UNSUPERVISED LEARNING ON SINGLE-CELL TRANSCRIPTOMIC DATA FROM ZEBRAFISH EMBRYOS

Anthony Degleris
degleris [at] stanford.edu

Spencer Guo
scguo [at] stanford.edu

Clara Kelley
clkelley [at] stanford.edu

May 7, 2020

ABSTRACT

Recent advances in high-throughput sequencing have generated unprecedented amounts of transcriptomic data for complex organisms such as zebrafish. However, analysis of these data is challenged by numerous issues such as high-dimensionality and technical and biological noise. We describe in this paper an unsupervised learning approach which attempts to robustly derive information about developmental processes and genetic archetypes in sequencing data from zebrafish embryos. Performing dimensionality reduction with feature selection and fitting low-rank models (singular value decomposition and non-negative matrix factorization) enabled identification of key genes across the time points and generation of sparse, interpretable transcriptomic archetypes. Furthermore, we discovered that latent Dirichlet allocation, a technique from natural language processing, allowed probabilistic interpretations, though at the expense of sparsity.

1 Introduction

One of the major goals of developmental biology remains to identify and track cell lineages from zygote to adult, to understand the mechanisms which underlie differentiation, patterning, and cellular fate. Because of recent technological advances in next-generation sequencing, researchers can now more easily map and quantify transcriptomic data through tools such as RNA-Seq. In addition, researchers can also determine single-cell expression profiles through single-cell RNA-Seq (scRNA-Seq), providing unparalleled detail of RNA expression between cells.

In particular, high-throughput methods and microfluidics have enabled single-cell mapping of tens of thousands of cells, as shown by a recent paper from Wagner and colleagues. In their work, they collected over 92,000 single-cell transcriptomes of zebrafish (*Danio rerio*) in the first 24 hours of development.[14] The massive amounts of data from these scRNA-seq experiments provide fertile ground for analysis. Because of the size of the datasets and the fact that scRNA-Seq data is often affected by technical issues with transcriptome collection, unsupervised learning techniques are often useful.

1.1 Task Definition

In this paper, we describe computational approaches towards better interpreting and analyzing the massive datasets generated by scRNA-seq data. In particular, we attempt to reduce the dimensionality of the high-dimensional gene expression data and glean meaninful insights about the underlying data structure. We break this up into two subproblems. First, we would like to identify the *most influential genes* in explaining transcriptome variance, which is directly addressed via the methods in Section 2, and also indirectly addressed through the methods in Sections 3 and 4. Second, we would like to identify *transcriptome archetypes* that naturally characterize the common cell types, which is the primary objective of Sections 3 and 4. Our results generally demonstrate challenges in analysis of data which often contains large amounts of biological and technical noise. In particular, scRNA-Seq data suffers from issues of dropout and low sequencing depth. Previous studies have attempted to address these issues by using clustering approaches such as k-means and Gaussian mixture models.[11]

1.2 Data

The dataset used by Wagner, et al. is available online at the Gene Expression Omnibus (GEO). The sequencing data was collected using the inDrops scRNA-seq technology, and the researchers obtained transcriptomes from dissociated wild-type zebrafish embryos during the first 24 hours of development.[14] For each cell in a sample the dataset provides the frequency of each gene expression in the features. In total, there are $G = 30,678$ genetic features (the mRNA's used in the RNA-seq experiments) The number of transcriptomes analyzed C_t for each time point (measured in hpf, hours post-fertilization) is listed in Table 2 (taken directly from [14]).

Since our dataset is sparse (approximately 90% of the entries are zero), we can represent the data using sparse arrays in our code and take advantage of sparse algorithms. Notationally, we represent the data at a given point in time t as a $C_t \times G$ matrix $X^{(t)}$. In particular, the jth row of $X^{(t)}$, corresponding to the expression profile of a single transcriptome at time t, will be denoted $x_j^{(t)} \in \mathbf{R}^G$.

2 Feature Identification

2.1 Background

This dataset is particularly high-dimensional because of the number of features. To work with the data most effectively, we needed to reduce the number of features down to a more workable percentage of the overall dataset. While this subsection of features will be used for further analysis in sections 3 and 4 of this research, feature selection itself can also serve an additional purpose. If we could uncover the most important or most relevant genes to this dataset, those could serve as a jumping off point for further genetic research.

Feature identification is the first angle of our approach to understanding the power of this dataset. Our goal is to determine key genes for further study using unsupervised learning methods.

2.2 Approach

For a baseline measurement at the start of this project, we simply selected the top 50 genes by contribution to variability in the dataset. As an oracle, we considered all the genes that contribute to the first 50 principle components of the dataset.

For the first pass at isolating the subset of gene expressions that would be most useful for this research, we used a variance thresholding approach. Variance-based elimination is the simplest method for feature selection. Given a threshold, the algorithm removes any feature with less variance over the dataset. For example, a gene with no variance (that has the same value for all samples in the dataset) would be removed at any threshold level. This method removes features that are particularly common and thus likely provide little new information for us to analyze. However, this method does not take into account the interaction between features, and is such not complex enough for identifying key genes for further study.

To isolate individual genes with interesting relationships to the dataset we used feature agglomeration. Feature agglomeration is a method for combining related features to reduce dimensionality of the data. The algorithm recursively merges features that have similar relationships in the data. When clustering samples rather than features, this is done by initializing each sample as a cluster and merging the two clusters together that minimally changes the variance between the members of each cluster, and repeating this recursively until a target number of clusters is reached. When this clustering is done with features, the same principles apply. This amounts to grouping gene expressions that appear frequently together.

For example, we ran Feature Agglomeration on the reduced dataset to create fifty gene groups out of 5931 identified by variance elimination. Only 6 of those clusters had a substantial number of features. Clusters 13 and 15 comprised of 2515 and 1831 genes respectively. Clusters with very few or a single gene on the other hand serve a unique purpose in the dataset. This suggests that these genes might be driving genetic factors worth investigating.

RNA-seq enables a researcher to analyze unusual and uncommon cell types as well as key genes. For this reason, isolating cell samples that are outliers in the dataset is key to identifying undiscovered cell types and cell mutations. Additionally, outlier detection is a useful pre-processing step preparing the data for further analysis down the road. For an additional selection experiment, we evaluated three different methods for identifying outlier or novel samples in the dataset. The three methods are One Class Support Vector Machine (SVM), Isolation Forest, and Local Outlier Factor. One Class SVM is a form of SVM that treats all data as belonging to a single classification, and determines outliers by degree of fit to model. Isolation Forest recursively determines a decision boundary for each feature such

as to have a particular percentage of outliers. Local Outlier Factor relies on a nearest neighbors model to eliminate samples over a certain threshold of distance to nearest neighbor.

2.3 Experiments

For each of these selection approaches: removing extraneous features, isolating key genetic factors, and identifying outlier cell samples, we tested a few different variations.

Variance Threshold

We experimented with different forms of variance thresholds to produce the best results. At a variance threshold of 25%, variance based elimination kept only 3759 of the original 30000+ features. At 20%, only 4200 features kept. At 10%, 5931 features were kept. At 5%, 8076 were kept. At 0%, or removing only features that appeared in the data with exactly one value, 29226 features were kept.

Feature Agglomeration

For identifying key features, we experimented with two different variables that affect the final feature agglomeration clusters: the number of original features in the dataset, the number of target clusters.

At a 10% threshold for variance, there are 5933 features remaining. With a target of 50 clusters, only 6 of those clusters had a substantial number of features. Clusters 13 and 15 comprised of 2515 and 1831 genes respectively. There were 31 clusters comprised of a single gene. With a target of 25 clusters, clusters 13, 7, and 2 contained a thousand or more genes. There were 12 clusters comprised of a single gene.

When 14606 genes (given a 1% threshold for variance) were recursively merged, three of fifty clusters were large: Clusters 11, 7, and 14 comprised 8756, 3214, and 1498 genes respectively. There were 31 clusters comprised of a single gene. With a smaller target of 25 clusters, clusters 11, 3, and 9 comprised 8756, 3214, and 1502 genes respectively. There were 12 clusters comprised of a single gene.

2.4 Analysis

Variance Threshold

A variance threshold is a very simple way to reduce dimensionality. It is difficult to determine if this method of preparing the dataset eliminates important information for further analysis. Using a very low threshold must be balanced with retaining irrelevant data that obscures important findings. There is a non-linear relationship between threshold and the number of features below that threshold; this holds with our knowledge about the sparsity of the data - many genes appear extremely infrequently across all cells if at all.

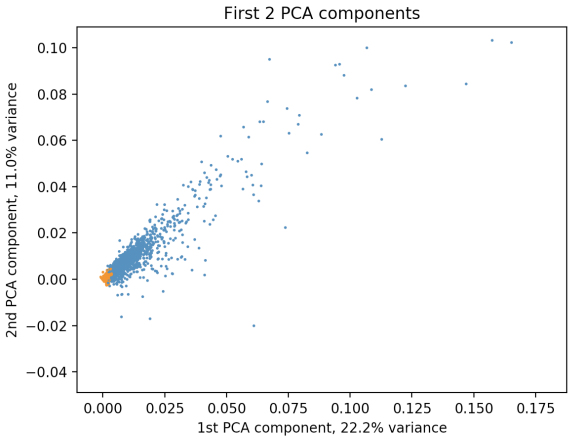


Figure 1: First two principle components of the full dataset at hour 4, with features below a 25% threshold for variance highlighted in orange (90%+ of the features)

Feature Agglomeration

To determine whether feature agglomeration is correctly identifying important and relevant genes, we can compare the genes in small clusters to NCBI's database of genomic information. The API Entrez allows researchers to program-

matically access information from their databases. For analysis of this research, we used number of related PubMed entries (scientific papers that mentioned or referenced a particular gene) as a proxy for the scientific relevance of that gene.

In the following figure we can see the average number of pubmed entries for the average gene in a cluster of each size: notice that these smaller clusters have genes with on average thousands of entries, where a random gene from the entire feature set has on average 494 entries (median: 1 entry per gene). This indicates that the feature agglomeration method might be useful for identifying important genes - the dataset used is entirely unsupervised, and thus the algorithm could not have selected genes based on known scientific value.

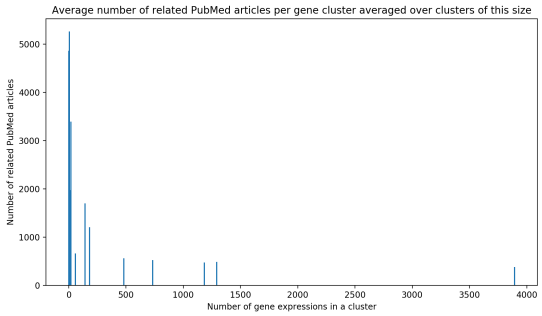


Figure 2: Bar chart of average PubMed article count in relation to gene cluster size

Outlier Cells

Despite initializing each model with the same contamination factor, the different models for detecting outliers produced very different sets of outlier samples. One class SVM and isolation forest had about 50% overlapping samples, but one class SVM and local outlier factor had an overlap of about 10%. This may mean that either the dataset is particularly noisy (which is possible with such a technologically complex technique as RNA-seq) or that further research is needed into the right outlier detection algorithm for this application. Perhaps more domain knowledge of the error behavior of RNA-seq transcriptomes might aid in the identifying of outlier samples.

3 Creating Genetic Archetypes: Low Rank Models

3.1 Background

In [14], two key steps in analyzing the zebrafish dataset were (1) projecting the data into a low-dimensional space using tSNE and (2) clustering transcriptomes using k-means. The first step, embedding the data in a low-dimensional space, comes at the cost of interpretability: while we can concretely make sense of two cells differing in their expression of a particular gene, it's difficult to understand what deviation in a tSNE axis means physically. The second step, clustering, is limited in its expressive power: many cells exhibit hybrid gene expression behavior that is not captured by hard clusters. For example, all cells may exhibit some 'global' expression behavior (e.g. expressing genes that construct basic proteins), some 'system-wide' expression behavior (e.g. neural cells), and some 'specific' expression behavior (e.g. the neural crest). Ideally, we would be able to say that the transcriptome of an individual cell 'belongs' to the global group, the system-wide group, and the specific group, but k-means only allows us to assign a single group to each cell. In this section, we seek to address both these limtations simultaneously through the lens of low-rank models.

3.2 Model and Algorithm

To address these limitations, we propose using low-rank models to simultaneously (1) project the data into a low-dimensional, but still interpretable, space, and (2) create expression archetypes that define the common RNA expression behaviors but still allow individual cells to exhibit multiple types of expression behavior. Indeed, if we define $K \ll G$ archetypes $h_1, \ldots, h_K \in \mathbf{R}^G$, then let each transcriptome $x_j^{(t)}$ be a linear combination of the archetypes, i.e. $x_j^{(t)} = H^T w_j$ for a $K \times G$ matrix H and some vector $w_j \in \mathbf{R}^K$, we can use the coefficients of w_j as our low-dimensional embedding. This can be concisely expressed using the following model: suppose $X^{(t)} = W^{(t)} H^{(t)} + E^{(t)}$, where $W^{(t)} \in \mathbf{R}^{C_t \times K}, H^{(t)} \in \mathbf{R}^{K \times G}$, and $E^{(t)} \in \mathbf{R}^{C_t \times G}$. The rows of $H^{(t)}$ are

Timepoint (hpf)	4	6	8	10	14	18	24
SVD	0.558	0.294	0.263	0.232	0.280	0.255	0.169
NMF	0.561	0.296	0.265	0.232	0.294	0.259	0.172
NMF with ℓ_1	0.565	0.297	0.266	0.236	0.284	0.259	0.172

Table 1: Losses for NMF, SVD, and sparse NMF (ℓ_1-regularizer weight set to $\lambda = 10$) for each timepoint.

the archetypes defining the data, the rows of $W^{(t)}$ are the expressions of each transcriptome (as a combination of archetypes), and $E^{(t)}$ is a (hopefully) small noise matrix. Note that $W^{(t)}$ gives us our low-dimensional embedding, and $H^{(t)}$ gives us interpretable archetypes that define soft clusters (i.e. a cell may partially belong to several clusters). Thus, we can address both (1) and (2) with a single model!

To fit this model, we minimized the objective function $J(W^{(t)}, H^{(t)}) = \|W^{(t)}H^{(t)} - X^{(t)}\|_F^2$, where $\|\cdot\|_F$ is the standard Frobenius norm. Using this objective function, we computed $W^{(t)}, H^{(t)}$ efficiently using the truncated singular value decomposition (SVD), which guarantees a globally optimal solution to this objective (oracle). However, we found the solution generated by the SVD is undesirable for interpretability reasons. In particular, the constraint that different factors must be orthogonal to one another led to archetypes that were not sparse, unlike the data, and had negative values, which does not make sense for RNA expression data. To fix this, we used nonnegative matrix factorization (NMF), we requires $W^{(t)} \geq 0, H^{(t)} \geq 0$, where the inequality is to be interpreted element-wise. First introduced in [8], NMF is known to produce more interpretable factors for nonnegative data. Moreover, NMF promotes sparse factorizations when the data is sparse, since the model does not allow weighting an archetype negatively, and thus having many nonzero entries will quickly drive up the reconstruction loss. NMF has also previously been used to model gene expression data [6, 9]; however, all the research we encountered looked at gene expression data across a population, instead of gene expression across multiple cells in a single organism. Finally, to encourage both sparse fits similar to the data and eliminate unnecessary factors, we used a sparsity regularizing when fitting NMF. In particular, we added a penalty that scales with $\|H\|_1$ and $\|W\|_1$ by some weight λ, which we chose experimentally by assessing the point in which the loss $\|X - WH\|_F$ began to rapidly increase.

3.3 Experiments and Analysis

We set $K = 50$ and applied both NMF and SVD to the data at every timestep. We used reconstruction error given by $J(W^{(t)}, H^{(t)})$ as a measure of accuracy, and normalized our loss by dividing by the norm of the data to allow comparison across timepoints. We then tested multiple values of λ to weight the ℓ_1-regularizer for NMF, increasing the weight by powers of 10 each time. We chose $\lambda = 10$ as it was the point in which the reconstruction began to increase rapidly. These results are displayed in Table 1.

Three findings were significant from these experiments. First, NMF performed nearly as well as SVD despite the nonnegativity constraint (and the fact that a local minimizer was used, since NMF is a nonconvex objective). This is extremely rare, and suggests that the data is well modeled as a mixture of archetypes. In Figure 3, the 5 most commonly expressed archetypes at 18hpf produced by SVD, NMF, latent Dirichlet Allocation (LDA), and NMF with ℓ_1-regularization are plotted. Note that, unlike SVD, NMF produces sparse factors that look similar to expression of actual transcriptomes. Second, adding ℓ_1 regularization eliminated many of the small, insignificant coefficients found in regular NMF, despite achieving nearly the same reconstruction error. This agrees with our hypothesis that the underlying transcriptome archetypes should also be sparse, like the transcriptomes themselves. The third finding was that, across timepoints, the archetypes appear to be similar. This finding suggests that we may be able to draw relationships between $H^{(t)}$ and $H^{(t+1)}$, i.e. the factors have structure that holds across timepoints.

In summary, we believe NMF to be an effective method for modeling RNA transcriptome archetypes because it encourages sparse, nonnegative archetypes. Moreover, these archetypes form 'soft' clusters, which in turn allows a cell to exhibit the expression behavior of multiple archetypes, supporting varying levels of specialization and hybrid cell types.

4 Factoring in Randomness: Topic Modeling

4.1 Background

Much of the difficulty in interpreting scRNA-Seq data lies in its heterogeneity and the complex cell hierarchies that change over time. A challenge in our work consists of identifying and characterizing cell types from diverse transcriptomic information. We might consider framing the problem as one of *topic modeling*, an idea drawn from natural-

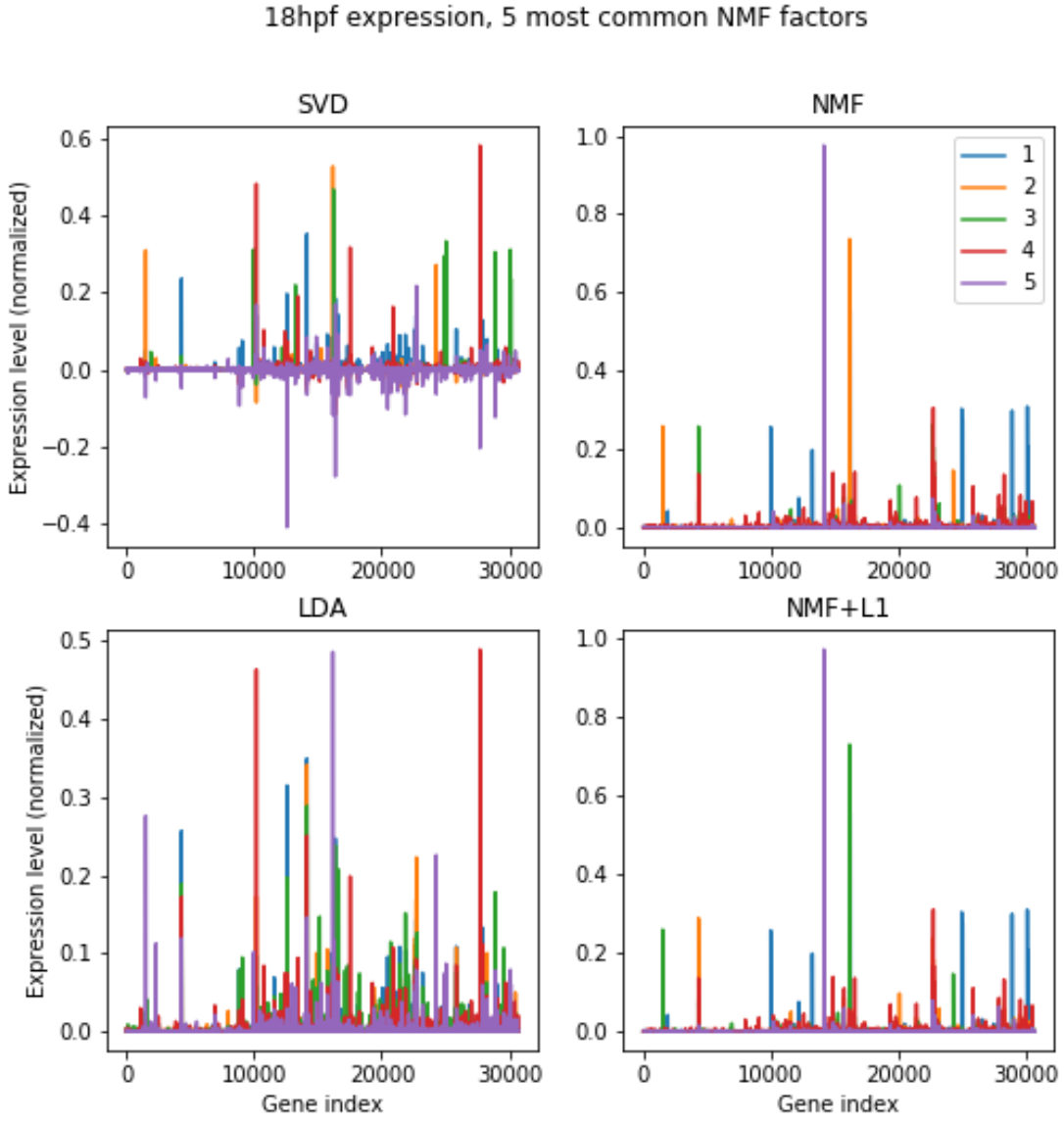


Figure 3: Expression profiles of the 5 most prominent archetypes. (a, top left) SVD at 18hpf. (b, top right) NMF at 18hpf. Notice that, unlike SVD, NMF produces strictly nonnegative factors, and encourages some degree of sparsity. (c, bottom left) LDA at 18hpf. Some factors appear to overlap significantly. (d, bottom right) NMF with ℓ_1 regularization at 18hpf. Factors are slightly more sparse, and many of the small coefficients in (b) are set to zero.

language processing and which motivates our usage of latent Dirichlet allocation (LDA). In this paradigm, we seek to classify genes by expression "topics," which can be loosely interpreted as a measure of developmental processes.[5] Latent Dirichlet allocation (LDA) is a generative probabilistic model originally conceived for the analysis of discrete data such as text corpora.[1] For the purposes of clarity, we will explain LDA using its traditional application to document analysis. As a hierarchical Bayesian model, LDA treats documents as collections of *topics* which in turn generate the probability distribution of *words* (Fig. 4) The distribution of topics k_t (there can be more than 1) generates the words of a document according to the Bayesian network shown in Fig. 1. LDA assumes the following process:

1. θ is drawn from a Dirichlet random with hyperparameter α.
2. The topics k_t are drawn from a multinomial with parameter by θ.

3. The word probabilities are parameterized by the topics k_t as well as another distribution parameterized by β describing the probabilities of words given a topic.

There are N words spread across C_t documents. Thus, learning in the LDA framework consists of determining the parameters for these distributions over topics and words.

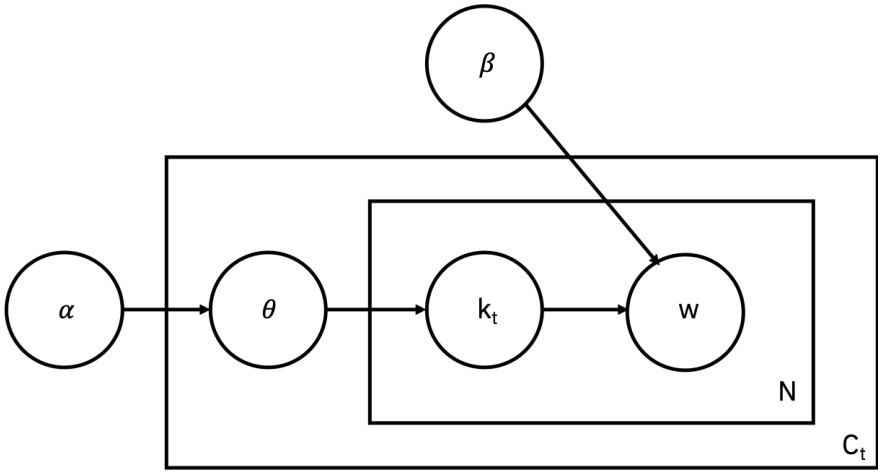


Figure 4: Schematic of Bayesian network for LDA. (Adapted from [1].)

For scRNA-seq data, the documents are the transcriptomes (cells) and the words, the genes expressions. Although the significance of "topics" in scRNA-seq data is not well-defined, we would expect that modeling with LDA should at least recover some information about known cell types. In particular, topics can be thought as latent developmental processes which generate transcriptomic profiles. Previous work using topic modeling has shown to robustly identify cell types, enhancers, and transcription factors in analysis of single-cell epigenomic data.[2] Furthermore, we expect that modeling topic changes over time will yield insight into the transcriptome dynamics throughout the developmental cycle of zebrafish.

4.2 Approach

For our analysis, we seek to fit each of the time steps with an LDA model with a given number of topics k which minimizes the *perplexity* (PP) on our test set, which roughly follows the likelihood of that data. In general, a lower perplexity indicates greater generalization performance.[1] Formally, the perplexity of a collection of C_t transcriptomes is defined as

$$PP(X^t) = \exp\left[-\frac{\sum_{j=1}^{C_t} \log p(x_j^{(t)})}{C_t}\right]. \tag{1}$$

The model is used directly from the Python library scikit-learn,[1] which implements the online variational Bayes algorithm to learn the parameters of the model. We let k_t to be a hyperparameter for each time step t and choose the smallest value which explains the data. For our tests, we have limit our search to k_t between 2 and 8, following to previous results which have suggested that too large of a value can easily lead to overfitting.[5] We determine the appropriate values of k_t by cross-validation with a 3-fold split at each time step, determining the value which maximized likelihood on the test set.

For all of our experiments, we performed 10 iterations of the EM-update in learning, which gives good convergence on our baseline tests (Fig. 6). At 4 hpf, it appears that LDA may benefit from $>$10 iterations for convergence, but we decided to use 10 to save on computation time.

We attempted various transformations to investigate their effects on LDA's performance. Generally LDA works well on handling high-dimensional data, but several studies have used preprocessing such as binarization and removal of low-variance genes.[2, 5] We first tested the effects of binarization using a 0 cutoff on the data (i.e. any points greater

[1] More information and documentation at scikit-learn.org.

that 0 are converted to 1 and otherwise 0). This transformation helps to reduce the effects of low sequencing-depth. We next applied feature selection by variance, as described in the previous section. For our LDA model, we tested two different cutoffs, 0.01 and 0.1 (i.e. we removed features which occurred in more than that number of genes). Since our data have been binarized, and thus follow a Bernoulli random variable X, the variances are

$$\text{Var}[X(p)] = p(1-p) \tag{2}$$

where $p = 0.01$ or $p = 0.1$.

4.3 Results

The best number of topics k_t for each time step and the estimated perplexity using the baseline data is shown in Table 3. For all of the timepoints, the best number of topics was either 6 or 7, indicating that our model had more generalizable performance with a larger number of topics, as expected. The perplexity also decreased monotonically through time, which appears consistent with the greater differentiation and thus more specific expression information. One might expect that the number of topics grows over time, tracking the increased complexity of the cell through development, though we do not observe this trend. The genes

However, looking at the top five words per distribution for the topics at a given time point shows a large redundancy and overlap between the topics. The genes for $t = 6$ and $k_6 = 6$ contain several genes repeatedly, including *hspa8*, *hspb6*, *cirbpa*, along with several others. These genes correspond to heat-shock proteins (*hspa8* and *hspb6*) and a cold-inducible RNA binding protein, respectively, and are both widely distributed among cells. These results are not surprising, given that LDA will learn higher weights for genes which appear frequently. However, this requires our analysis to remove or ignore this redundancy in some way.

The results after transformation demonstrate the effects of reducing dimensionality. For a cutoff of 1%, approximately half of the genes were removed, leaving anywhere from 13,000 to 18,000 genes, a large reduction from the original number of ∼30,000. For a cutoff of 10%, the reduction is even more drastic, leaving around 4,000 to 7,500 genes, or a reduction of over 75% in the dimensionality. As expected, both transformations have the benefit of reducing fitting time for LDA which is compoutaionlly beneficial especially if larger datasets are used. The minimum cross validation perplexities for both cutoffs are reported in Table 4. Interestingly, the perplexities for the baseline are in between the values determined for the two cutoffs tested. In general, with higher cutoffs and greater dimensionality reduction, the perplexity tends to decrease, reflecting the easier fitting process with smaller data sets. However, it is possible that with a cutoff of 1%, there are conflicting effects due to reduced dimensionality but also decreased ability to fully describe the data with a given number of topics.

Furthermore, analyzing the distribution of probabilities learned by LDA seems to indicate poorer performance after removing low-variance features (Fig. 5, 7, 8). It is interesting to note how much sparsity one loses after removal of low-variance features, which seems to contradict the intuition that less information would enable clearer modeling of developmental processes. However, inspection of the genes seems to suggest that variance-based feature elimination removed many important genes, such as the ones that occurred frequently in the baseline LDA, mentioned earlier. Fig. 5 shows the distribution of LDA probabilities over the approximately 30,000 genes for each timestep.

As noted before these plots indicate less sparse structure than with NMF or SVD, but one can still see clustering around each topic. For example topic number 7 (colored magenta in Fig. 5) appears to have similar distributions over the 8hpf and 10hpf time steps. This suggests a common developmental process which would generate the same process. It is more challenging to discover similar structural features over time in 7 and 8, but we note as an example that there is a similar clustering of topic 2 (orange) between 8hpf and 10hpf in 7, despite large levels of noise. These results are promising for LDA's ability to learn developmental processes. Further work is required in order to analyze whether specific genes in topics correspond to marker genes of cell types, and whether the probabilities are useful for predicting cell type.

5 Discussion

When we first evaluated this dataset we expected to find neat relationships between cells and their RNA transcriptomes that could be discovered with deep analysis. However, genetic data is not quite so simple. Our research demonstrates that there are many ways to approach analyzing and identifying genes and gene archetypes for further study.

Feature agglomeration produces an interesting jumping off point for identifying key developmental genes. Though many of the genes in isolated clusters have already been identified and thoroughly studied, several are practically unknown. Research in the biological effects of these genes is required to determine whether these are outliers or

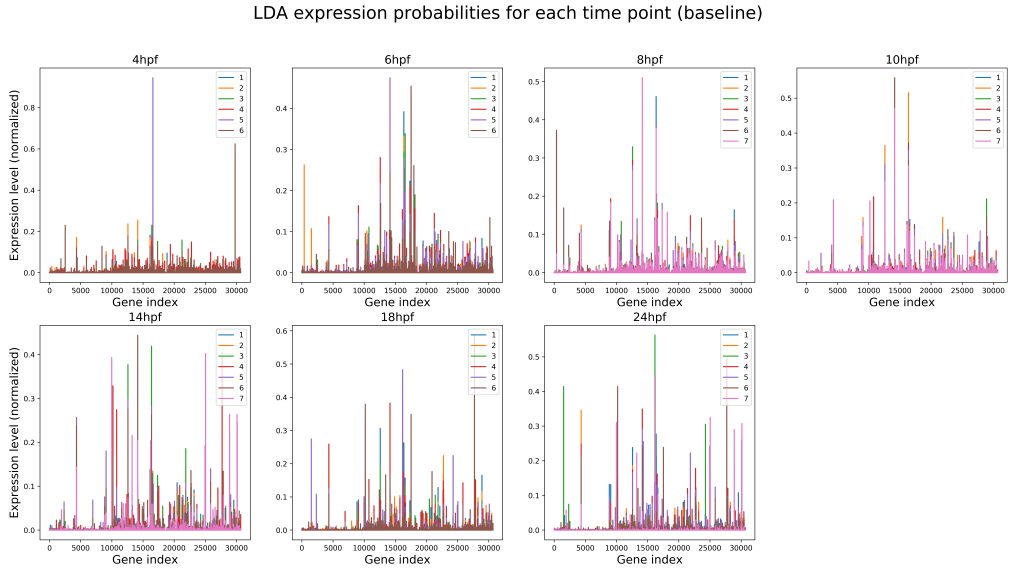


Figure 5: Distribution of LDA probabilities for each time step using unprocessed data.

novel identifications. In the future, we believe these types of methods could be valuable in isolating critical genes that determine biological and health outcomes.

Similarly, modern low-rank moels appear to be an effective method for modeling genetic archetypes. Two methods were particularly effective. First, nonnegative matrix factorization efficiently identified genetic archetypes that were sparse and nonnegative, allowing Latent Dirichlet Allocation also seems to be an appropriate and novel way of investigating transcriptomes. Like NMF, LDA naturally produces nonnegative archetypes that resemble the original data, and has the added benefit of a probabilistic interpretation, modeling the inherent randomness in the data. The technique, borrowed from natural language processing, has an analogous interpretation in genetic research. Genes can be thought of as words, archetypes as topics, and cells as documents. Furthermore, since the dictionary of genes is large, the data is naturally sparse like a text corpus. However, one limitation of LDA in comparison to NMF is that LDA does allow for hybrid cells—each cell is assigned to an archetype via some distribution. This leads to overlapping, non-sparse factors that represent the common genes expressed by all cells, which may not be desirable if we wish to identify the diverse genetic patterns that distinguish cells. This result suggests that future work should consider a model that combines the positive aspects of NMF and LDA, supporting both hybrid expression behavior and a probabilistic interpretation. Further investigation into LDA should test various data processing methods to encourage ease of interpretation, and analysis to determine cell type discovery may also present a useful challenge for these kinds of NLP techniques.

5.1 Literature Review

This project is an expansion on the work done by Wagner et. al. in 2018 [14]. Their work focused on identifying cell types using clustering methods and on the evolution of cell types over time using known genetic markers for zebrafish embryo cells. Our research focused more on identifying key individual genes and gene interactions in the development of the embryos. Recent research using RNA-seq for this purpose tends to focus on differentially expressed genes or DEGs between two sets of samples. One example from 2019 is an investigation of maize crop which isolated hundreds of genes that may contribute to development of seed embryos [15]. The model used by those researchers to identify important genes is complementary to this research, which examines primarily gene expressions at a single time step.

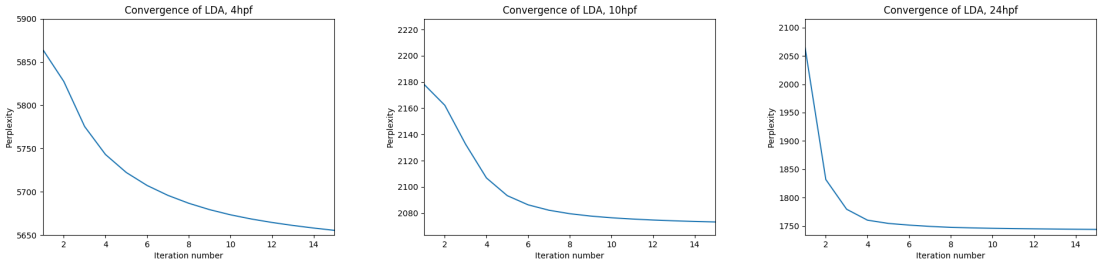
6 Code

The public repository containing our work can be found on GitHub at github.com/clkelley/cs221_final_project. The dataset can be accessed through the GEO portal at www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE112294.

Appendix

Table 2: Number of transcriptomes analyzed at each time point.

Timepoint (hpf)	# of Transcriptomes
4	4277
6	5692
8	3568
10	4280
14	4001
18	6962
24	34750



(a) (b) (c)

Figure 6: Convergence of LDA (measured by perplexity) at selected time points using an online variational Bayes algorithm. (a) 4hpf. (b) 10hpf. (c) 24hpf.

Table 3: Results of LDA baseline model fitting for each timepoint.

Timepoint (hpf)	4	6	8	10	14	18	24
Best number of topics (k_t)	6	6	7	7	7	6	7
Perplexity (PP_t)	5550	2870	2618	2009	2385	2257	1614

Table 4: Minimum perplexity for dimensionally-reduced data using LDA

Cutoff	0.1		0.01	
Timepoint (hpf)	Number of topics	Perplexity	Number of topics	Perplexity
4	5	3541	5	7613
6	4	5470	4	7934
8	3	4599	5	8642
10	6	4735	7	8429
14	7	4911	7	8957
18	7	5549	7	9588
24	8	4167	8	8043

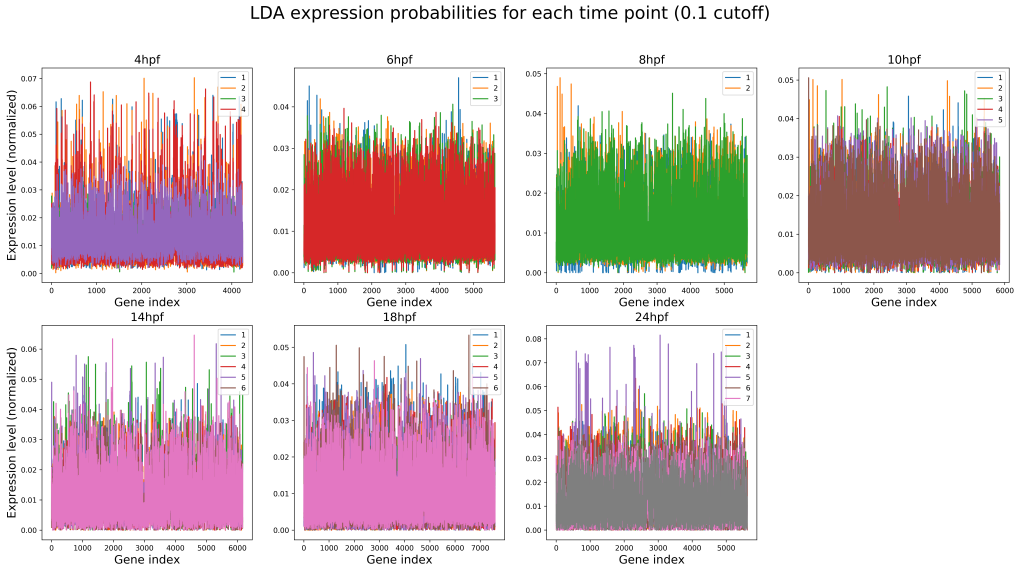


Figure 7: Distribution of LDA probabilities for each time step using cutoff of 10%

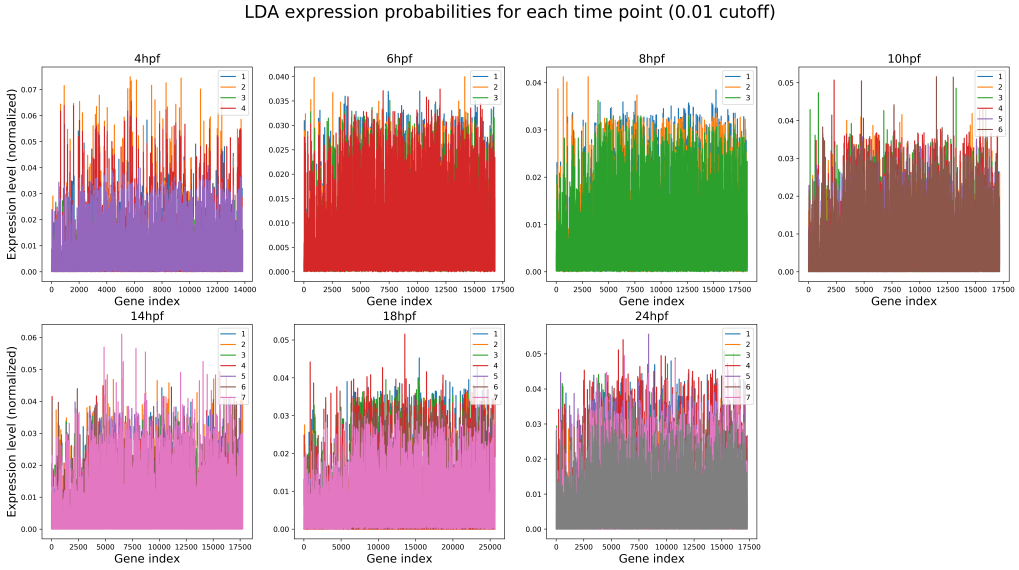


Figure 8: Distribution of LDA probabilities for each time step using cutoff of 1%

References

[1] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res. 3* (Mar. 2003), 993–1022.

[2] BRAVO GONZÁLEZ-BLAS, C., MINNOYE, L., PAPASOKRATI, D., AIBAR, S., HULSELMANS, G., CHRISTIAENS, V., DAVIE, K., WOUTERS, J., AND AERTS, S. cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nature Methods 16*, 5 (2019), 397–400.

[3] BUETTNER, F., NATARAJAN, K. N., CASALE, F. P., PROSERPIO, V., SCIALDONE, A., THEIS, F. J., TEICHMANN, S. A., MARIONI, J. C., AND STEGLE, O. Computational analysis of cell-to-cell heterogeneity in

single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology 33*, 2 (feb 2015), 155–160.

[4] DENG, Q., RAMSKOLD, D., REINIUS, B., SANDBERG, R., KRUTZIK, P. O., FINCK, R., BRUGGNER, R. V., MELAMED, R., TREJO, A., ORNATSKY, O. I., BALDERAS, R. S., PLEVRITIS, S. K., SACHS, K., PE'ER, D., TANNER, S. D., AND NOLAN, G. P. Single-Cell RNA-Seq Reveals Dynamic, Random Monoallelic Gene Expression in Mammalian Cells. *Science 343*, 6167 (jan 2014), 193–196.

[5] DUVERLE, D. A., YOTSUKURA, S., NOMURA, S., ABURATANI, H., AND TSUDA, K. CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinformatics 17*, 1 (dec 2016), 363.

[6] GAUJOUX, R., AND SEOIGHE, C. Semi-supervised nonnegative matrix factorization for gene expression deconvolution: a case study. *Infection, Genetics and Evolution 12*, 5 (2012), 913–921.

[7] KIM, P. M., AND TIDOR, B. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome research 13*, 7 (2003), 1706–1718.

[8] LEE, D. D., AND SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, 6755 (1999), 788.

[9] LI, Y., AND NGOM, A. Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data. In *2010 IEEE international conference on bioinformatics and biomedicine (BIBM)* (2010), IEEE, pp. 438–443.

[10] LU, Y., COHEN, I., ZHOU, X. S., AND TIAN, Q. Feature selection using principal feature analysis. In *Proceedings of the 15th ACM International Conference on Multimedia* (New York, NY, USA, 2007), MM '07, ACM, pp. 301–304.

[11] MOUSSA, M., AND MĂNDOIU, I. I. Single cell RNA-seq data clustering using TF-IDF based methods. *BMC Genomics 19*, S6 (aug 2018), 569.

[12] PAATERO, P., AND TAPPER, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics 5*, 2 (1994), 111–126.

[13] SHAHNAZ, F., BERRY, M. W., PAUCA, V. P., AND PLEMMONS, R. J. Document clustering using nonnegative matrix factorization. *Information Processing & Management 42*, 2 (2006), 373–386.

[14] WAGNER, D. E., WEINREB, C., COLLINS, Z. M., BRIGGS, J. A., MEGASON, S. G., AND KLEIN, A. M. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science (New York, N.Y.) 360*, 6392 (jun 2018), 981–987.

[15] ZHANG, X., HONG, M., WAN, H., LUO, L., YU, Z., AND GUO, R. Identification of key genes involved in embryo development and differential oil accumulation in two contrasting maize genotypes. *Genes 10*, 12 (2019), 993.