AC297r Project: Harvard - Inari

Targeted Gene Modulation Using Gene Coexpression Networks

Victor Avram, Sergio Miguel Moya Jimenez, M. Eagon Meng, Wenhan Zhang

March 3, 2021

1 INTRODUCTION AND MOTIVATION

As a plant breeding technology company, Inari's work comprises three high-level stages: (1) using computational methods to build genetic knowledge; (2) genetic editing; and (3) delivery of altered genetic information to specific parts of the plant. Our project sits within the first stage of Inari's work and seeks to answer the question "what are the effects on the rest of the maize genes when we perturb a subset of the maize genes?" Essentially, we hope to construct a genetic network that can act as a look-up table to inform Inari of the genetic effects and side-effects of perturbing particular maize genes, as opposed to only observing unexpected effects later in the breeding process.

From the outset, we are aware of a few challenges. To illustrate these, we can use as reference the Connectivity Map project (Subramanian et al, 2017) which achieved 81% accuracy when predicting the rest of the human genome from the 1,000 landmark genes.

Firstly, our dataset is not perturbation-driven such as that in the CMap. By profiling specific genetic perturbations, CMap was able to link variations in genetic profiles directly to specific genes. Secondly, our dataset of 480 tissue samples across 25 individuals is significantly smaller 12,031 genetic profiles CMap analyzed. Lastly, CMap made inference on 11,350 human genes, while we were initially tasked with predicting the rest of the 46430 maize genes.

Therefore, with these challenges in mind, our EDA has the goals of firstly, reducing the number of relevant genes to be inferred therefore reducing the prediction task, and secondly, observing patterns and linkages in genetic expressions so as to determine whether there is sufficient data for us to create the maize counterpart of CMap.

2 DATA

The data is comprised of gene expression values derived from 26 individual maize plants. Multiple samples were taken from 10 different tissues from these 10 individuals and subsequently sequenced in order to obtain gene expression counts. The number of samples contributed

varies per individual and varies per tissue. As well, samples were collected at different developmental stages. The samples collected from 1 of the individuals are set aside for validation (referred to as the test set), leaving the samples collected from the remaining 25 individuals as the data to be used for model creation and model improvement (referred to as the training set). These datasets are provided in tabular form. Each entry represents the quantification of the expression level for the given gene in the given sample. Therefore, the set of genes analyzed is consistent across all samples. The training set contains 480 samples, each with expression levels for 46430 genes. The test set contains 21 samples, each with expression levels for the same 46430 genes found in the training set.

The raw gene expression counts are first converted to transcripts per million (TPM). The steps of this preprocessing technique are as follows: 1) Divide the expression level by the length of the given gene in kilobases, 2) Divide by the summation of gene length normalized expression levels in the given sample, 3) Multiply by 1,000,000. The process of converting raw counts to TPM first normalized the expression levels by the gene length. More fragments are likely to map to longer genes and vice versa for shorter genes. These gene length normalized expression levels are then adjusted for sequencing depth. Sequencing depth refers to the total counts attributed to a given sample. Given the inability to sequence all samples at the same time and with the same sequencer, sequencing depth normalization is an important step in being able to make comparisons across samples. Lastly, the expression levels are scaled by a large factor.

3 DATA PREPROCESSING

TPM expression data is normalized by gene length and by sequencing depth, making it amenable for many downstream gene expression analyses. However, further preprocessing is often done before performing these analyses. Gene expression data is typically skewed with a relatively low number of very high values. As well, it is more biologically meaningful to look at proportional discrepancies as opposed to additive discrepancies. The expression data was $\log_2$ transformed with an offset in order to handle 0 values. The log-transformed data log-TPM is derived as follows for every a_{ij} entry in the $n \times p$ expression matrix.

$$\text{transformed } a_{ij} = \log_2(a_{ij} + 1), i = 1, \ldots, n, j = 1, \ldots, p$$

Lowly expressed genes often do not provide a substantial amount of information and do not elucidate relationships between groups or genes. For example, removing lowly expressed genes when performing differential expression between groups of samples will increase the power to detect significant differences given that the correction for multiple comparisons will be less stringent. Genes were deemed as lowly expressed if they met the following criteria: 1) No expression in at least 80% of samples, 2) The maximum expression across all samples was less than or equal to 2 TPM. 7733 genes were considered as lowly expressed and subsequently removed, leaving 38697 genes.

4 DATA EDA

4.1 Gene Expression Distributions

After data preprocessing, the data was first analyzed by looking at the mean expression levels across all genes. These distributions can be found in Figure 1. Mean expression levels from all individuals follow a similar distribution.

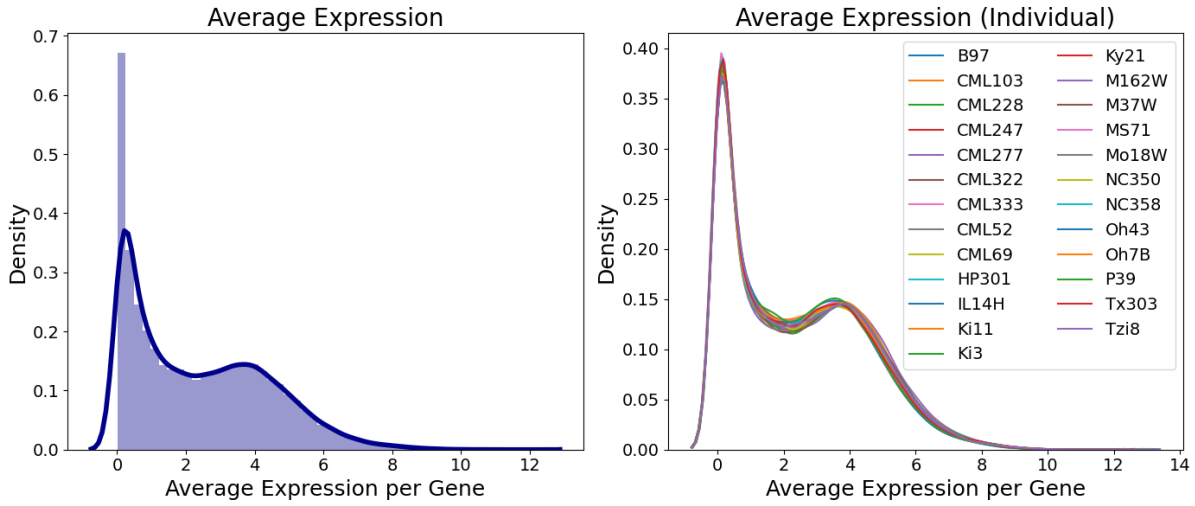

Figure 1: Distributions of the mean expression level per gene. The plot on the left shows the distribution when taking the mean expression level across all samples. The plot on the right shows 25 distributions derived from taking the mean expression level across samples from a particular individual.

Picking the top g most expressed genes to serve as a feature set for predictive modeling may seem to be an advisable initial selection criterion. However, highly variable genes adjusted for the mean are likely to serve as better predictors. Figure 2 shows the relationship between the coefficient of variation (CV) and mean, as well as the distribution of CV values.

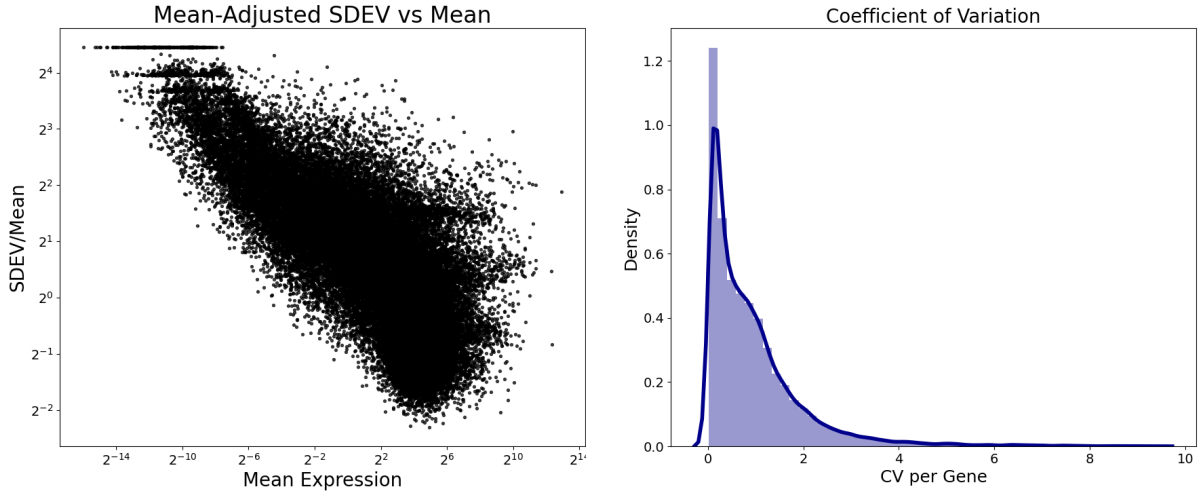

Figure 2: Left) Coefficient of variation vs mean for the expression level across all genes. Right) The distribution of the coefficient of variation values. The expression data used are unprocessed TPM values.

4.2 Dimensionality Reduction Visualizations

Dimensionality reduction techniques were used in order to quickly assess whether or not the data clustered based on certain sample metadata. Principal component analysis (PCA) was

performed on the preprocessed expression data. As well, t-Distributed Stochastic Neighbor Embedding (tSNE) was performed on the preprocessed expression data. TSNE is a nonlinear dimensionality technique that is often used when analyzing single-cell data, but can be applied to any high-dimensional data. The main objective is to preserve the local structure of the data, while tSNE's main pitfall is its inability to preserve global structure and therefore its ability to draw conclusions from between-cluster distances. Explanations of these dimensionality reduction techniques can be found in the Appendix. Figures 3 and 4 show PCA and tSNE plots colored by individual and by tissue (organism part). Figure 4 indicates that there is distinct clustering by tissue. For this reason, it may be advisable to perform separate analyses and create separate predictive models per tissue.

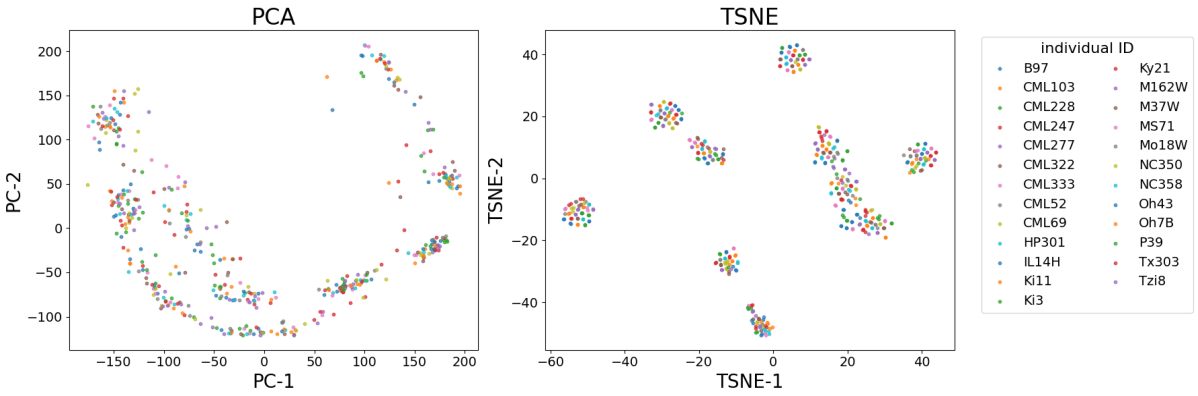

Figure 3: PCA and tSNE dimensionality reduction were used on preprocessed TPM expression values. The datapoints are colored by individual ID. There is no obvious clustering based on individual ID.

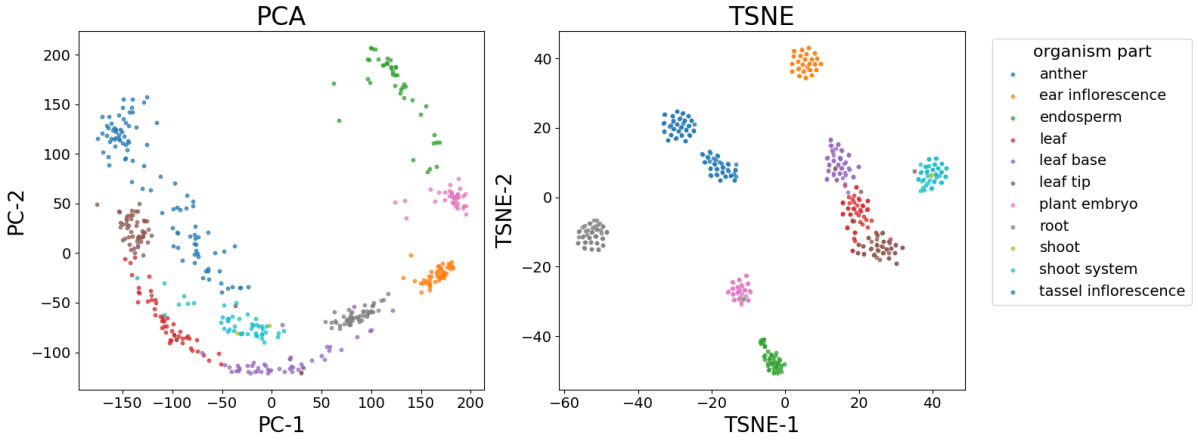

Figure 4: PCA and tSNE dimensionality reduction were used on preprocessed TPM expression values. The datapoints are colored by tissue. Distinct clustering based on tissue is present.

•

5 NETWORK CONSTRUCTION EDA

Proper construction of the gene coexpression network is critical if network/graph based predictive models will be used. Gene coexpression networks, regardless of the method used for network generation, consist of the same basic components. Nodes represent genes and edges represent relationships between genes. These networks can either be weighted or unweighted and signed or unsigned. Weighted networks assign quantities to the edges, these quantities being proportional to the strength of the relationship between the given genes. Unweighted networks do not assign weights to edges and therefore use hard-thresholding (i.e. a cutoff that dictates which edges should be dropped from the network) in order to prune unwanted edges. Signed networks maintain information regarding the directionality of the relationship between two given genes. Unsigned networks strictly contain edges with non-negative weights and therefore cannot be used to discern the directionality of a given relationship.

5.1 WGCNA

Weighted Correlation Network Analysis (WGCNA) is a widely used tool to understand gene expression data, the relationships between genes, and the relationships between sample metadata and gene clusters. Given $p = 38697$ genes in the preprocessed expression dataset, a 38697×38697 adjacency matrix A, derived from a 38697×38697 correlation matrix, serves as the basis for WGCNA network construction. Each element in the adjacency matrix a_{ij} defines the relationship between nodes i and j. An intermediate value s_{ij} is calculated and denotes the strength of the relationship. s_{ij} for unsigned and signed networks are given below on the left and right respectively.

$$s_{ij} = |cor(\text{gene}_i, \text{gene}_j)| \qquad\qquad s_{ij} = cor(\text{gene}_i, \text{gene}_j)$$

Hard thresholding creates an unweighted network where $a_{ij} = 1$ given $s_{ij} \geq \tau$ and $a_{ij} = 0$ given $s_{ij} < \tau$ for the hard thresholding paramter τ. However, weighted networks created by using soft thresholding are better suited for modeling the continuous nature of biology and are therefore preferred. The adjacency matrix is defined as follows for a soft thresholding power β.

$$a_{ij} = s_{ij}^{\beta}, \beta \geq 1$$

WGCNA was performed on the data, generating an adjacency matrix that can be used for graph-based predictive modeling. The choice for β which was determined to be 15. The optimal value for β is one that maintains a scale-free network topology. Network connectivity that follows a power law distribution (i.e. relatively few nodes are highly connected and many nodes are lowly connected) is consistent with known biological interactions. A small proportion of proteins act as "master regulators" that interact with many different counterparts, activating or deactivating proteins, and upregulating or downregulating genes. Until graph-based predictive models are implemented, assessing the validity of this network will not be possible.

6 PREDICTIVE MODELING

A key discovery is that the most significant differences in the genetic profiles is caused by the difference in the organism parts. In other words, organism parts, more than the differences in individuals, are the main driver of differences in genetic expression levels.

The logical direction would then be to analyze genetic profiles of different organism parts separately. To put this into an analogy for humans, it would not be relevant to analyze the genetic

profiles of an eye cell, neuron and skin cell combined; up-regulating the genes that are specific to neurons would not be a relevant predictor of genetic expression of the rest of the genes in a skin cell.

6.1 Regression Methods

As a first step, we decided to apply OLS "Ordinary Least Square Models" using the 1000 most expressive genes to predict each one of the next 1000 ones. After eliminating a total of 2,432 genes which were not expressed at all across in none of the samples, we sorted the genes in two ways: 1) Using the average of expression across all samples and 2) Getting both the standard deviation and average of expressions across all samples per gene, dividing them and applying a sorting. This allowed us to account for different scales across genes, as it would to rank by variability in a more representative way.

In total we ran 1000 models per method using a train-test split of 70-30 (i.e., 70% for the training set and 30% for the test set). The models were fitted using the training set and R^2 of the test sets were obtained for further analysis of accuracy.

For our first method of sorting, the reader can see below the summary statistics for the R^2 for all models run. As it can be seen, the results were really good in general.

Name	Value
Count	1000
Mean	0.747
Standard Deviation	0.285
Min	-2.521
25th Percentile	0.694
50th Percentile	0.829
75 Percentile	0.906
Max	0.992

Table 1: Summary Statistics R^2 without considering for scaling

This contrasts the results obtained using the second method of sorting, which can be seen in the table below:

Name	Value
Count	1000
Mean	-1.8995e+16
Standard Deviation	5.2427e+17
Min	-1.6541e+19
25th Percentile	-2.4124e+02
50th Percentile	-1.5933
75 Percentile	-1.4337e-01
Max	9.2419e-01

Table 2: Summary Statistics R^2 considering for scaling

In general, almost all R^2 resulted to be negative, which allows us to conclude that the models resulted to be worst predictors than horizontal lines. Given this reason, we decided to apply several more complete models that will allow us to do better predictions, which are going to be discussed in the next sections of this report.

7 FURTHER MODELING

7.1 Problem Analysis

One of the central motivations behind our problem is the attempt to understand the full relationship graph of all expressed genes, or more specifically, reconstructing some kind of Gene Regulatory Network. GRN's in the literature have been well studied for more than a decade, but the area remains largely an "unsolved problem" (Saint-Antoine et al., 2019) and is complicated by a bevy of computational challenges.

More formally, work on GRNs fully encompasses our current problem setting as follows: we consider N genes and represent their expression levels (the entirety of the data we have) as random variables $\{X_1, X_2, ..., X_N\}$. Each X_i thus represents a node in our network, and any relation $r(X_i, X_j)$ describes an edge where r models the regulatory relationship between the two genes. These relations can be modeled as directed, implying causality, or with signs, hence representing downregulation or upregulation.

This formal description of the problem leads to the natural formation of any model that wishes to describe a GRN as some subset of all possible relations $R = \{r \mid r : (X_i, X_j) \mapsto s\}$, where s is typically some score representing the importance of that relation. Indeed, the vast majority of methods in reconstructing GRNs use precisely such a metric, and one of the most popular competitions Dream Challenges ([homepage - Dream Challenges](http://dreamchallenges.org/)) provides gold-standard evaluation data where the target labels are exactly these edge importance values, as verified experimentally or through other modalities.

One key discrepancy between our current problem formulation (i.e. regression on subset of most highly expressed to find expression levels of the rest) and the existing literature is that our evaluation metric is directly based on expression levels, while all other existing methods try to evaluate based on real biological understanding of gene regulatory pathways. There are a number of clear downsides to metrics based purely on expression levels:

- Consider genes with low variability (almost always expressed at a constant level, or almost always 0, etc.). All aggregate metrics of accuracy over any test set Y would be arbitrarily inflated in accuracy by inclusion of any low-variability genes as all regression methods would simply set feature coefficients to 0 and be a constant predictor.
- Even with a reasonably accurate regressor, it is not trivial to then extrapolate that information and establish which genes are related, what sub-networks exist, what kind of motifs of regulatory action are revealed, etc. Physiologically speaking, we have no experimental variation in our data, and cannot expect to capture consistent patterns tested by perturbation, and rely solely on random individual variation.
- As follows, it will also be hard to generalize the utility of results based on cross-validation of these metrics, given the two-fold difficulty of lacking experimental variation as well as any basis of interpretability.

7.2 Existing Approaches

There are a number of classes of approaches in reconstructing Gene Regulatory Networks, each of which begin with the same kind of data that we have (purely gene expression levels, without other modalities of data). Each of these are potentially interesting avenues to pursue,

and are worth further experimentation and consideration in our attempt to derive genuinely interesting insight from our data. Figure 5 from Saint-Antoine et al. (2019) describes each class with their respective qualities and advantages/disadvantages.

Algorithm Class	Temporal Data Required?	Directionality	Advantages	Disadvantages	Examples
Correlation	No	Undirected	• Fast, scalable • Detection of feed-forward loops, fan-ins, and fan-outs	• Possibly over-simplistic • False positives for cascades	WGCNA [13] PGCNA [14]
Regression	No	Directed	• Good overall accuracy	• Bad detection of feed-forward loops, fan-ins, and fan-outs	TIGRESS [15], GENIE3 [16], bLARS [17]
Bayesian - Simple	No	Directed	• Performance on small networks	• Performance on large networks. • Inability to detect cycles	[19,20]
Bayesian - Dynamic	Yes	Directed	• Performance on small networks • Detection of cycles and self-edges	• Performance on large networks.	[21]
Information Theory	No	Undirected (at least in simplest form)	• Detection of feed-forward loops, fan-ins, and fan-outs • Similar to correlation methods, with better accuracy	• False positives for cascades	ARACNE [25], CLR [26], MRNET [27], PIDC [28]
Phixer	No	Directed	• Parsimonious output due to pruning step.	• Possible loss of overal accuracy due to pruning step (this can be removed if the user chooses)	[31]

Figure 5: The classes of models used throughout literature in attempting to reconstruct Gene Regulatory Networks.

- **Correlation:** This class of network inference works purely off correlation matrices calculated from the data, and uses varying threshold strategies to try to establish network edges. Baseline models involving WGCNA fall into this category, though their use in predictive settings will devolve equivalently into basic linear regression.
- **Regression:** The setting that most resembles our problem statement. Models in this class like TIGRESS and GENIE3 all work the same way by predicting the expression of a leave-one-out gene from the rest of the expression levels, using some kind of regressor (Least Angle Regression for TIGRESS, and random forest for GENIE3). This is a potential approach for the most direct next step of our modeling.
- **Bayesian:** Either classified simple or dynamic, this class of analysis is computationally expensive and thus well suited for smaller sized networks, but perform quite well in the literature. In the case of dynamic Bayesian networks that incorporate temporal or pseudo-temporal features, self-regulating genes (modeled by self-edges) can also be discovered.
- **Information theory:** Similar most to correlation approach, these models use mutual information or Shannon information to understand distributional characteristics and test the

importance of gene relations under this framework. A popular approach that can deal with multi-step relations, unlike regression approaches.

7.3 Graph Neural Networks

We have done extensive research into many classes of potential Graph Neural Networks to use, and will look into compatibility for the next milestone: StellarGraph, PyTorch Geometric, Deep Graph Library to name a few complete code libraries that exist with good candidate selections. The primary difficulty of usage based on current experimentation is formation of the data and in particular evaluation strategy: prediction on single-attribute nodes is not a very common task in generic graph neural networks currently, and will require an effort of coadapting the models.

To be more specific, graph neural networks aim to solve the following classes of problems:

- **Node classification:** where we infer the class of nodes, e.g. tissue type. We can adapt this to a regression setting, but given that each node's data is essentially one-dimensional (just expression value), this has not been the easiest fit as many networks in this class expect wide nodes with a fair amount of information in each.
- **Link prediction:** inference on missing edges and the relationships of nodes. This is much more prevalent in work in research and pertinent for describing GRNs, but our predictive setting cannot leverage this class of models unless we redefine objectives.
- **Community detection:** this would be an interesting avenue to pursue to do unsupervised clustering of related genes, but does not directly contribute to predictive efforts. Worth keeping on the horizon as potentially a preprocessing step.
- **Graph classification:** classification on the level of an entire graph (i.e. some phenotype given full expression), less relevant as we have no labeled data to this effect.

Combining the power of graph neural networks with the existing approaches in literature in reconstructing GRNs is one of our immediate future directions, and a potential area for novel research and genuine contribution given that we can resolve formulations of evaluation metrics in a way that can adapt to network architectures.

8 APPENDIX

The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

Let X be the matrix of all the independent variables (in this case, the genes) where each column is a gene and each row corresponds to a sample of maize. We then calculate the mean for each column and subtract the respective mean from each variable. This gives us a matrix whose columns has zero mean. We then divide each data with the standard deviation of each columns to center the matrix. We call this centered matrix Z. Covariance matrix of Z is then calculated by multiplying it with Z^T, which is the transpose of matrix Z.

After obtaining the covariance matrix of the centered matrix Z, we then find its eigenvalues and eigenvectors. Since the covariance matrix is a semi-definite matrix, the eigenvalues and eigenvectors can be calculated by eigen value decomposition. The eigen decomposition of $Z^T Z$ is where we decompose $Z^T Z$ into PDP^{-1}, where P is the matrix of eigenvectors and D is the diagonal matrix with eigenvalues on the diagonal and values of zero everywhere else. The eigenvalues on the diagonal of D will be associated with the corresponding column in P — that is, the first element of D is λ_1 and the corresponding eigenvector is the first column of P. This holds for all elements in D and their corresponding eigenvectors in P. We will always be able to calculate $PDP^{-}1$ in this fashion. Take the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$ and sort them from largest to smallest. In doing so, sort the eigenvectors in P accordingly. We call this sorted matrix of eigenvectors P^*. (The columns of P* should be the same as the columns of P, but perhaps in a different order.) Note that these eigenvectors are independent of one another.

We then calculate the matrix with principal components Z^* = ZP^*. This new matrix, Z^*, is a centered/standardized version of X but now each observation is a combination of the original variables, where the weights are determined by the eigenvector. Since columns of P^* are independent, the columns for Z^* are also independent.

One downside from the PCA is that it might lead to poor visualization especially when dealing with non-linear manifold structures. Instead of being concerned with preserving large pairwise distances to maximize variance as the PCA, the TSNE focuses on preserving only small pairwise distances or local similarities. It calculates similarity measure between pairs of instances in the high dimensional space and in the low dimensional space, and uses a cost function to optimize between these measures.

Saint-Antoine, M. M., Singh, A. (2020). Network inference in systems biology: recent developments, challenges, and applications. Current Opinion in Biotechnology, 63, 89–98. https://doi.org/10.1016/j.cop

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., Davis, J. F., Tubelli, A. A., Asiedu, J. K., Lahr, D. L., Hirschman, J. E., Liu, Z., Donahue, M., Julian, B., Khan, M., Wadden, D., Smith, I. C., Lam, D., Liberzon, A., ... Golub, T. R. (2017). A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. Cell, 171(6), 1437–1452.e17. https://doi.org/10.1016/j.cell.2017.10.049