Clustering Temporal Gene Expressions of Iron-Oxidizing Zetaproteobacteria

Stephen Herbein sherbein@udel.edu
Michela Taufer taufer@udel.edu
Sean McAllister mcallis@udel.edu
Clara Chan cschan@udel.edu

University of Delaware
Newark, DE 19716

ABSTRACT

Zetaproteobacteria are an iron-oxidizing bacteria that play a big role in the rusting of underwater structures. In order to protect these underwater structures, scientists want to better understand the genes involved in the oxidation of iron. To help the scientists narrow down their search for iron-oxidizing genes, we apply two automated clustering methods to quickly and effectively find similar gene expression patterns. We then tune the input parameters of these clustering methods using three different clustering metrics. Finally, we analyze the resulting clusterings to determine which are the most helpful for future scientific investigation. Using this workflow, we are able to isolate three separate clusters of genes, all of whose responses are correlated with the introduction of iron into the bacteria's environment.

1. MOTIVATION

Zetaproteobacteria are an iron-oxidizing bacteria commonly found at deep-sea hydrothermal vents. These bacteria play a big role in the rusting of underwater ship hulls, metal piling, and pipelines. In order to protect these underwater structures, scientists want to better understand the genes involved in the oxidation of iron. The problem with isolating the genes involved in iron oxidation is illustrated in Figure 1. Scientists are normally able to narrow down the search for iron-oxidizing genes by grouping the genes based on their temporal expression. Unfortunately, clustering 2,000 genes by hand is a daunting task for scientists. To help the scientists narrow down their search for iron-oxidizing genes, we apply several automated clustering methods to quickly and effectively find similar gene expression patterns.

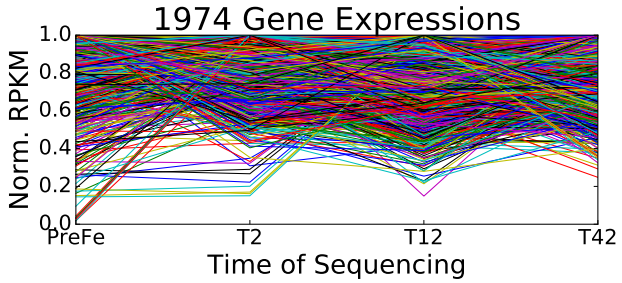

Figure 1: All 1,974 gene expressions within our dataset

2. DATA ANALYSIS WORKFLOW

Our workflow consists of data collection, normalization, cleaning and clustering. We evaluate our clustering methods for the optimal number of clusters and provide two candidates for the optimal number of clusters.

Data Source: Our bacterial samples came from the Loihi Seamount in Hawaii. The samples were preserved on site by Sean McAllister and Anna Leavitt. They were preserved at four different times. The first preservation time was before any disturbance to the environment of the bacteria. After this first preservation, iron was injected into the samples for the zetaproteobacteria to oxidize. The samples were then preserved three times after the injection of iron: 2, 12, and 42 minutes. The preserved samples were then returned to the University of Delaware for genomic sequencing.

Data Normalization & Cleaning: The raw data from the sequencing are counts of each gene expression over the four time points (i.e., PreFe, 2, 12, and 42 minutes). In preparation for clustering the data based on gene expression, we perform two stages of normalization and two stages of cleaning. The first stage of normalization consists of using a common, domain-specific function called Reads Per Kilobase per Million mapped reads (RPKM) [1]. This allows for comparing short genes against long genes as well as comparing genes over time. The second stage of normalization consists of normalizing each gene based on its maximal expression over the four sampled time points. This allows us to compare genes that have differing levels of absolute expression but exhibit the same general pattern of expression over time [2]. The sequencing method used has an error rate such that any read less than 50 is statistically uninteresting [1]. Thus, the first stage of cleaning consists of removing genes that have less than 50 raw reads across all four time points. The sequencing method used occassionally produces large outliers when it fails to filter out ribosomal RNA [1]. To counteract this, the second stage of cleaning consists of removing all genes that have an RPKM greater than 10,000 at any time point.

Clustering Methods: In order to extract patterns from the gene expression data, we utilize two different clustering methods. First, we use k-means clustering; which is a general-purpose clustering method that favors spherical clusters of equal size. K-means is highly scalable with respect to the number of gene expressions and moderately scalable with respect to the number of clusters. Second, we use spectral clustering; which is a clustering method that favors non-spherical clusters of equal size. Spectral clustering is moderately scalable with respect to the number of gene

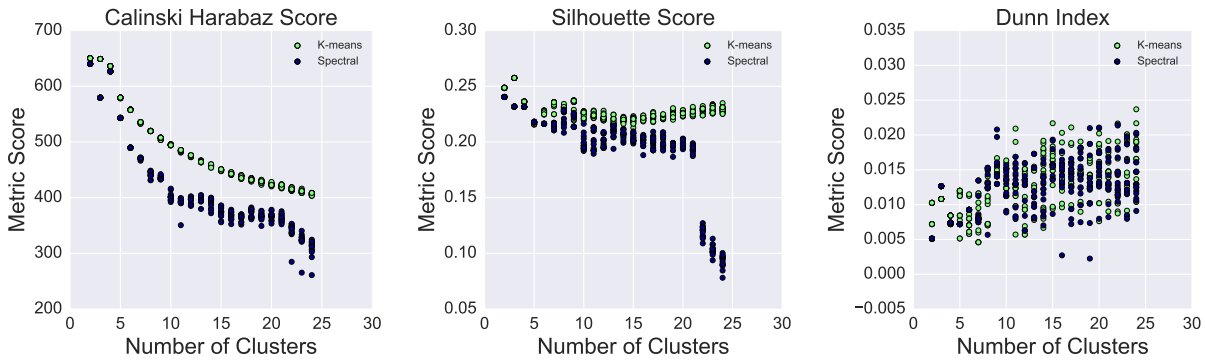

Figure 2: Evaluation of both clustering methods using 3 unsupervised metrics

expressions but not scalable with respect to the number of clusters. Both clustering methods require the desired number of clusters as an input parameter.

Determining the Optimal Number of Clusters: In order to determine the optimal number of clusters for the two clustering methods, we utilize three different clustering metrics: the Calinski Harabasz score, the Silhouette score, and the Dunn index. All three metrics are unsupervised, i.e., they do not need a ground truth to compare against. All three of the metrics are ratios of two criteria: the compactness of the clusters (i.e., how related the gene expressions in each cluster are) and the separation of the clusters (i.e., how distinct each cluster is from every other cluster) [3]. Using these metrics, we can determine the optimal number of clusters by varying the number of clusters and choosing the amount with the highest score. As shown in Figure 2, both the Calinski Harabasz and Silhouette scores indicate approximately 4 clusters is optimal, but the Dunn index indicates that approximately 20 clusters is optimal.

3. RESULTS

Since our clustering metrics disagree on a single optimal for the number of cluster, we investigate using both 4 clusters and 20 clusters under spectral clustering (similar results were seen with k-means clustering). Figure 3 shows the clustering of gene expressions produced by Spectral clustering when run with a target of 4 clusters. Although this particular clustering scores well on two different clustering metrics, it is not useful for further scientific investigation because it has too coarse a granularity. Many unique gene expression patterns are grouped together in cluster #4. It is also hard for scientists to target specific genes for further investigation when the clusters contain over 200 genes. Figure 4 shows the clustering of gene expressions produced by Spectral clustering when run with a target of 20 clusters. This particular clustering scores well with the Dunn index and produces a clustering that is very useful for further scientific investigation. This clustering is more useful because it better isolates the different gene expression patterns from one another. Better isolation allows for a more targeted investigation of the genes involved in the specific expression patterns that scientists find interesting. For example, all the genes in clusters #10, #15 and #20 exhibit a delayed response to the introduction of iron. This delayed response is interesting to scientists because it indicates that those genes are connected to the process of iron oxidation.

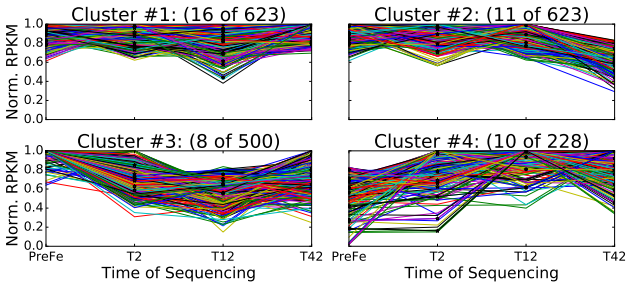

Figure 3: Spectral clustering with 4 clusters

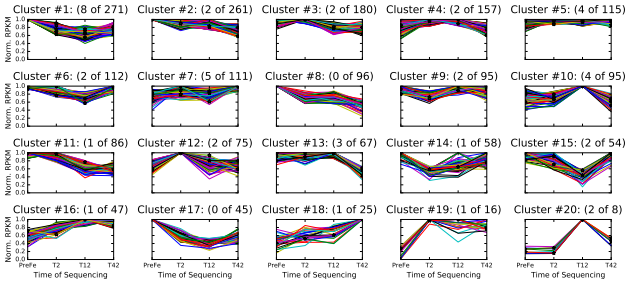

Figure 4: Spectral clustering with 20 clusters

4. REFERENCES

[1] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, and A. Mortazavi. A survey of best practices for rna-seq data analysis. *Genome Biology*, 17(1):13, 2016.

[2] T. N. M. Jewell, U. Karaoz, E. L. Brodie, K. H. Williams, and H. R. Beller. Metatranscriptomic evidence of pervasive and diverse chemolithoautotrophy relevant to C, S, N and Fe cycling in a shallow alluvial aquifer. *The ISME journal*, pages 1–12, 2016.

[3] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916, Dec 2010.