Ayaana Patel Sikora

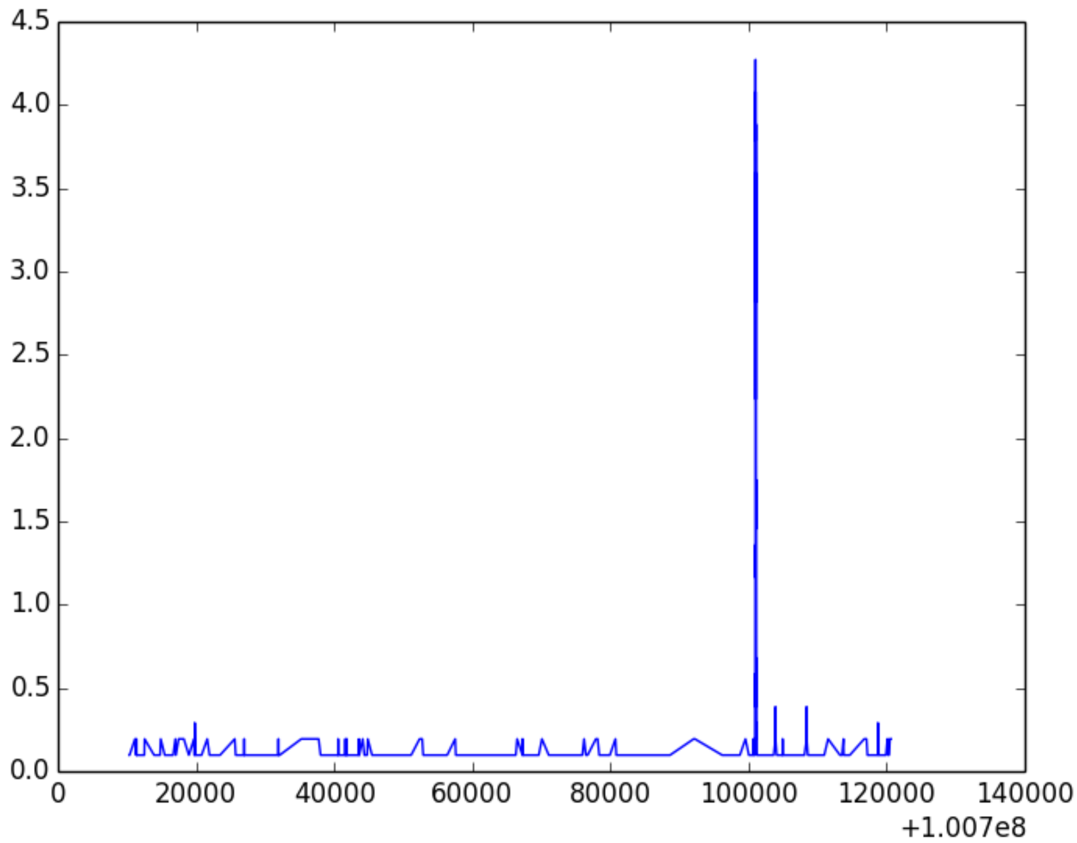
Bi 181 PS 5

Question 1:

1. The binding peaks were found by analyzing the data and looking at the highest density points:

   Peak 1: Start: 100801017, Stop: 100801018, Density: 4.2725
   Peak 2: Start: 100801021, Stop: 100801022, Density: 4.0783
   Peak 3: Start: 100801025, Stop: 100801026, Density: 4.0783

   A plot of the data is seen below:



2. The peak region can be defined as the region in which the densities are higher than 1:
   The average density in all the data = 0.193710409269
   The number of events being looked at (number of base pairs with a density higher than 1) = 142.

We can use the Poisson Formula:

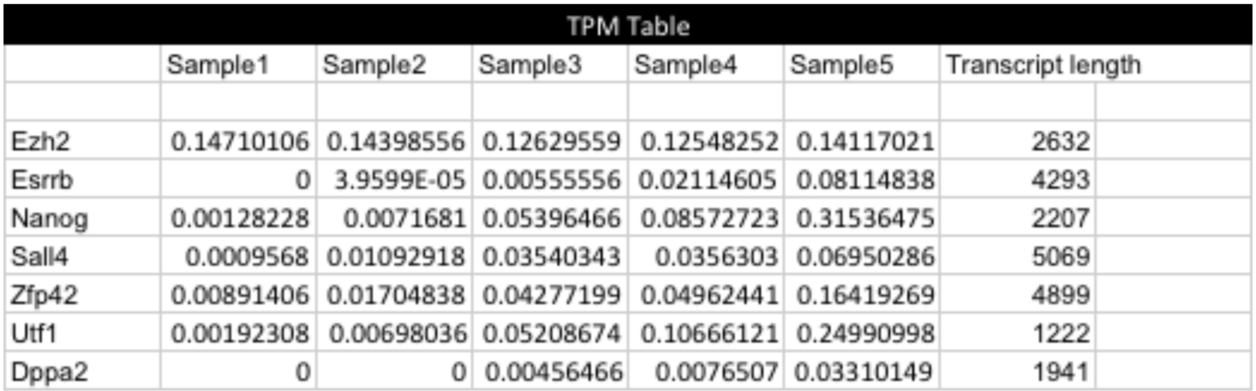
$P(k\ events\ in\ interval) = \frac{\lambda^{k} * e^{-\lambda}}{k!}$ where λ is the average density and k is the number of events being looked at.

Thus, P(142 events in interval) = $\frac{0.193710409269^{142} * e^{-0.193710409269}}{142!}$ =1.82379 * 10^{-347}

3. This P value is incredibly low, which tells us that the probability of the 142 events occurring to give this peak is very low.

Question 2:

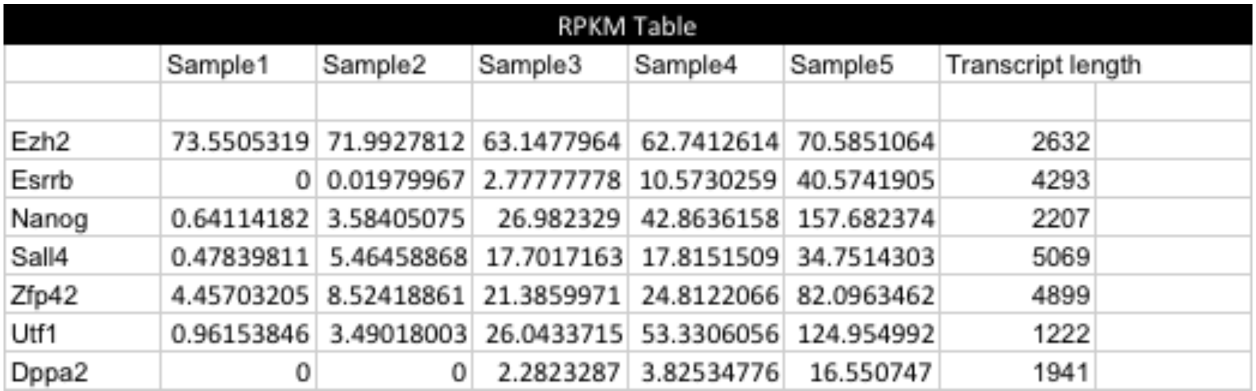
1. TPM was calculated using the formula: $10^{6} * \frac{\#\ of\ reads * read\ length}{T * Transcript\ Length}$, where T = 5 billion and the read Length = 50bp.



TPM Table

	Sample1	Sample2	Sample3	Sample4	Sample5	Transcript length
Ezh2	0.14710106	0.14398556	0.12629559	0.12548252	0.14117021	2632
Esrrb	0	3.9599E-05	0.00555556	0.02114605	0.08114838	4293
Nanog	0.00128228	0.0071681	0.05396466	0.08572723	0.31536475	2207
Sall4	0.0009568	0.01092918	0.03540343	0.0356303	0.06950286	5069
Zfp42	0.00891406	0.01704838	0.04277199	0.04962441	0.16419269	4899
Utf1	0.00192308	0.00698036	0.05208674	0.10666121	0.24990998	1222
Dppa2	0	0	0.00456466	0.0076507	0.03310149	1941

RPKM was calculated using the formula: $10^{9} * \frac{\#reads}{R*Transcript\ Length}$, where R = 200 Million



RPKM Table

	Sample1	Sample2	Sample3	Sample4	Sample5	Transcript length
Ezh2	73.5505319	71.9927812	63.1477964	62.7412614	70.5851064	2632
Esrrb	0	0.01979967	2.77777778	10.5730259	40.5741905	4293
Nanog	0.64114182	3.58405075	26.982329	42.8636158	157.682374	2207
Sall4	0.47839811	5.46458868	17.7017163	17.8151509	34.7514303	5069
Zfp42	4.45703205	8.52418861	21.3859971	24.8122066	82.0963462	4899
Utf1	0.96153846	3.49018003	26.0433715	53.3306056	124.954992	1222
Dppa2	0	0	2.2823287	3.82534776	16.550747	1941

2. TPM is the measure of the relative molar RNA concentration per sample. Unlike the RPKM, it takes into account the read length of the sample. In TPM, the value T represents the total number of transcripts in the genome. Meanwhile, in RPKM, the

value R represents the total number of reads across a sample. The relationship between R and the total number of transcripts sampled depends on the size distribution of RNA transcripts, which can differ between samples. Thus TPM is different from PRKM because it accounts directly for molarity of an RNA in the pool. The average TPM for each sample is guaranteed to be the same for any sample, unlike the average RPKM.

3. Nanog and Sall4 positively correlate with each other. Furthermore, Nanog has a higher level of gene expression in all the samples, except for Sample 2. This is because it has higher TPM and RPKM values in all the samples except Sample 2.

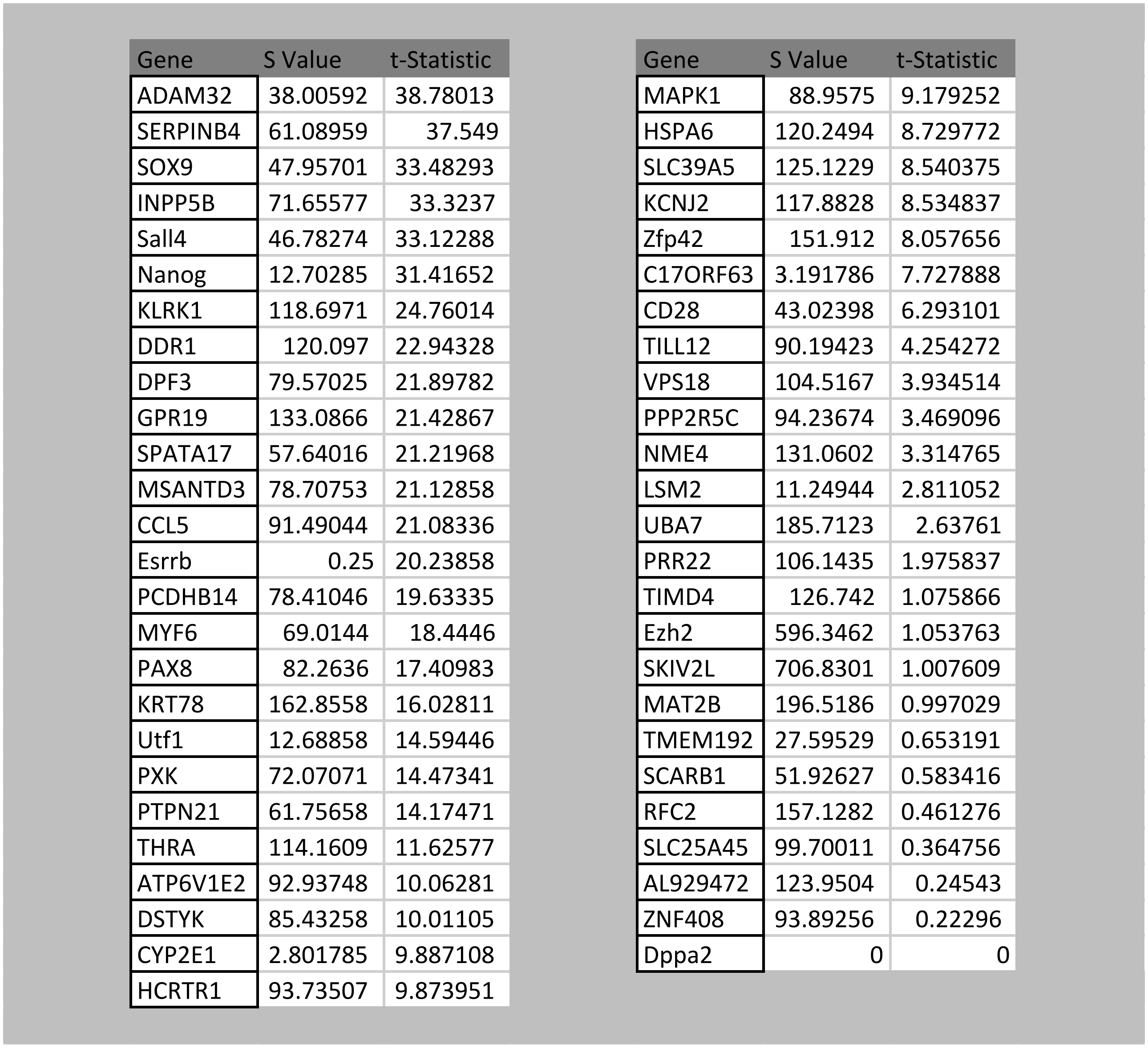
Question 3

1. Formula for the FPKM = Formula for the RPKM = $10^9 * \frac{\#fragments}{R*Transcript\ Length}$, where R = 10 million. For the Exon intersection, #fragments = 8 and the Transcript Length = 400. Thus, the FPKM value = $10^9 * \frac{8}{10000000*400}$ = 2
2. For the Exon Union Method: $10^9 * \frac{20}{R*1000}$, where R = 10 million . Thus, the FPKM value = 2.

Question 4

The S value and t-statistic were derived using the formula from the lecture slide:

$S^2 = \frac{Var(Group1)+Var(Group2)}{m+n-2}$ and $t = \frac{AVG(Group\ 1)-AVG(Group\ 2)}{S\sqrt{m+n}}$, where m and n are then lengths of each Group so m = 5 and n =5. The values are compile in the following table:



Gene	S Value	t-Statistic
ADAM32	38.00592	38.78013
SERPINB4	61.08959	37.549
SOX9	47.95701	33.48293
INPP5B	71.65577	33.3237
Sall4	46.78274	33.12288
Nanog	12.70285	31.41652
KLRK1	118.6971	24.76014
DDR1	120.097	22.94328
DPF3	79.57025	21.89782
GPR19	133.0866	21.42867
SPATA17	57.64016	21.21968
MSANTD3	78.70753	21.12858
CCL5	91.49044	21.08336
Esrrb	0.25	20.23858
PCDHB14	78.41046	19.63335
MYF6	69.0144	18.4446
PAX8	82.2636	17.40983
KRT78	162.8558	16.02811
Utf1	12.68858	14.59446
PXK	72.07071	14.47341
PTPN21	61.75658	14.17471
THRA	114.1609	11.62577
ATP6V1E2	92.93748	10.06281
DSTYK	85.43258	10.01105
CYP2E1	2.801785	9.887108
HCRTR1	93.73507	9.873951

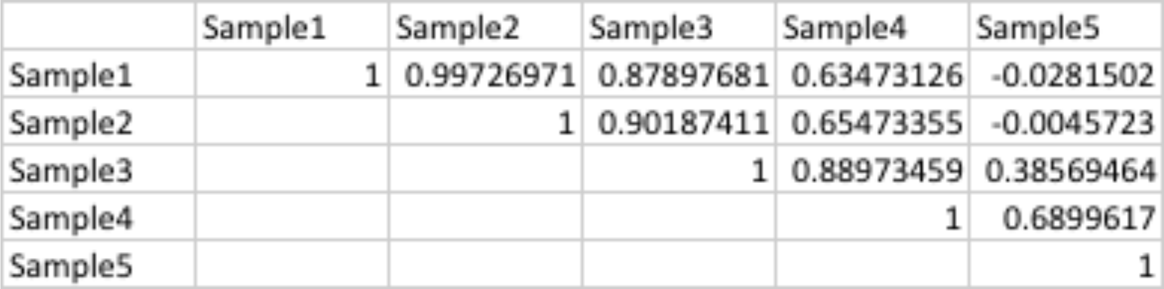
Gene	S Value	t-Statistic
MAPK1	88.9575	9.179252
HSPA6	120.2494	8.729772
SLC39A5	125.1229	8.540375
KCNJ2	117.8828	8.534837
Zfp42	151.912	8.057656
C17ORF63	3.191786	7.727888
CD28	43.02398	6.293101
TILL12	90.19423	4.254272
VPS18	104.5167	3.934514
PPP2R5C	94.23674	3.469096
NME4	131.0602	3.314765
LSM2	11.24944	2.811052
UBA7	185.7123	2.63761
PRR22	106.1435	1.975837
TIMD4	126.742	1.075866
Ezh2	596.3462	1.053763
SKIV2L	706.8301	1.007609
MAT2B	196.5186	0.997029
TMEM192	27.59529	0.653191
SCARB1	51.92627	0.583416
RFC2	157.1282	0.461276
SLC25A45	99.70011	0.364756
AL929472	123.9504	0.24543
ZNF408	93.89256	0.22296
Dppa2	0	0

Thus, the genes with the highest t-statistic value are:

ADAM32: 38.78013
SERPIN84: 37.549
SOX9: 33.48293
INPP5B: 33.3237
Sall4: 33.12288
Nanog: 31.41652
KLRK1: 24.76014
DDR1: 22.94328
DPF3: 21.89782
GPR19: 21.42867

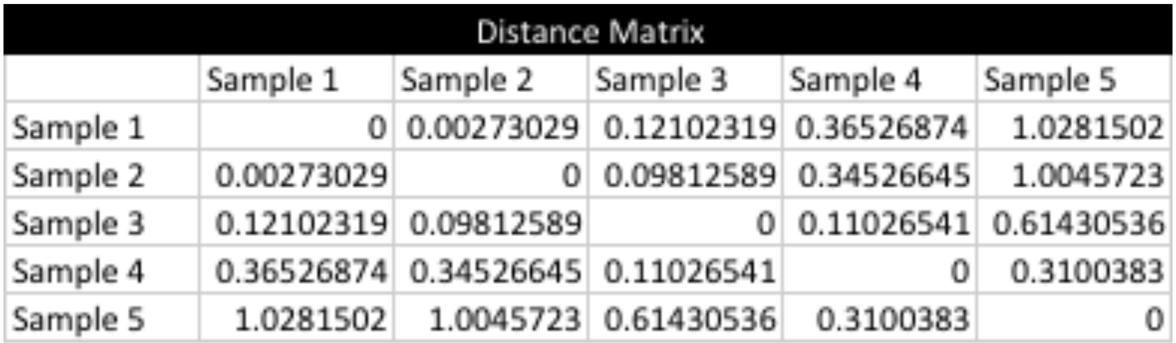
Question 5:

1. Correlation Matrix for the 5 Samples computed using Excel's Correl Function:



	Sample1	Sample2	Sample3	Sample4	Sample5
Sample1	1	0.99726971	0.87897681	0.63473126	-0.0281502
Sample2		1	0.90187411	0.65473355	-0.0045723
Sample3			1	0.88973459	0.38569464
Sample4				1	0.6899617
Sample5					1

2. The distance matrix version of this is:



Distance Matrix

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Sample 1	0	0.00273029	0.12102319	0.36526874	1.0281502
Sample 2	0.00273029	0	0.09812589	0.34526645	1.0045723
Sample 3	0.12102319	0.09812589	0	0.11026541	0.61430536
Sample 4	0.36526874	0.34526645	0.11026541	0	0.3100383
Sample 5	1.0281502	1.0045723	0.61430536	0.3100383	0

Let us denote Sample 1 to 5 as A through E.

D_1(A, B) is the smallest value of D_1, so we join elements A and B with a length of 0.00273/2 = 0.0013651.

D_2((A, B), C) = (D_1(A,C) + D_1(B, C))/2 = 0.1095745
D_2((A, B), D) = (D_1(A,D) + D_1(B, D))/2 = 0.3552676
D_2((A, B), E) = (D_1(A,E) + D_1(B, E))/2 = 1.0163613

	(A, B)	C	D	E
(A, B)	0	0.10957454	0.3552676	1.01636125
C	0.10957454	0	0.11026541	0.61430536
D	0.355267595	0.11026541	0	0.3100383
E	1.01636125	0.61430536	0.3100383	0

D_2 ((A, B), C) is the smallest value so we join (A, B) and C with a length of 0.10957454/2 = 0.0547873

$D_3(((A, B), C), D) = ((D_2(A,B), D * 2) + D_2(C, D))/3 = 0.2736002$
$D_3(((A, B), C), E) = ((D_2(A, B), E * 2) + D_2(C, E))/3 = 0.8823426$

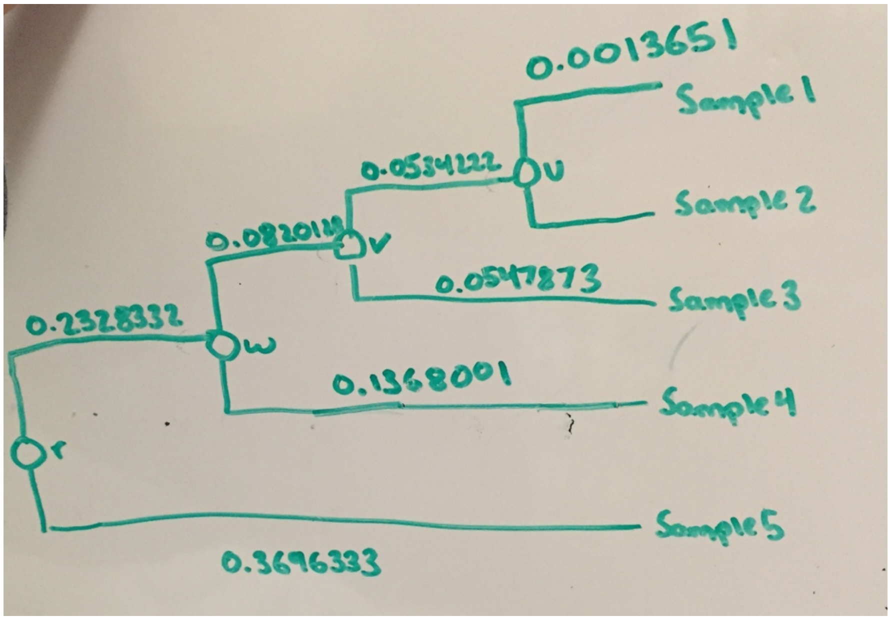
	((A, B), C)	D	E
((A, B), C)	0	0.2736002	0.88234262
D	0.2736002	0	0.3100383
E	0.88234262	0.3100383	0

$D_3(((A, B), C), D)$ is the smallest so we join ((A, B), C) and D with a length of 0.2736002/2 = 0.1368001

$D_4((((A, B), C), D), E) = ((D_{3(}((A,B),C), E) * 3) + D_3(D, E))/4 = 0.7392665$

	(((A, B), C), D)	E
(((A, B), C), D)	0	0.73926654
E	0.73926654	0

Thus, we join (((A, B), C), D) and E with a length of 0.73926654/2 = 0.3696333



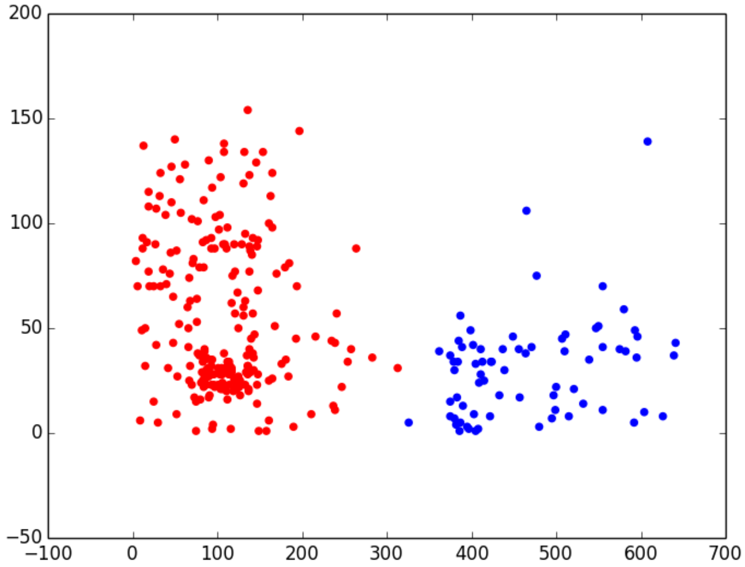
Question 6

1. In classification, the training samples are labeled unlike in clustering. Additionally, there is a rule to assign new samples to classes in classification. Meanwhile, in clustering, “close” points are clustered in groups. This allows us to identify the structure and

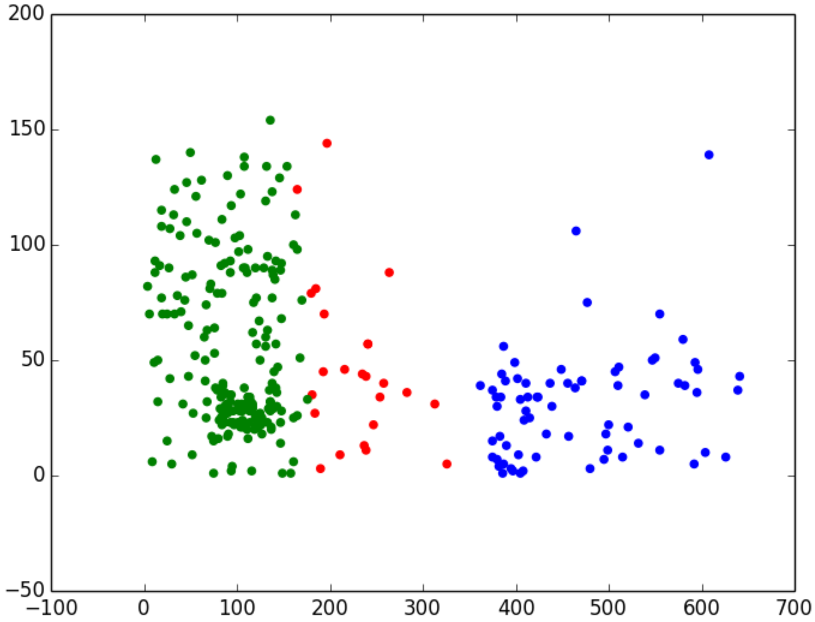
discover new classes. Classification is defined by supervised learning, while clustering is defined by unsupervised learning.

2. See code in Question6.py

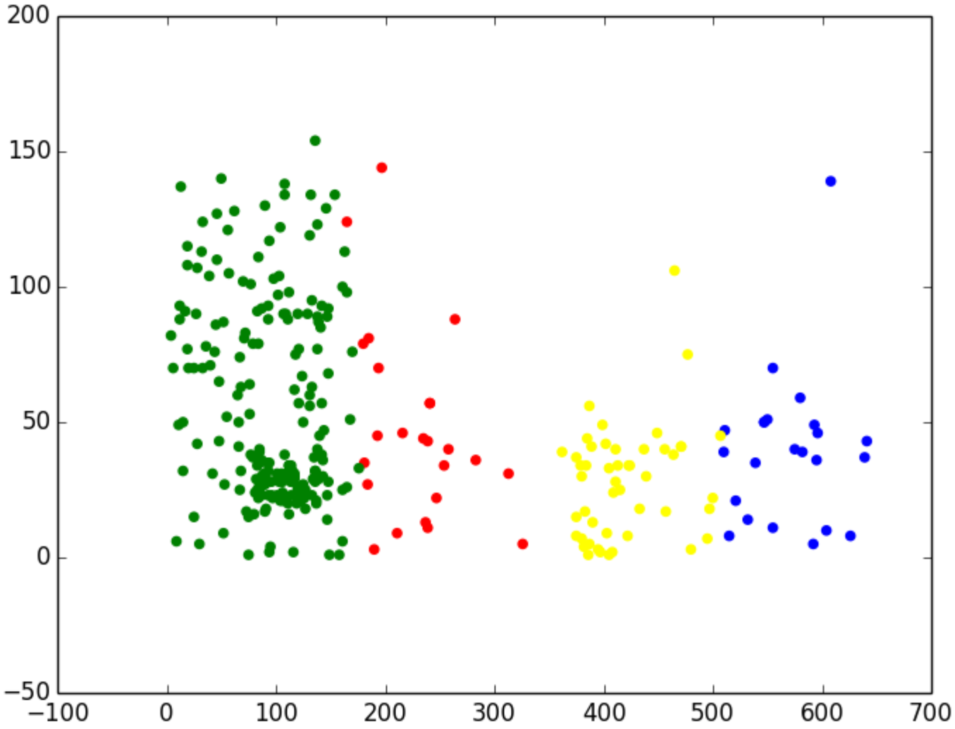
k = 2:



k = 3:



k = 4:



3. K = 2 or k = 3 are both reasonable k values. The k = 4 plot seems to be unnecessary (i.e. having 4 clusters is unnecessary, based on the fact that the data looks like it only needs to be split into 2 or 3). 2 seems the most reasonable because the two clusters seem very split.

Question 7

1. Current RNA-seq differential analysis methods focus on tackling one of two major challenges - first, accurately deriving gene and isoform expression values from raw sequencing reads and second, accounting for variability in measurements across biological replicates of an experiment. No algorithm has been able to rigorously address both problems simultaneously. Methods to control for variability have been mainly focused on controlling in raw read data, so they miss key aspects of transforming reads into gene expression values accurately. Furthermore, alternative splicing and repetitive regions introduce uncertainty into gene expression measurements. When methods fail to control for this, errors during differential analysis are introduced. Thus, current methods for differential analysis of RNA-seq are unable to control for both sources of variability and transcript level resolution, which means they are not able to accurately

capture transcriptome dynamics.

2. The Poisson model is discussed as one of the simplest models to control for variability. In this model, the variability is estimated by calculating the mean count across replicates, which allows one to calculate a P-value for any observed changes in a transcript's fragment count. However, this model fails to account for count uncertainty (which is the observation that in RNA-seq experiments, it is common for up to 50% of reads to map ambiguously to different transcripts) and count overdispersion (which is the fact that experiments that produce count data are often more variable across replicates than what is expected according to this model). The method used for Cuffdiff 2 addresses both of these issues by modeling how variability in measurements of a transcript's fragment count depends on both its expression and its splicing structure. This method estimates uncertainty by calculating the confidence that each fragment is correctly assigned to the transcript that generated it. Uncertainty is captured as a beta distribution and overdispersion is captured with a negative binomial. This model does not work for every case, which is a definite drawback.

3. Another probabilistic model that can be used is the Hidden Markov model. This is a stochastic model used to model randomly changing systems where it is assumed that future states depend only on the present state and not on the sequence of events that preceded it. In the hidden markov model, each state has its own probability distribution and the machine switches between states according to this probability distribution. This model is computationally efficient and quite accurate. However, the state sequence must be inferred (i.e. it is a con that the exact state sequence is not known). However, the HMM state sequence does overcome the boundary detection challenge in the normal Markov model.