57. gene\_labels='both')
58. T3 = scprep.io.load\_10X(os.path.join(download\_path, "scRNAseq", "T4\_5C"),
59. sparse=sparse,
60. gene\_labels='both')
61. T4 = scprep.io.load\_10X(os.path.join(download\_path, "scRNAseq", "T6\_7D"),
62. sparse=sparse,
63. gene\_labels='both')
64. T5 = scprep.io.load\_10X(os.path.join(download\_path, "scRNAseq", "T8\_9E"),
65. sparse=sparse,
66. gene\_labels='both')
67. T1.head()
69. # Now, merge all datasets and create a vector representing the time point of
70. # each sample
71. EBT\_counts, sample\_labels = scprep.utils.combine\_batches(
72. [T1, T2, T3, T4, T5],
73. ["Day 0-3", "Day 6-9", "Day 12-15", "Day 18-21", "Day 24-27"],
74. append\_to\_cell\_names=True
75. )
76. del T1, T2, T3, T4, T5 # removes objects from memory
77. EBT\_counts.head()

80. #==============================================================================
81. #===== Preprocessing: filtering, normalizing, and transforming ================
82. #==============================================================================
83. # This is PART B from the homework (technically most of this isn’t asked for in
84. # the homework, but I figured it made more sense to follow the tutorial exactly
86. # Remove (suspected) dead cells
87. mito\_genes = scprep.utils.get\_gene\_set(EBT\_counts, starts\_with="MT-")
88. # Get all mitochondrial genes. There are 14, FYI.
89. scprep.plot.plot\_gene\_set\_expression(EBT\_counts, mito\_genes, percentile=90)
90. # Plot number of cells that have a certain amount of mitochondrial RNA,
91. # remove cells that are above the 90th percentile. (Line below)
92. EBT\_counts, sample\_labels = scprep.filter.filter\_gene\_set\_expression(
93. EBT\_counts, mito\_genes,
94. percentile=90,
95. keep\_cells='below',
96. sample\_labels=sample\_labels)
98. # Now filter out cells that have either very large or very small library sizes.
99. # Library size is somewhat analogous to sample size. We'll eliminate the
100. # bottom 20% of cells for each sample.
101. scprep.plot.plot\_library\_size(EBT\_counts, percentile=20)
102. EBT\_counts, sample\_labels = scprep.filter.filter\_library\_size(
103. EBT\_counts, percentile=20,
104. keep\_cells='above',
105. sample\_labels=sample\_labels,
106. filter\_per\_sample=True)
108. # Now remove rare genes (genes expressed in 10 or fewer cells)
109. EBT\_counts = scprep.filter.remove\_rare\_genes(EBT\_counts, min\_cells=10)
111. # Normalization: accounting for differences in library sizes, divide each cell
112. # by its library size and then rescale by the median library size.
113. EBT\_counts = scprep.normalize.library\_size\_normalize(EBT\_counts)
115. # Transformation: use square root transform (similar to using log transform
116. # but has the added benefit of dealing with 0's automatically).
117. EBT\_counts = scprep.transform.sqrt(EBT\_counts)

120. #==============================================================================
121. #===== Applying PHATE to data =================================================
122. #==============================================================================
123. # (In the tutorial, this section is "Embedding Data Using PHATE")
125. # Default parameters for the PHATE function are:
126. # k: number of nearest neighbors, default is 5
127. # a: alpha decay, default is 40
128. # t: number of times to power the operator, default "auto", 21 for these data
129. # gamma: informational distance constant, default is 1.
130. phate.PHATE()
131. phate\_operator = phate.PHATE(n\_jobs=-2)
132. Y\_phate = phate\_operator.fit\_transform(EBT\_counts)
134. # Now plot using phate.plot.scatter2d
135. phate.plot.scatter2d(Y\_phate,
136. c=sample\_labels,
137. s=3,
138. figsize=(12,8),
139. cmap="Spectral")
141. # PART D (from homework): run PHATE on the data using a different value of t.
142. # Plot the PHATE coordinates colored by time point and include the plot.
143. # Based on the results, do you think your chosen value of t is better than the
144. # parameter chosen using the "knee point" of the VNE plot (the default value)?
145. # Will the VNE be higher or lower for your chosen value of t than that
146. # selected (by default) in part(b)?
147. phate\_op\_2 = phate.PHATE(n\_jobs = -2,
148. t = 30)
149. Y\_phate\_2 = phate\_op\_2.fit\_transform(EBT\_counts)
151. # Now plot using phate.plot.scatter2d having used t = 30
152. phate.plot.scatter2d(Y\_phate\_2,
153. c=sample\_labels,
154. s=3,
155. figsize=(12,8),
156. cmap="Spectral")
158. # PART E (from homework): run PHATE on the data using default parameters to
159. # obtain 3d coordinates. Plot the 3d coordinates. Rotate the plot such that
160. # it's different from what the tutorial has.
161. phate.plot.scatter3d(phate\_operator, c=sample\_labels, s=3, figsize=(8,6),
162. cmap="Spectral")
163. # This saves the 3D plot as a gif
164. phate.plot.rotate\_scatter3d(phate\_operator, c=sample\_labels,
165. s=3, figsize=(8,6), cmap="Spectral",
166. filename="phate.gif")
167. # This saves the 3D plot as an MP4 (which is also a cool gun in my opinion)
168. phate.plot.rotate\_scatter3d(phate\_operator, c=sample\_labels,
169. s=3, figsize=(8,6), cmap="Spectral",
170. filename="phate.mp4")