Recovering Cell Type Composition of Tissue Samples from Gene Expression Data

Adam Freilich and Alex Stein

May 9, 2016

1 Introduction

Tissue samples are made up of cells of different types. It is not possible, in general, to determine directly the fraction of a tissue sample which is of a specific cell type. We can, though, determine the gene expression levels of the sample by counting the number of reads which are present in the sample originating from each gene (and scaling by the length of gene to get RPKM data). Our goal is to recover the cell type decomposition from the gene expression data.

We will first discuss some assumptions we make about the data whose accuracy we were unable to determine due either to lack of resources, time or knowledge and address how one could determine their accuracy. We will then describe our methods for recovering the cell type decomposition given our assumptions. We will discuss our data as a specific example of our methodology and then the software and implementation details of our project.

2 Assumptions

Assumption #1: Primary Linear Algebraic Assumption

Our first assumption is that if we have a sample consisting of some fraction x of cells of one type (with average gene expression levels given by some vector v) and a $1-x$ fraction of cells with gene expression levels u then the gene expression levels given by the RPKM values for the whole sample will be given by $xv+(1-x)u$. This assumption seems reasonable if we consider cells as being a "bag of reads". As the expression levels are measured as fraction of total reads, we get the linear scaling described above.

Assumption #2: Consistency of Gene Expression Levels in a Given Cell Type Between Individuals

In order to take advantage of the linear algebraic structure we will need to assume that our data is in the linear span of some small set of vectors corresponding to cell types. We assume that in any red blood cell, in anyone's

body, gene expression levels are approximately equal. Equivalently, they are all approximately equal to one vector $v_{\text{red blood}}$. This assumption, specifically that these expression levels are consistent *between different individuals* may very well be erroneous and account for our data's distance from a low rank matrix.

Perhaps this assumption could be removed if we used a more general framework of tensor factorization which is a much more technically challenging feat.

Assumption #3: Additive Normal Error

In order to determine the rank of our data matrix, even though our data is noisy (there is some error introduced by the approximate nature of assumption #2 and by measurement error), we need to be able to determine the volume of the noise. In order to do that, we come up with an approximation based on the assumption that our noise has the form of a random matrix E in which each entry is drawn at random from a normal distribution. Then the data that we see is equal to $\hat{D} + E$ where $\hat{D}$ is our true noiseless data.

3 Methods

3.1 Expressing the Problem as a Matrix Factorization Problem

Let our data be represented as a matrix as follows: Let D be an $m \times n$ matrix where m is the number of genes and n is the number of samples. We let $D_{i,j}$ be the expression level of gene i in sample j. We refer to the jth column of this matrix as v_j and the jth sample.

We note that each sample is (up to a small error term) a linear combination of our cell types. That is $v_j = \sum_{k=0}^{K} c_{k,j} t_k$ where K is the total number of cell types and e_k represents the gene expression levels of cells of type k. We note that this universal set of vectors exists by our second assumption. This observation tells us that our data D is the product of two matrices $T \times C$ where T has K columns, the kth is equal to t_k, and C has entries $c_{k,j}$. Note that all of these matrices are non-negative. This justifies our assumption that D is (approximately) a low-rank matrix whose rank is equal to the number of cell types in our samples.

3.2 Dealing with the Error Term

Our data is actually equal to $D = \hat{D} + E$ where E is our error term which includes measurement error as well as variance in the gene expression level in cells of a given type due to individual differences. It is well known that measurement error is usually normally distributed. We make the additional assumption that other forms of error we will encounter are also normal and additive.

We observe that the "true factorization" of D as $T \times C$ would still have distance $\|E\|$ from our matrix D due to error. Anything that does better than that is therefore too good to be true. Any factorization which does not achieve

that distance from our data is not good enough as we know that there exists a factorization (namely $T \times C$) which achieves that distance.

If we know the error $\|E\|$, we can then try to factorize D into rank k matrices until we find a factorization (and a value for k) which is about distance $\|E\|$ from our data.

We present an approximation to the error $\|E\|$ based on our assumption that it is additive and normal. We make the observation that some of the rows of our data are close to constant but that because the error is random and normal it is unlikely that the variance of any (or more than a small fraction) of the rows is below the variance of the entries of E. We can then take the variance of the row with the least variance (or of a row in the first percentile of variances) to approximate the variance of E. In order to approximate $\|E\|$ we can then fill in its entries from a normal distribution centered about 0 and with variance taken by the procedure described above.

We note that, especially if we think most of the error comes from measurement error, this is a rather indirect way to measure error. A more sensible approach would be to run experiments with the machine perhaps by sequencing one sample multiple times and observing the resulting variance.

This variance was actually very low, meaning that there is likely very little error in our data. Performing the factorization with the above numbers tells us that the rank of our data is actually very high (K is very large) or that it is very far from being low rank. It is rather likely that we have made a mistaken assumption up to this point.

3.3 Recovering Cell Type Decomposition from Matrix Factorization

Having determined the rank of the data, we can now factor the matrix into $T \times C$ using Nonnegative Matrix Factorization (NMF). We assume that we can recover T (which we can view as a set of basis elements) up to permutation and scalar multiplication. This assumption is not known to be true in general but we may be able to assume it here because we have some additional structure (for example, that the factorization is non-negative). One further direction for this project is to test this assumption with simulated data.

We have established so far how we can use NMF and gene expression levels to retrieve the gene expression levels of the different cell types present in our samples. We now describe how to determine which gene expression levels correspond to which cell types. The main piece of information which we use to determine which basis element corresponds to which cell type is the sample types it appears in most often. For example, if a certain basis element forms, on average, %90 of each blood sample, it is likely to be a blood cell.

We note that given a factorization of our data matrix we can determine the fraction of a sample that comes from a given basis element. If a sample $v_j = \sum_{k=0}^{K} c_{k,j} t_k$ then t_k accounts for $\frac{c_{k,j}|t_k|}{|v_j|}$ where the norm is the 1-norm (the

sum of the entries). Note that the sum of these fractions over all basis elements would be 1 in a noiseless scenario.

Therefore, we factor samples of a few types, and make a table which has the t_ks given by our factorization and the percentages with which they appear in each of the sample types. Such a table appears in our next section.

4 Implementation

All of this data analysis was implemented in python using `numpy` to store the data and `plotly` to export the outputted histograms. Because the data retrieved from Gtex was very large (50000 by 8000) parsing through the data each time we wanted to run the analysis was extremely costly in terms of run time. We solved this problem by creating a class called `sample_parser`.

This package takes in some data matrix (be it the full matrix or some reduced version), the mapping of sample key to sample type, whether this is the full matrix or a reduced version and lastly the name of the tissue being analyzed (you can pass in more than one type of tissue). The package preprocesses the data, and stores the data. It has methods that allow to cut down any columns of samples that are not desired and that remove rows of genes that are expressed less than .1, 70% of the time. This data can be saved for later use.

The package also has a method to get the minimum variance among all the rows which is used to calculate the random error matrix.

The main method reads the data for the correct matrices (and if they do not exist yet for the desired tissue, they create them). After finding the minimum variance we create the random normally distributed matrix to compare with our factorized matrices. This is all done using `numpy` and `nimfa`. Lastly, to create the histogram data, for each tissue, for each basis element we get $\frac{\sum b_i \cdot coef_{i,j}}{\sum data_j}$.

5 Data

In order to be able to recover the cell type information, we will want to include samples of a few different tissue types (here we chose whole blood, lung, thyroid, ovary and tibial nerve samples). We have 1437 of these samples. Additionally, so that our calculations could run in a feasible amount of time we removed all genes which aren't expressed at .1 RPKM in over %70 of the samples.

5.1 How well is our matrix approximated by a rank k matrix?

As in the process described above we see how well our matrix can be approximated by a factorization into rank k matrices and choose the matrix which is about as close as we expect our error to be. That is for each k we factor $D \approx T_k \times C_k$ and we find the last k for which $\|D - T_k \times C_k\| > \|E\|$. Below is a table of values k on the left and $\|D - T_c \times C_k\|$ on the right.

We note that these errors are rather large and that this fact should convince us that one of our assumptions has failed us.

k	Error
1	1404042
2	984676
3	739052
5	540274
10	470079
20	434870
30	424854
40	419829

5.2 What do we expect is the variance of the error?

Above we described a few methods for determining the variance of our random noise. We list the methods on the left, the variances in the middle and what error values they give on the right.

Estimation Method	Variance	$\|E\|$
Minimum Variance Row	0.00459192736781	354.390872697
Maximum Variance Row	27281331962.9	863715594.431
Mean Variance Row	2234875.34351	7818037.43075
Median Variance Row	3.42909183632	9684.26615869
1st Percentile Variance Row	0.0314070730976	926.607150768
85th Percentile Variance Row	110.627906798	55013.8717282
97th Percentile Variance Row	4968.63189406	368674.443681
98th Percentile Variance Row	10743.8945676	541969.533658
99th Percentile Variance Row	47680.7667504	1141680.37572

The reasonable approaches to approximating the variance (among them minimum variance of a row, 1st percentile variance and even median variance) all give us error values small enough to convince us that our matrix isn't low rank (it is of rank $\gg 40$).

In order to demonstrate our techniques we take the variance to be in accordance with the last row of the above table giving us that our matrix is of rank 4.

5.3 Which cell types are present in which tissue types?

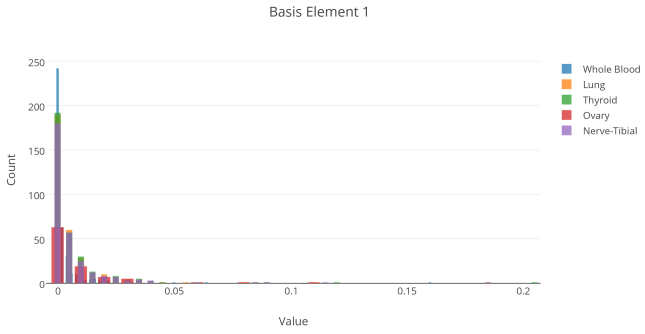
	Whole Blood	Lung	Thyroid	Ovary	Tibial Nerve
t_1	0.0030	0.0065	0.0067	0.0080	0.0067
t_2	0.0009	0.0021	0.0022	0.0029	0.0022
t_3	0.4662	0.8495	0.8813	0.8675	0.8946
t_4	0.0047	0.0108	0.0113	0.0112	0.0114

The form of this table would allow us to observe that "cell type" t_3 forms most of the lung, thyroid, ovary and tibial nerve samples, but a little less than half of the blood samples. If we knew of a cell that makes up most of those four types but not of blood we would be able to make an identification of what cell type t_3 corresponded to.

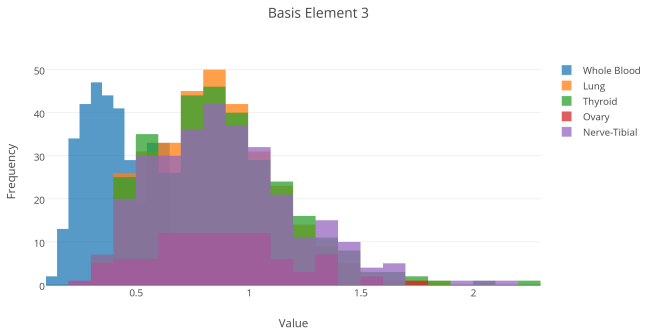
5.4 Distribution of Cell Types in Samples:

Here we show how the fraction of samples consisting of cell type t_1 varies between the different individuals. This is a histogram of blood samples sorted by percentage of cell type t_1 on the bottom.

Percentage of t_1 in the samples by sample type:



Percentage of t_3 in the samples by sample type:



We note that some of the samples are listed as being over 100 percent of type basis element t_3. This is to be expected as first, as we faced an issue with our assumptions this factorization is incorrect and as matrix factorization is can only approximate our matrix, some of the coefficients in out matrix C are likely off by a non-negligible multiplicative factor.

6 Conclusion

We have described above a linear algebraic technique for recovering the cell type composition of tissue samples from gene expression data. We have made a few assumptions and based on our data we expect that at least one of them (perhaps specifically assumption #2) is false.