Grouping Genes by Patterns of Gene Expression

Madison Hobbs

7/13/2017

Profile Assignment, PAM Clustering, and K-means Clustering on Filtered Data

We are deciding to remove the samples with third quartiles below 10 raw counts: 0.00_B which has a median of 1 and third quartile of 4 raw counts, 0.35_B which has a median of 1 and third quartile of 3 raw counts, and 20.40_A which has a median of 2 and third quartile of 8 raw counts. We also remove any rows whose maximum raw count is less than 5, which constitutes over 25% of the genes. In doing this, we reduce the number of genes we will consider from 14,108 to 10,376. Finally, we normalize the allCounts raw read data set with those columns and rows removed. We use DESeq2 (Love, Huber, and Anders 2014) to normalize.

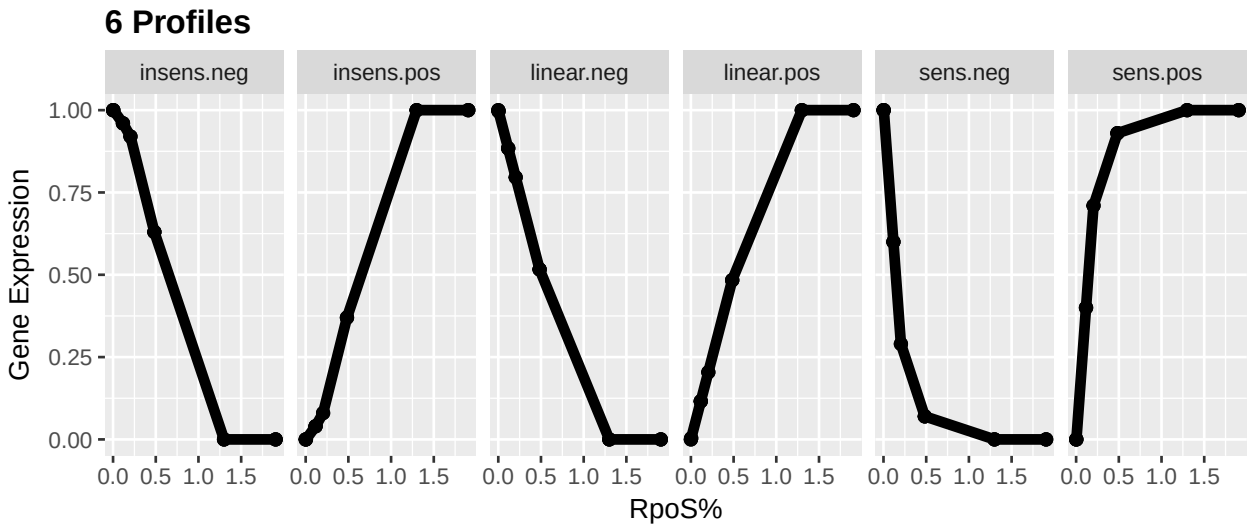
```
## # A tibble: 10 x 2
##     feature `n()/18`
##       <chr>    <dbl>
##  1   AS_CDS     3317
##  2   AS_IGR     1282
##  3 AS_ncRNA       38
##  4  AS_rRNA       17
##  5  AS_tRNA       38
##  6      CDS     4040
##  7      IGR     1472
##  8    ncRNA       62
##  9     rRNA       22
## 10     tRNA       88
```

Next, we locate the genes which are differentially expressed between the lowest and highest conditions using DESeq2 (Love, Huber, and Anders 2014). These will be what we consider to be the genes regulated by RpoS.

```
##
## out of 10374 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)     : 1061, 10%
## LFC < 0 (down)   : 795, 7.7%
## outliers [1]     : 1, 0.0096%
## low counts [2]   : 2816, 27%
## (mean count < 5)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

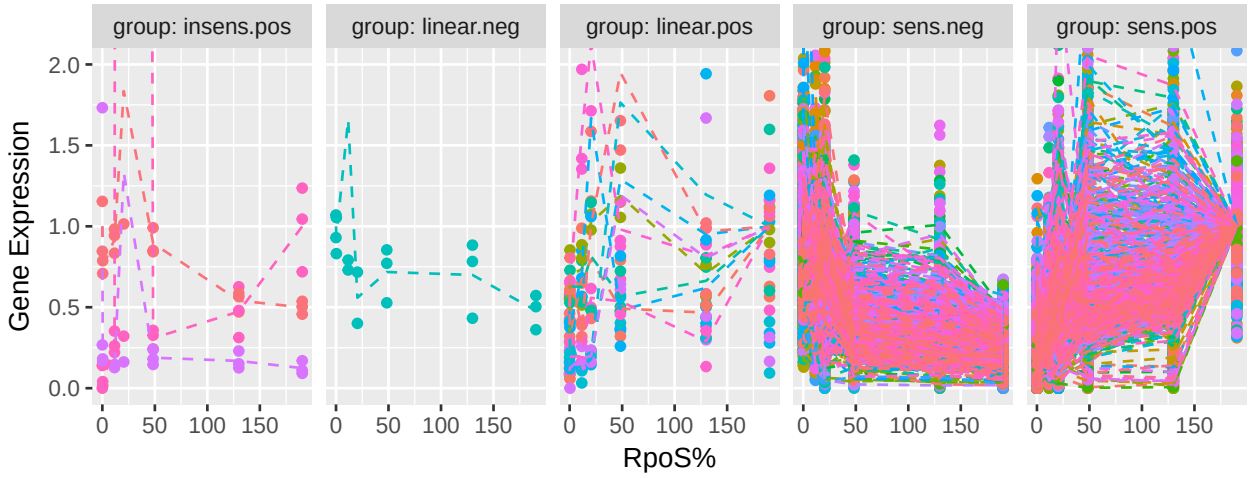
Profile Assignment - Extrapolated Profiles

We use the profiles extrapolated from profiles that worked to successfully classify the data with three levels (see Re-Rewritten2017_RNAseq_analysis using data from Wong et al. 2017).




```
## # A tibble: 5 x 2
##        group `n()/18`
##        <chr>    <dbl>
## 1 insens.pos        3
## 2 linear.neg        1
## 3 linear.pos       10
## 4   sens.neg      792
## 5   sens.pos     1050
```

```
## # A tibble: 6 x 3
## # Groups:   regulation [?]
##   regulation      group `n()/18`
##        <chr>      <chr>    <dbl>
## 1   negative insens.pos        2
## 2   negative linear.neg        1
## 3   negative   sens.neg      792
## 4   positive insens.pos        1
## 5   positive linear.pos       10
## 6   positive   sens.pos     1050
```



We notice that the majority of genes are most highly correlated with the sensitive positive or sensitive negative Shapes. It is also clear from the plots that the genes that do fall into insensitive positive, linear negative, and linear positive categories don't exactly have shapes which match the profile shapes. We see a lot of non-monotonicity throughout the plots, which is curious and merits further investigation along with other potential problems in the data [see Questions_Regarding_2017_Data_Simplified].

However, if we are to proceed with this data, it makes sense to try other grouping techniques, such as unsupervised clustering, which tell us what are the most common shapes present in our data.

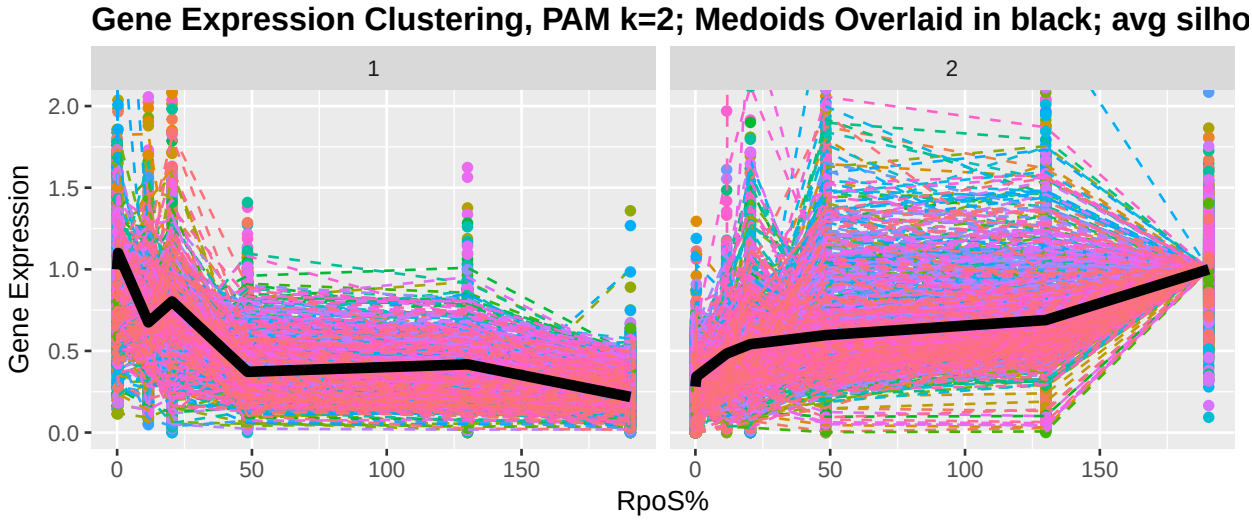
Partitioning Around Medoids (PAM) Clustering Analysis

Partitioning Around Medoids (PAM) (Kaufman and Rousseeuw 1990) is a robust clustering algorithm which serves to assign genes to some number k clusters by minimizing the average dissimilarity of genes to some number k of representative genes, called "medoids." We cluster on the 1,849 genes gathered above which are differentially expressed between 0% and 190.38% RpoS from the subset of genes do not have their maximum raw count below 5 and exluding the conditions 0.00_B, 0.35_B, and 20.40_A (which have third quartiles below 10). We select the optimal number of clusters by looking at a range of values for k.

```
## [1] "Adjusted Rand Index for k = 2 vs k = 3:  0.767"
## [1] "Adjusted Rand Index for k = 2 vs k = 4:  0.596"
## [1] "Adjusted Rand Index for k = 2 vs k = 5:  0.462"
## [1] "Adjusted Rand Index for k = 6 vs k = 7:  0.813"
## [1] "Adjusted Rand Index for k = 2 vs k = 5:  0.462"
## [1] "Adjusted Rand Index for k = 5 vs k = 5:  1"
```

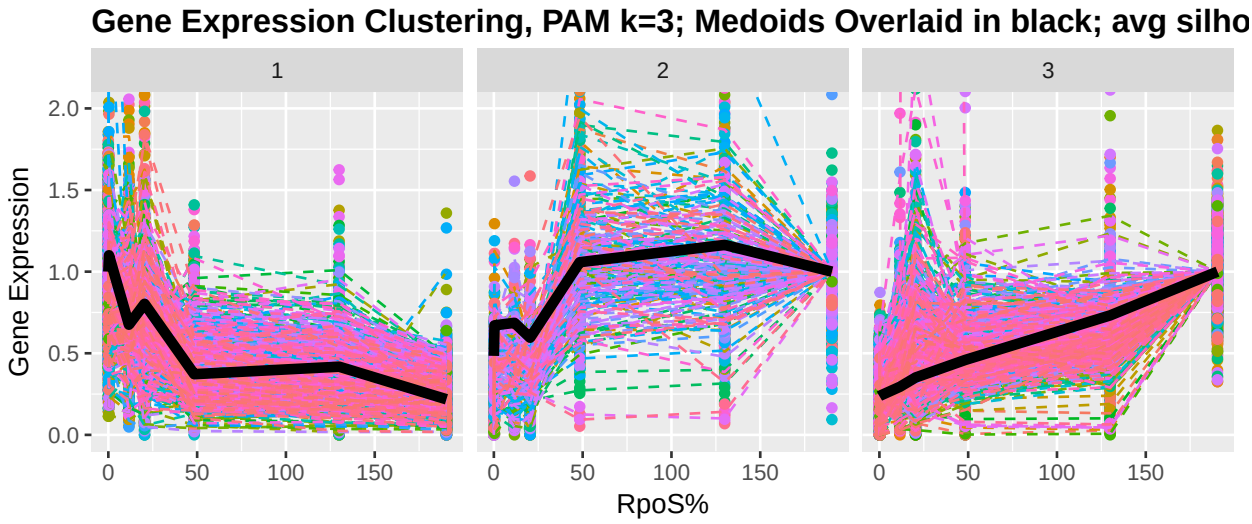
Adjusted Rand Index is a measure of how similarly two clustering methods cluster the data, with 0 meaning the clusters match up as well as they would by random chance and 1 representing perfect synchronozation between the clusters. Any ARI values below 0 would indicate that the clusters are more dissimilar than random chance would have it.

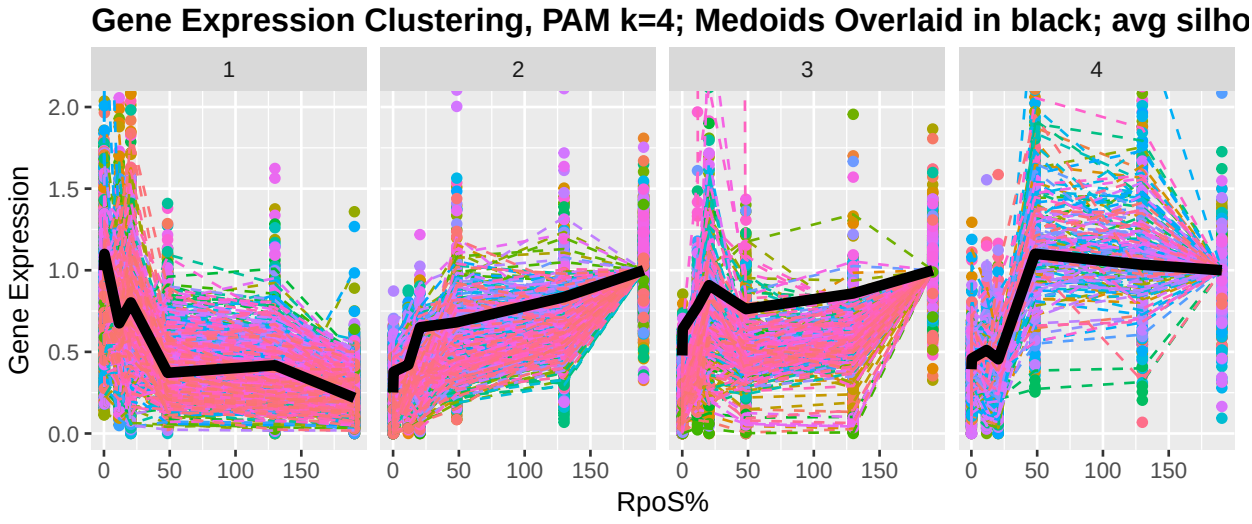
Predictably, the ARI is closer to 1 for values of k which are closer to each other, and ARI decreases as the two values of k we compare become more distant. For the same value of k, ARI is across two repetitions of k-medoids is always 1, indicating that the cluster medoids do not change for different iterations of PAM. This is probably because PAM employs a swapping process in the algorithm which checks to see if replacing each selected medoid with any other object would yield a better result.

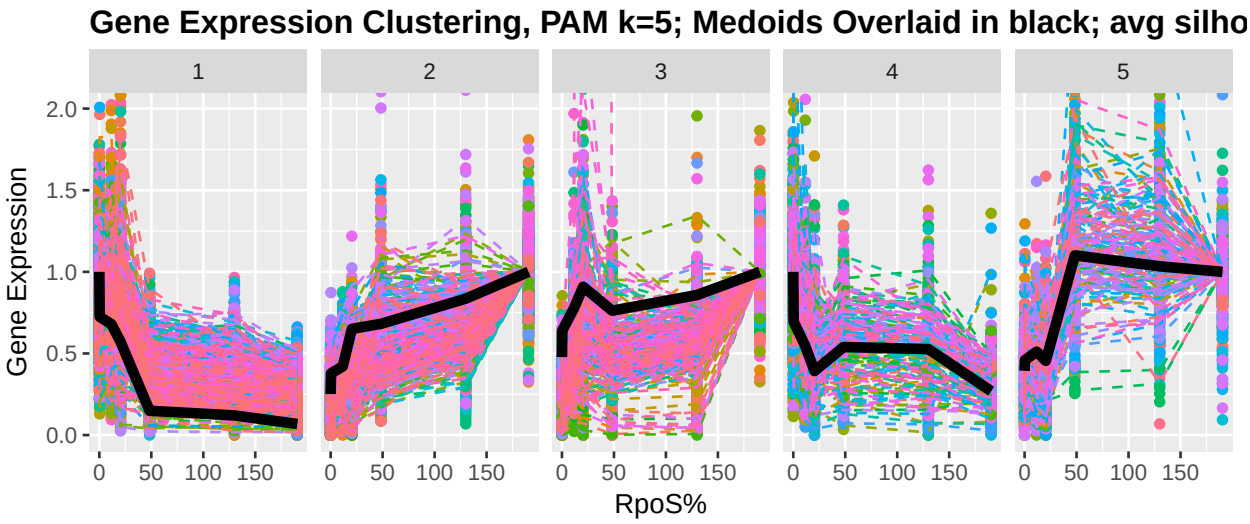


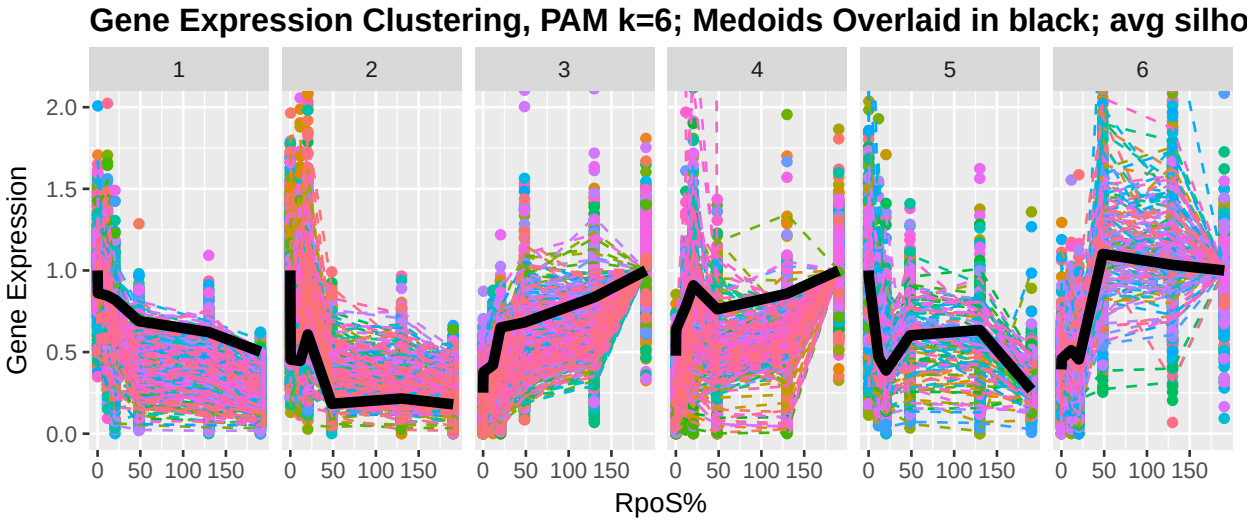
```
## # A tibble: 3 x 3
## # Groups:   regulation [?]
##   regulation pam.clustering.1 numGene
##        <chr>            <int>   <dbl>
## 1   negative                1     795
## 2   positive                1       2
## 3   positive                2    1059
```

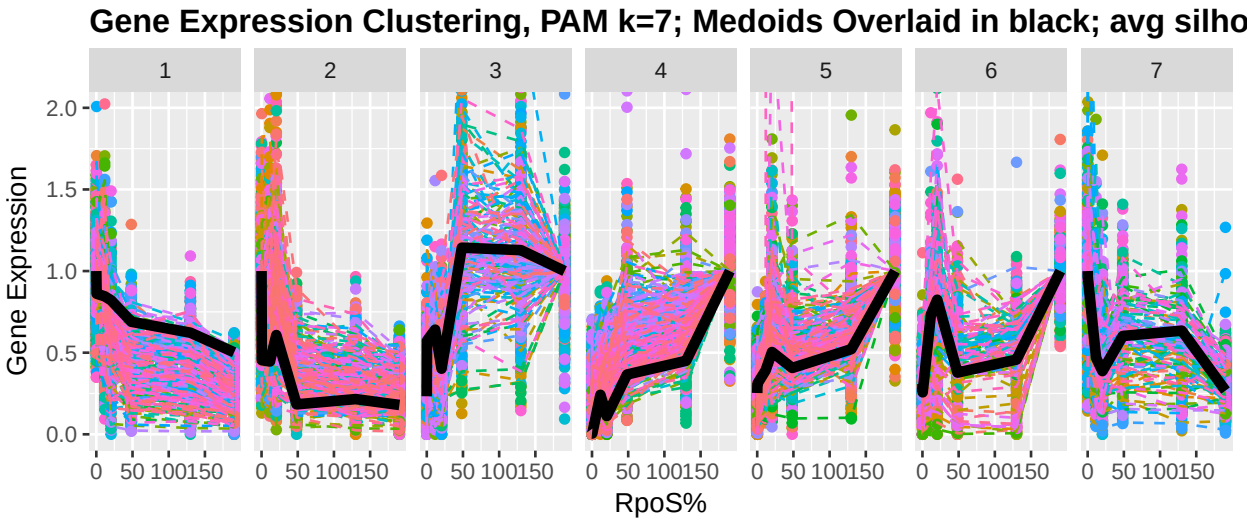
It is affirming that PAM independently clusters the genes nearly perfectly between the up- and down-regulation found by differential expression analysis.






Gene Expression Clustering, PAM k=5; Medoids Overlaid in black; avg silho
1
2
3
4
5
Gene Expression
2.0
1.5
1.0
0.5
0.0
0
50
100
150
RpoS%


Gene Expression Clustering, PAM k=6; Medoids Overlaid in black; avg silho
1
2
3
4
5
6
Gene Expression
2.0
1.5
1.0
0.5
0.0
0
50
100
150
RpoS%


Gene Expression Clustering, PAM k=7; Medoids Overlaid in black; avg silho
1
2
3
4
5
6
7
Gene Expression
2.0
1.5
1.0
0.5
0.0
0
50
100
150
RpoS%

The PAM plots help us see the most typical shapes of gene expression present in our data. Average silhouette width is a measure of how well each gene is classified into each cluster, and the closer this value is to 1, the more differentiable the clusters. Average silhouette width decreases as k increases for our data. This makes sense, as k = 2 (showing positively regulated and negatively regulated genes) has the highest silhouette width, telling us that the most prominent division in our data is into positively and negatively regulated genes. At seven clusters, average silhouette width is low (the closer to 0, the less differntiable the clusters). For larger values of k, the average silhouette width continues to decrease which shows us that larger values of k would yield progressively more overfitting and is probably not approrpriate.

With this Goldylocks-style thinking in mind, we look for values of k which yield differentiable clusters (relatively high average silhouette width) and at the same time produce medoids specific enough for us to characterize each cluster's general expression shape. Letting k be between 4 and 6 appears the best range for these purposes.

Looking at the clusters and gene shapes for the k=5 clustering results, we see a sensitive negative shape (cluster 1) and a sensitive positive shape (cluster 5). Non-montonoicity is common enough that two of the medoids have prominent non-monotonic shape, one whose general trend is positive (cluster 3) and the other whose general trend is negative (cluster 4). Finally, there exists a prominent roughly linear positive pattern (cluster 2).

```
## [1] "ARI profile assignment vs PAM k = 2:  0.98"
```

```
## [1] "ARI profile assignment vs PAM k = 6:  0.368"
```

Finally, we use the Adjusted Rand Index tocompare the PAM clustering results to those given by the assignment to the profiles we designed. Adjusted Rand Index (ARI) measures how similarly two clustering methods classified objects. An ARI of -1 would suggest that the two methods are in perfect disagreement, and an ARI of 1 would suggest that the methods are in perfect agreement. An ARI of 0 would suggest that the two methods are no better matched than they would be by random chance (based on the hypergeometric probability distribution).

The PAM cluster results with k = 2 is the closest match to the profile assignment results (ARI = 0.979). This makes sense as most of the genes were classified by profile assignment into one of two groups, sensitive positive and sensitive negative. The PAM cluster results with k = 6, the same number of groups used by profile assignment gives a much lower ARI (0.369), showing that that two methods do not classify genes in a very similar manner. We conclude that 6 groups generated by PAM are more representative of the shapes present in our data as compared to the six profiles we designed.

K-means

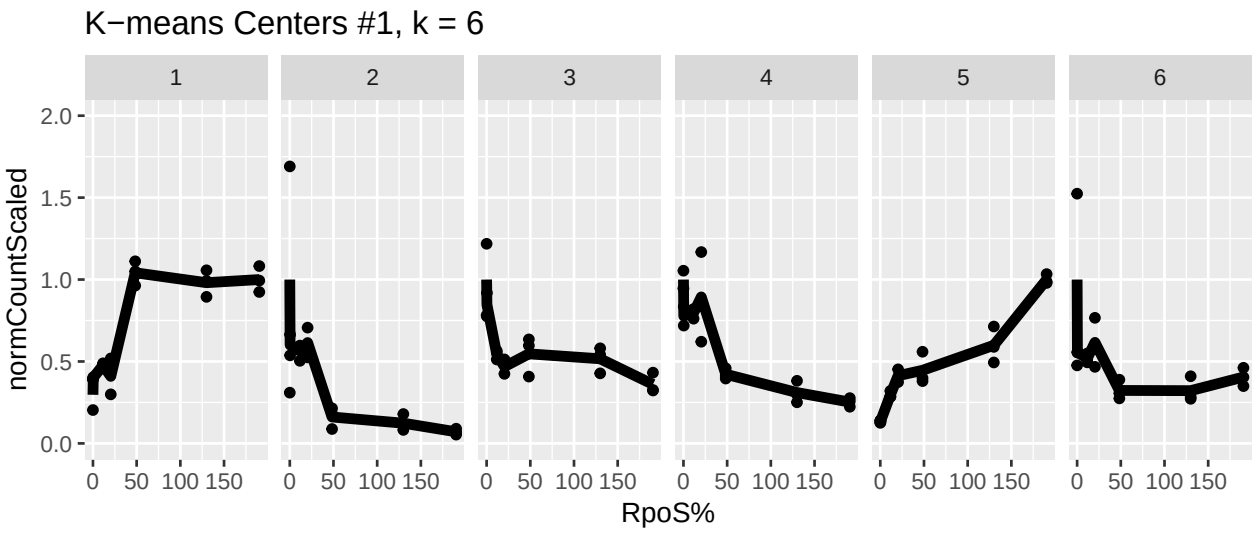
We also try the k-means clutering algorithm, as this is a common partitioning clustering algorithm and has been used in research addressing similar gene expression classification goals (**???**).

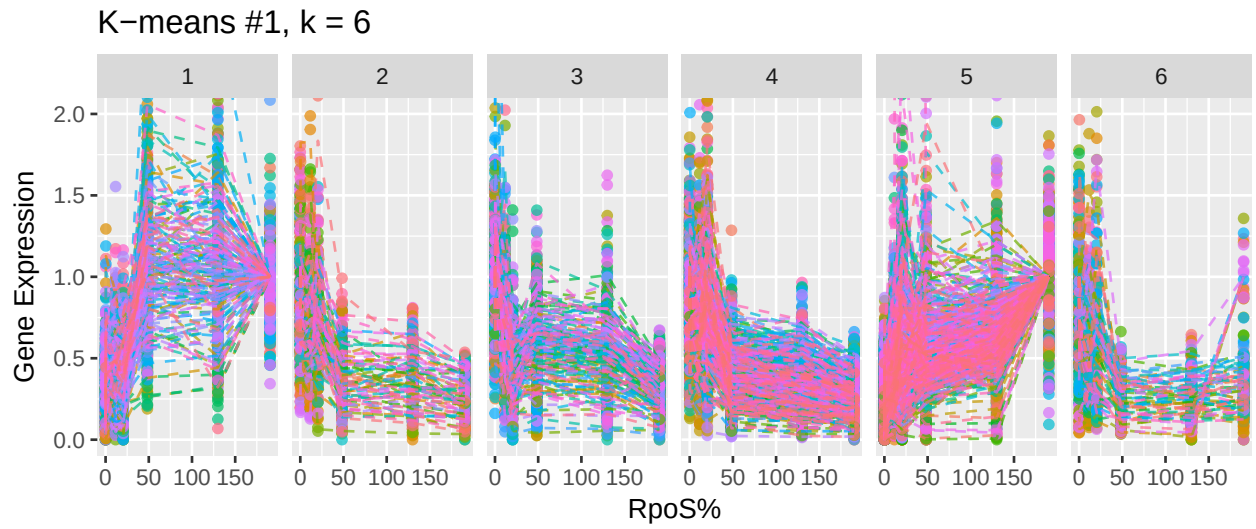
K-means randomly selects a k starting points, clusters the genes according to those k points, then finds the mean of each cluster (an amalgam of the genes in each cluster, so these centers end up not being real genes from our data). K-means then re-clusters according to these new clusters, takes the mean of each of the clusters it found, and re-clusters according to those new clusters. This process continues recursively until the cluster centers and clusters they form do not change (converge).

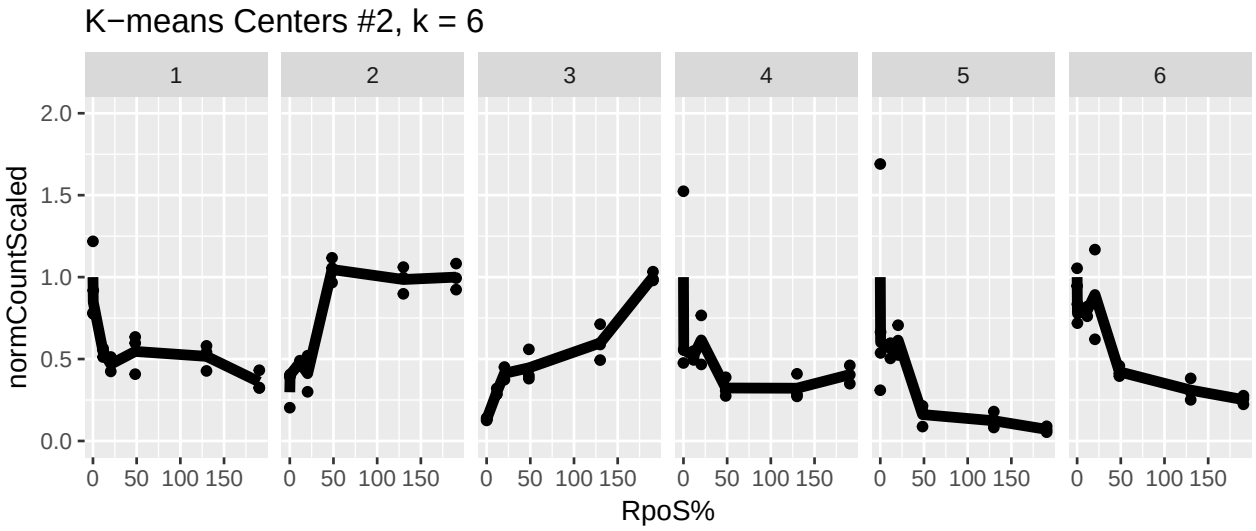
```
## # A tibble: 7 x 3
## # Groups:   regulation [?]
##   regulation kmeans.cluster `n()/18`
##        <chr>          <int>    <dbl>
## 1   negative              2       87
## 2   negative              3      206
## 3   negative              4      429
## 4   negative              6       73
```

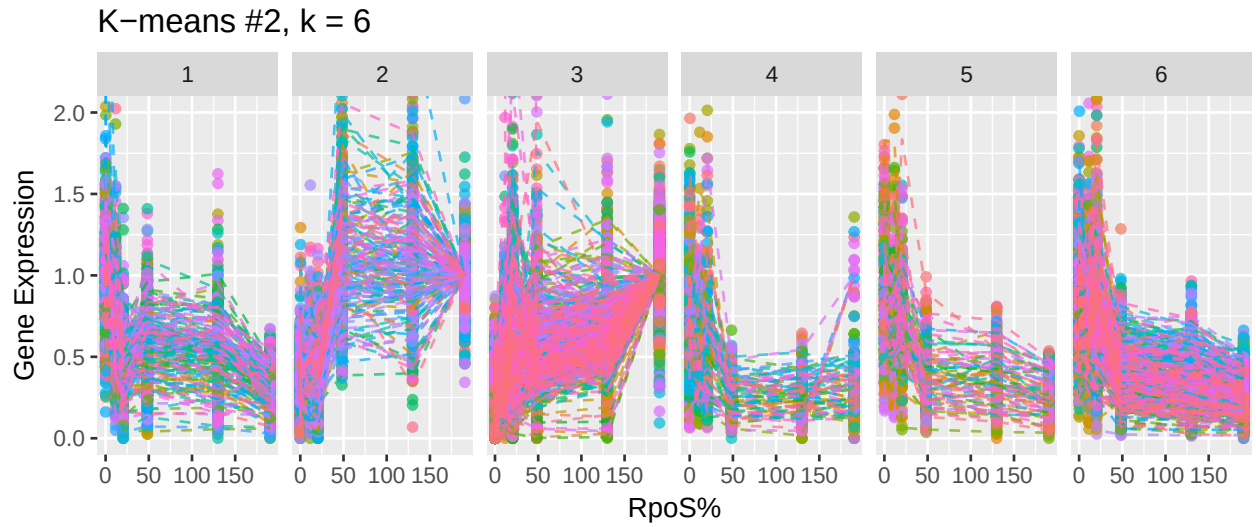
```
## 5   positive              1      213
## 6   positive              5      841
## 7   positive              6        7
```

```
## # A tibble: 7 x 3
## # Groups:   regulation [?]
##   regulation kmeans.cluster.2 `n()/18`
##        <chr>            <int>    <dbl>
## 1   negative                1      206
## 2   negative                4       77
## 3   negative                5       85
## 4   negative                6      427
## 5   positive                2      203
## 6   positive                3      852
## 7   positive                4        6
```










```
## [1] "ARI k-means #1 vs k-means #2; k = 6:  0.982"
```

```
## [1] "ARI k-means #1 vs PAM; k = 6:  0.471"
```

```
## [1] "ARI k-means #2 vs PAM; k = 6:  0.474"
```

What we notice from the two above plots is that even with the same input for k (6), k-means yields different clustering results. This is possible within the k-means algorithm because despite the convergence, the random selection of starting centers at the beginning allows for different possibilities for eventual convergence.

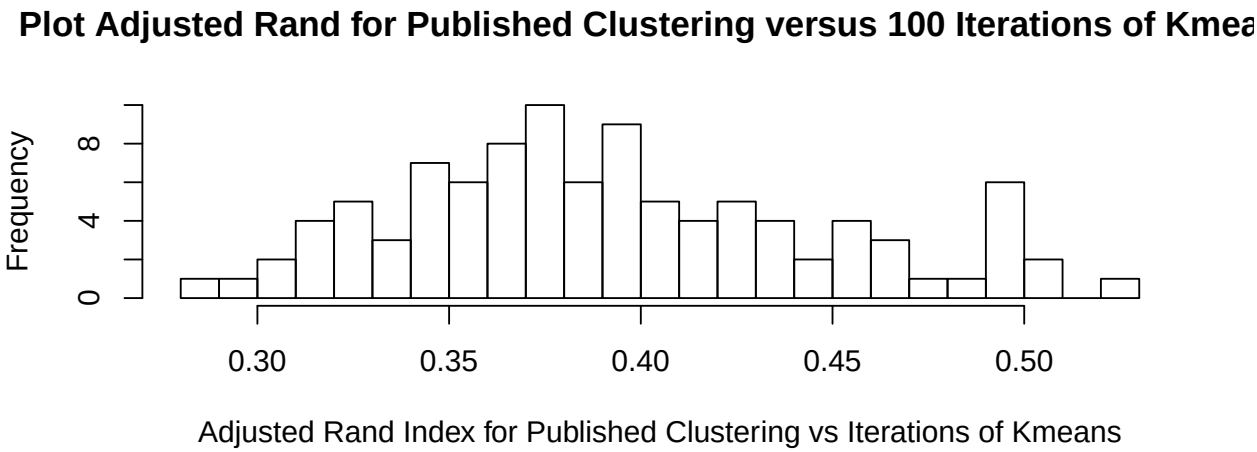
The ARI between two iterations of k-means with the same value of k is low (0.515), confirming what we see in the plots that the two methods do not cluster genes in the same way. Especially as compared to PAM, which gives the same clustering results each time for a constant input of k, using k-means to cluster our genes gives us pause. Comparing the two k-means iterations to the results given by PAM, we also notice that both k-means results are both rather dissimilar to the resutls given by PAM. In conclusion, we prefer the more robust Partitioning Around Medoids approach.

Clustering Analysis B. subtilis CodY Data Set

The published clusters on the B. subtilis data set investigating the CodY regulon (Brinsmade et al. (2014)) we clustered according to k-means with 14 clusters and using Pearson correlation as a dissimilarity metric. As we observed different results each time through the k-means algorithm above, we wanted to investigate how similar the clustering would be if we reproduced the k-means clustering. Note that we use k-means from the amap package in R, and Brindsmade et. al used MATLAB, but that is the only difference. We use the same regulon and RPKMO normalized counts provided in the paper.

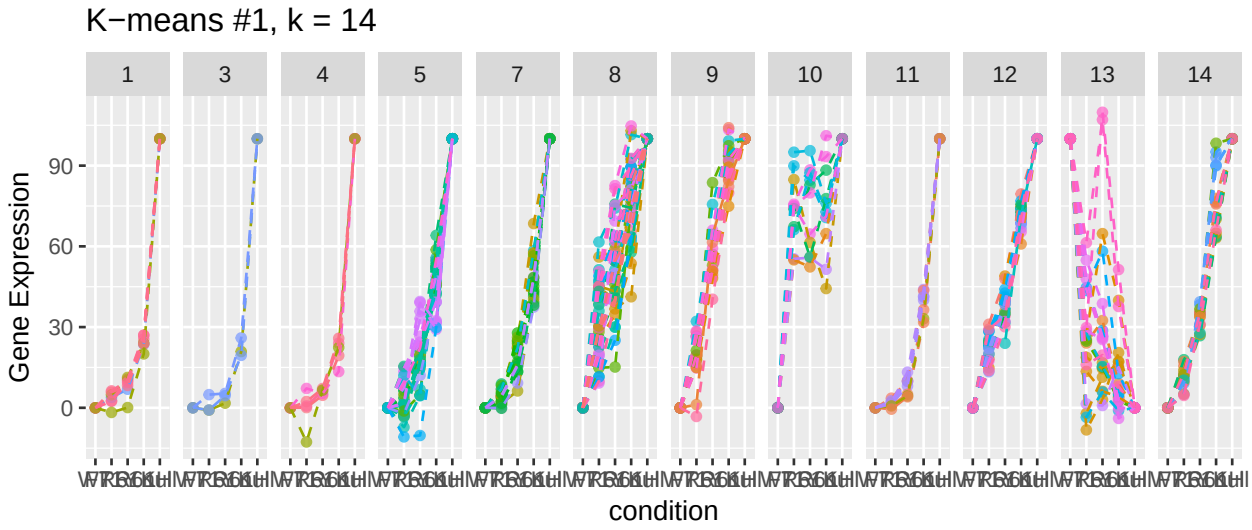
Compare to the published findings

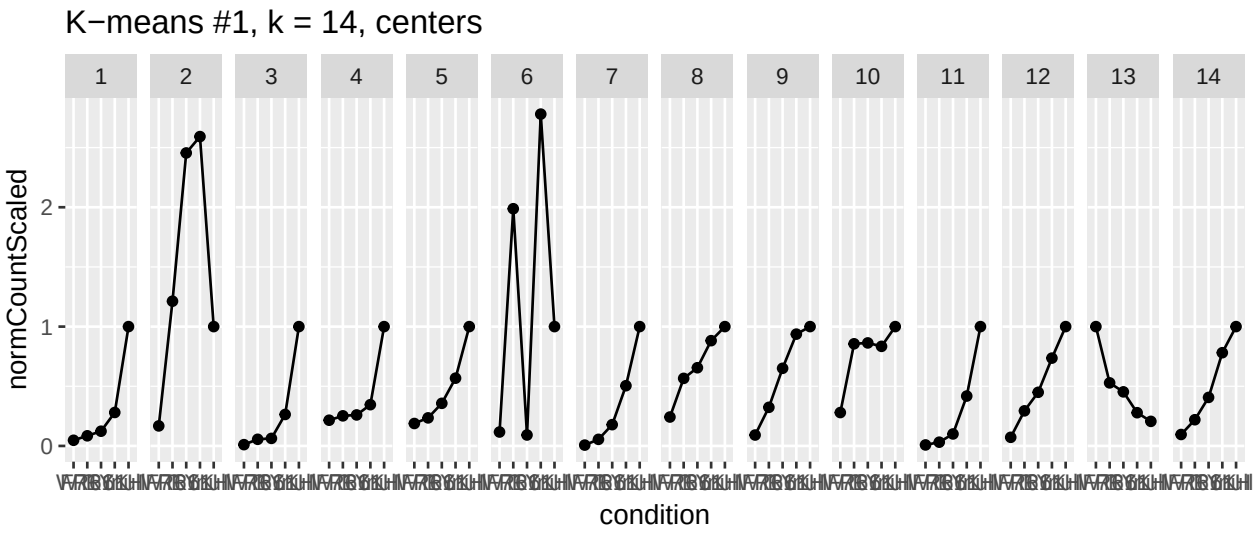
Kmeans Clustering Comparisons



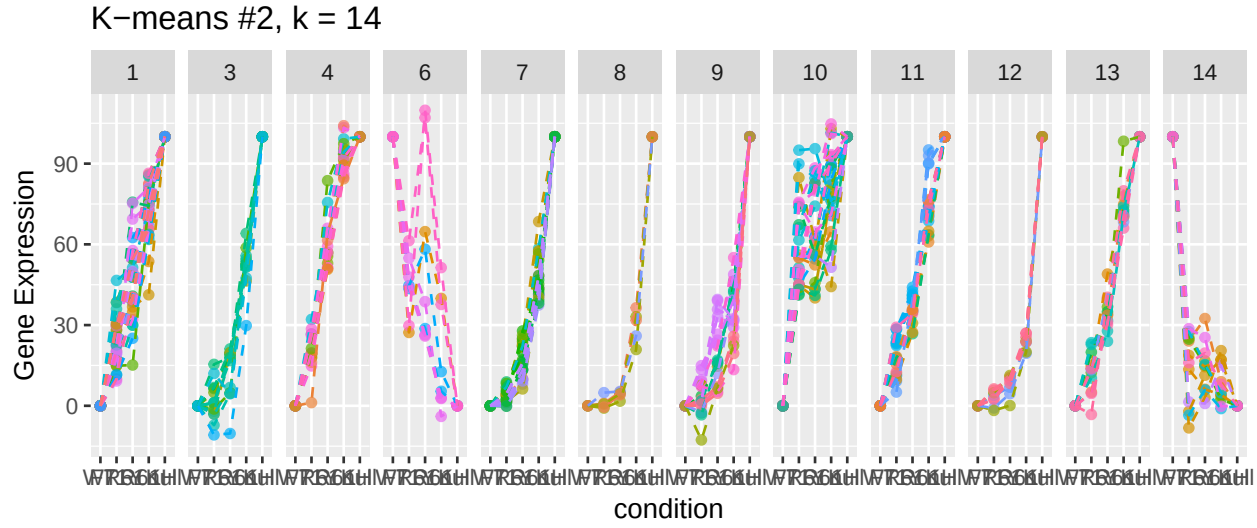
We notice the changes in clustering and centers in each run of the kmeans algorithm. After calculating the Adjusted Rand Index for the published clustering results (Brinsmade et al. (2014)) versus the clustering given by each iteration of Kmeans we perform above, we see the distribution of Adjusted Rand Indices varies between around 0.3 and 0.5. This tells us that the published results agree with our iterations of Kmeans better than completely random assignment (ARI = 0), but not much better than that.

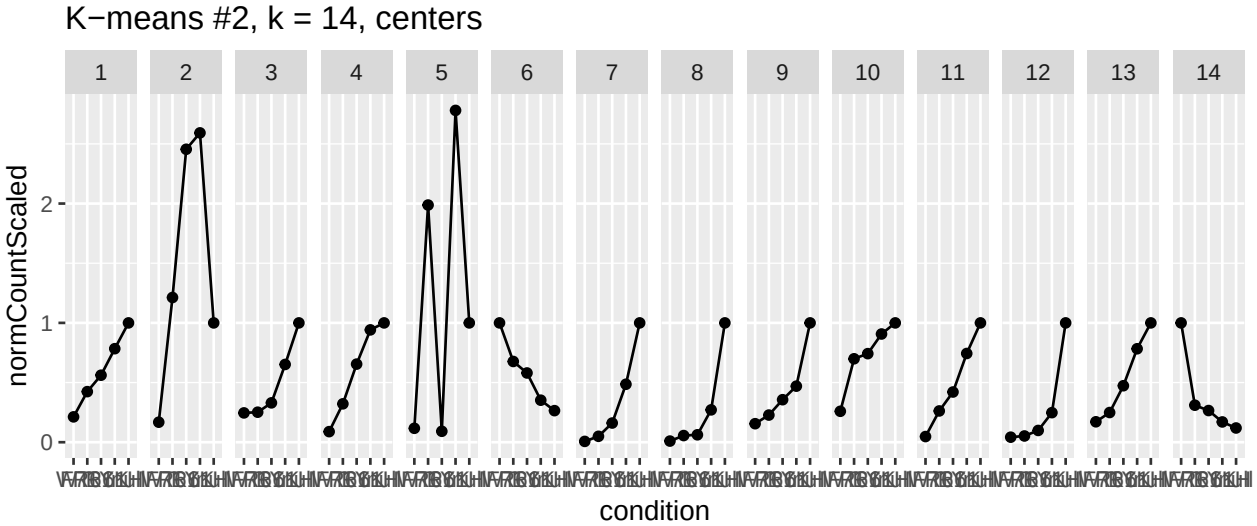
```
## [1] 33
```




K−means #1, k = 14, centers
1
2
3
4
5
6
7
8
9
10
11
12
13
14
normCountScaled
0
1
2
condition

```
## [1] 4
```


K−means #2, k = 14
1
3
4
6
7
8
9
10
11
12
13
14
Gene Expression
0
30
60
90
condition

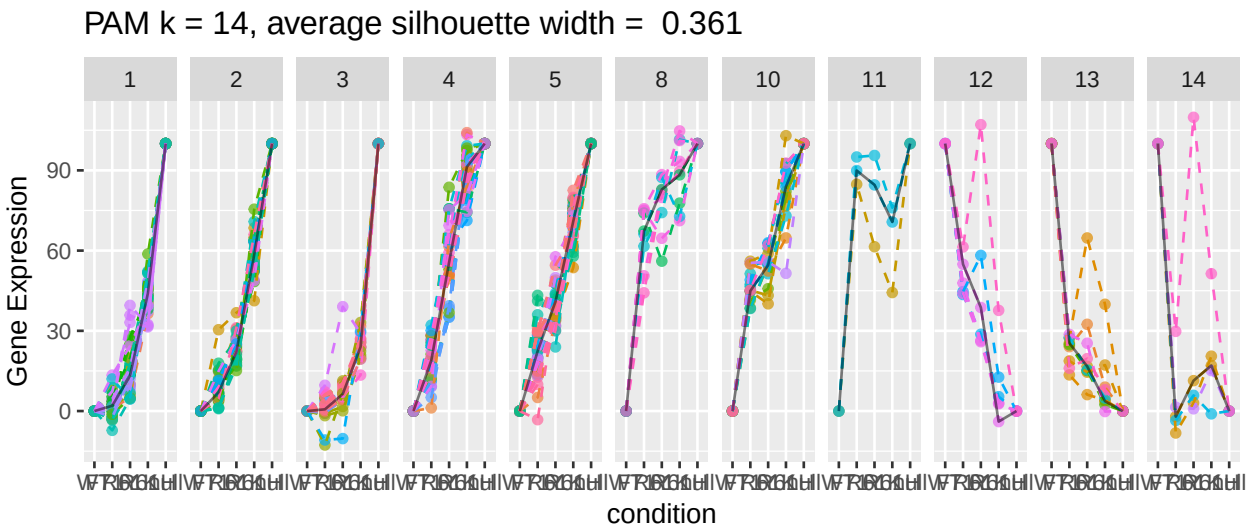

K−means #2, k = 14, centers
1
2
3
4
5
6
7
8
9
10
11
12
13
14
normCountScaled
0
1
2
condition

```
## [1] "Adjusted Rand Index for K-means33 vs K-means4:  0.566"
```

The two randomly-selected kmeans iterations plotted above both happen to demonstrate empty clusters, which is a phenomenon possible in the kmeans algorithm. This phenomenon may suggest the number for k is wrong, and that the data is overfit. One can see an example of the Kmeans algorithm producing an empty cluster here (http://user.ceng.metu.edu.tr/~tcan/ceng465_f1314/Schedule/KMeansEmpty.html). This problem and a solution in the form of a modified K-means algorithm has been introduced (Pakhira (2009)).

PAM on B. subtilis Data

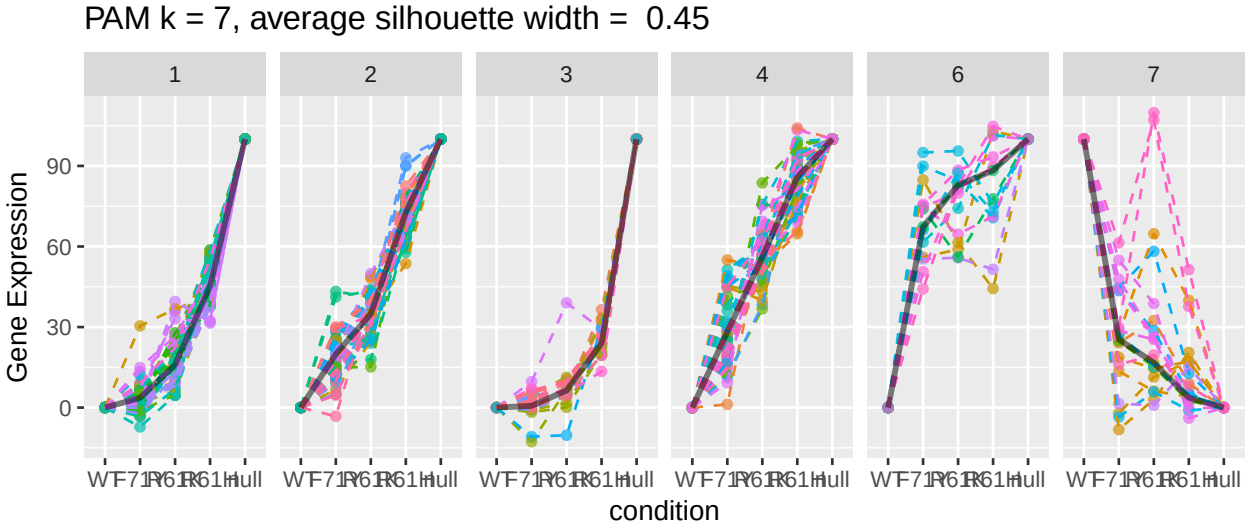
Kmeans does not have an equivalent measure of silhouette width as in PAM. We are curious about the silhouette width on the B. subtilis data with 14 clusters, and so we use PAM to cluster the genes.



The silhouette width is not the worst, but let's see if it could be any higher:

```
## [1] "Average Silhouettte Width, k = 2 is 0.878"
```

The silhouette width is highest at k = 2, probably denoting the up- and down-regulation as seen in our data. The silhouette width stabilizes at around 0.44 between 5 and 8 clusters. Plotting with 7 clusters, with the highest silhouette width for k = 5-8, we obtain:



We see in the clustering with k = 7, the same shapes are represented as in the k = 14 clustering. However, to make distinction between clusters with and without monotonicity, more clusters than 7 appear needed.

Future Work

1) Assess the problems and concerns we have with this data set (see Questions_Regarding_2017_Data and Taxonomy_Matchup), possibly resequencing or redoing the experiment/
2) We that the majority of raw read counts in each sample belonged to the region which contains ssrA, a tmRNA. It would be interesting to know if this high level of ssrA is expected under our experimental constants (such as starvation). See Taxonomy_Matchup #3 for more information.
3) Assess our clustering techniques with simulations of randomly generated RNASeq counts using R packages such as ballgown and polyester.

References

Brinsmade, Shaun R, Elizabeth L Alexander, Jonathan Livny, Arion I Stettner, Daniel Segrè, Kyu Y Rhee, and Abraham L Sonenshein. 2014. "Hierarchical Expression of Genes Controlled by the Bacillus Subtilis Global Regulatory Protein Cody." *Proceedings of the National Academy of Sciences* 111 (22). National Acad Sciences: 8227–32.

Kaufman, Leonard, and Peter Rousseeuw. 1990. *Finding Groups in Data*. John Wiley & Sons, Inc.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for Rna-Seq Data with Deseq2." *Genome Biology* 15 (12): 550. doi:10.1186/s13059-014-0550-8.

Pakhira, Malay K. 2009. "A Modified K-Means Algorithm to Avoid Empty Clusters." *International Journal of Recent Trends in Engineering* 1 (1).

Wong, Garrett T, Richard P Bonocora, Alicia N Schep, Suzannah M Beeler, Anna J Lee Fong, Lauren M Shull, Lakshmi E Batachari, et al. 2017. "Genome-Wide Transcriptional Response to Varying Rpos Levels in Escherichia Coli K-12." *Journal of Bacteriology* 199 (7). Am Soc Microbiol: e00755–16.