Building and using a hypothesis recommendation system to infer the tissues of origin for blood samples

Mira Moufarrej – miramou@stanford.edu
Departments of Bioengineering and Computer Science
Stanford University, Stanford, CA

Abstract

Often, in biology, discovery experiments leave researchers with the open question of what hypotheses and leads to pursue next. In one class of experiment growing in popularity because of its non-invasive nature, specific RNA molecules present in blood correlate with disease risk - often for diseases with no treatment. Toward developing such treatments, it would be helpful to generate hypotheses about where these predictive molecules identified in blood come from. Here, we present and compare two recommendation methods based on nearest neighbors and GraphSage to recommend candidate tissues of origin and quantify performance using observed single tissue of origin samples.

1. Introduction

Non-invasive diagnostics that use molecules present in blood (cell-free RNA, cfRNA) to predict disease (e.g., cancer and prenatal risks) have recently risen in popularity due to their ease of use and rich information content; however, interpreting such predictions remains unclear. Because treatments still do not exist for many of the predicted risks, physicians are left with an interesting problem - a potential diagnostic without any viable treatment options. One way to develop specific treatments might be to infer which tissue a given set of cfRNA molecules predictive of the disease originated from. However, because cfRNA is measured in blood, it represents RNA from many cells in many organs that signal to one another or died and released their contents. Also, most RNA molecules are not tissue specific alone, rather the co-occurrence of several genes of specific subtypes may prove more informative [19, 12]. So, how can we infer which tissues certain cfRNA molecules like those predictive of disease originated from?

Here, we propose tackling the latter as a tissue recommendation problem - similar to product recommendation systems. Given a sample with certain measured genetic features and a tissue label, create a highly descriptive embedding such that any samples from the same tissue are close in the embedding space and consequently "recommended" as similar. Importantly, recommendation poses the optimal learning problem since we want any embedding method to have the flexibility to propose multiple tissues as similar (in the case of a tissue mixture like in cfRNA) and to estimate the fractional contribution of each tissue. Because multi-class classification aims to predict one single best class to fit the data, to predict mixtures as posed here, we would have to either enumerate an exhaustive list of tissue label mixtures or leverage the final layer's softmax vector. Neither of these solutions is ideal. Similarly, link prediction poses a less than optimal goal as well since it only suggests the existence of an edge without estimating the fractional contribution of that edge and therefore tissue.

Recommendation as a task allows us to both estimate tissue mixtures since a good embedding of a mixture should lie between the tissues it represents and similarly estimate fractional contribution based on the same principle. We will compare two recommendation techniques that apply cosine similarity and GraphSage [21] using a hold-out set composed of samples from a specific tissue. To leverage graphical methods, we model the relationship between genetic features and samples as a bi-partite graph where distinct feature types (e.g., splice junction ratios) and samples each form a graph part. Edges connect features that were measured in a given tissue with edge vectors quantifying the measurement. Feature node vectors consist of node weights describing the tissue specificity of a given feature. Sample node vectors are labeled using the tissue of origin during training. To train the embedding method, we will use a subset of the 17,631 samples of human tissue available from the Gene Tissue Expression (GTEx) consortium [7] and Encyclopedia of DNA Elements (ENCODE) consortium [6, 8]. We then quantify the performance of each recommendation system by comparing the estimated fraction of recommended labels to the true fractions of labels. For true cfRNA samples with no label, we can then tie these results back to biology to qualitatively assess tissue contributions hypotheses in the context of a specific disease.

2. Related Work

In recommender systems, we aim to recommend similar nodes based on the content of labeled nodes used during training and for some systems, their neighbors as well. This presents several challenges especially when considering the domain-specific task presented above. In this section, we will explore present work to address each of the aforementioned challenges: (1) Manipulating biological data to form networks, (2) The advances and limitations of a unique, promising form of biological data, cell-free RNA, specifically, here, as related to prenatal care and (3) The current state of recommendation based systems and necessary requirements to apply such methods to this problem.

2.1. Building tissue-specific networks

Given a matrix that describes the connection between certain tissue samples and molecular information, we would like to build a graph that informatively describes feature co-occurrence and reveals some hidden biology. To this end, Saha et al. built built two types of gene co-expression networks for 26 human tissues [18]. First, to understand differences in regulation between tissues, they built networks on a per tissue basis using graphical lasso. Each network contained two nodes types - genes (e.g., Gene A vs Gene B) and gene isoforms (gene subtype) (e.g., Gene A Subtype 1 vs Gene A Subtype 2) with edges connecting node types that co-occur. They then described node roles in such networks, revealing that hub nodes corresponded to key cellular regulators. These were distinct on a per tissue basis with tissues of similar composition (e.g. fatty tissues) sharing key hub nodes. Saha and colleagues further built tissue specific networks using only gene abundance data and applying Bayesian biclustering to identify network edges unique to single tissues. They note that they only successfully built tissue specific networks for 26 of the 50 tissues for which they have data. They reported that networks provide an improved understanding of regulatory mechanisms across tissues citing specific examples on a per tissue basis.

Both network types clearly captured some rich information; however, focusing on tissue-specific networks, the authors chose to build a network using gene abundance data alone to describe edges - stating that they ignored relative isoform abundance for simplicity. Other related work (although not focused on networks) indicates that gene abundance does not correlate well with tissue specificity [20, 12].

Using the same dataset leveraged by Saha et al., Li, Knowles, and colleagues describe that instead, using relative isoform abundance not only parses data by tissue type but also that such parsing is conserved across species. Because they observe the same isoforms for a given tissue in multiple species, it suggests that certain isoforms may be necessary for tissue function, and therefore would be expected to correlate well with tissue specificity. Further, Uhlen et al. found by combining the data used in Saha et al. with similar tissue datasets from other groups, that most genes are detected in all tissues and therefore few (on the order of 10s out of 10^4 genes) can be called truly tissue specific [19].

2.2. Advances and limitations using cfRNA data

In the 1990s, scientists from multiple groups identified pregnancy-related nucleic acids in maternal circulation, termed cell-free DNA (cfDNA) [13] and RNA (cfRNA) [17]. As the placenta reshapes itself to accommodate a growing fetus, these molecules specific to the placenta and fetus appear in the mother's blood in appreciable amounts that change over gestation, providing a window into pregnancy development and fetal health. Using cfRNA present in maternal circulation, previous work has shown that transcript abundance changed in maternal circulation as pregnancy progresses [11, 15]. Ngo et al. then leveraged cfRNA to predict two key measures of prenatal health - time to delivery and risk of preterm delivery, the leading cause of infant death under the age of 5, which to date has no good predictors [14].

Taken together, these papers show the power of using 1 milliliter or less of blood to diagnose multiple prenatal conditions. Focusing on the work by Ngo et al., they interestingly suggest which molecules correlate with prenatal risks; however, they make no comment on possible biological causes behind such correlation. They also point out that many of the molecules identified as predictive of prenatal risks were implicated via signaling (i.e., cellular communication) and therefore used by many tissues. Critically, developing treatments for such prenatal risks, which have no effective options at present, relies on understanding where these molecules that appear in circulation come from. So, how can we move beyond prediction and toward interpreting these results and creating treatments?

2.3. Recommendation system Methods

Recommendation methods aim to capture content from a given node and its neighbors (often using weighted importance sampling) to identify node embeddings that are similar to the query node. These methods all work to embed nodes such that nodes of similar type are close and others are far. As such, the information used during embedding varies like neighboring node information - expanding to encapsulate information from further neighbors - h-hops away - and weighting certain edges more than others (e.g., those with low degree).

These methods have trade-offs - for instance, using local information only can be less computationally intensive; however, may miss long range yet important connections. Further, because all embedding techniques focus on maintaining certain information from high-dimensional space

while discarding other details, such methods should be applied keeping in mind the problem's context and what type of relational information we wish to preserve. Finally, shallow embedding methods that describe simple lookup tables do not work here since they are transductive and do not lend themselves well to predicting the label for unknown samples like cfRNA. Here, we use two recommendation methods. The first is a simple baseline based on taking the cosine similarity between all node pairs and then recommending for a test node the k nodes to which it is most similar (K top similarity scores). The second, GraphSAGE [10], applies principles of neural networks to graphical sampling and prediction.

Since TransE [5] and content embedding [1, 2] are shallow encoders, they do not work here since the labels for cfRNA samples are unknown and consequently l for TransE and related sample labels for content embedding methods are missing.

3. Data

We will be working with RNA-sequencing data collected by Stephen Quake's group at Stanford (cfRNA data), the Gene Tissue Expression (GTEx) consortium [7], and Encyclopedia of DNA Elements (ENCODE) consortium [6, 8].

The cfRNA sample set consists of 480 samples collected from healthy pregnant women with five to seven samples per woman corresponding to the 29th-37th week of pregnancy prior to delivery. Importantly, because we will be using this data as part of our test set, we must consider the unique features of pregnancy like additional organs present during pregnancy (e.g., the placenta) and contributions from the fetus - both of which have been quantified previously using cfRNA in the third trimester [14].

To build a bi-partite tissue graph and understand connections between tissues and genetic features, we will use data from the GTEx consortium and ENCODE. The GTEx consortium collected 17,382 RNA-sequencing on tissue samples from 980 people (653 men, 327 women) who consented to donate their organs at death. With multiple tissue samples per organ, we can obtain good embeddings for each tissue type. However, notably missing from the data set are placental samples. Additionally, the data set skews male, which may be a problem when trying to estimate tissue contributions during pregnancy. Further, GTEx represents data collected from adults aged 20-80 does not allow us to estimate fetal tissue contributions. Data from ENCODE can supplement that from GTEx with 249 samples collected during various stages of human embryonic development and the placenta itself. This will allow us to incorporate needed placental and fetal samples that may help further explain our test cfRNA data, collected from pregnant women.

All together, we have data from 18,111 samples - 17,382 GTEx samples, 249 ENCODE samples, and 480 cfRNA samples. Importantly, some of this data exists as either raw sequencing reads (fastq files) for some databases (ENCODE, Quake Group), and consequently requires additional heavy preprocessing prior to use. Additionally, once all data has been converted to junction count tables, we apply additional normalization to account for technical noise.

3.1. Data pre-processing

3.1.1 Pre-processing and aligning reads

Reads were aligned using two-pass STAR (Spliced transcripts alignment to a reference) alignment [9]. In two-pass alignment, each read file is passed through the aligner once to generate a sample specific splice junction file, and then is passed a second time along with the sample-specific splice junction file to definitively align each sample. STAR alignment presents the fastest accurate alignment tool for RNA sequencing available today. Prior to alignment, read adapter sequences are trimmed with trimmomatic [4]. After sequencing, aligned reads are sorted, indexed, and deduplicated using Picard [3].

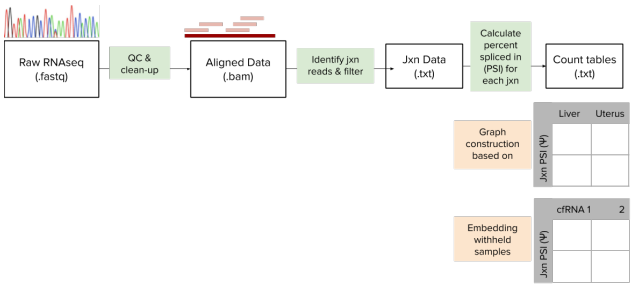

Figure 1. Overview of sample pre-processing pipeline

3.1.2 Estimating splice junction counts

Junction were estimated using aligned reads. Sample files were then merged into a matrix where every row is a feature and every column is a sample as outlined in Fig 1.

4. Methods

4.1. Building the graph

We propose a bi-partite knowledge graph, an example of which is shown in Figure 2. Here, genetic features of a specific type represent one part of the graph. In Fig. 2, only 1 genetic feature part is shown; however, one can extend the same structure to incorporate information sharing across multiple types of genetic information.

Edges connect features with the samples where they were observed. Each edge is associated with a feature vector describing normalized quantitative measurements associated with the genetic feature measured (see Section 4.2

for further details). Further, because in both the GTEx and ENCODE datasets, there typically exists more than 1 sample per tissue, we can split the data into training, validation, and test sets (green diamond nodes). These labeled sample nodes will allow us to compare the performance of distinct recommendation methods, in contrast to unknown cfRNA samples (orange nodes) for which ground truth (about tissue origin) is unknown.

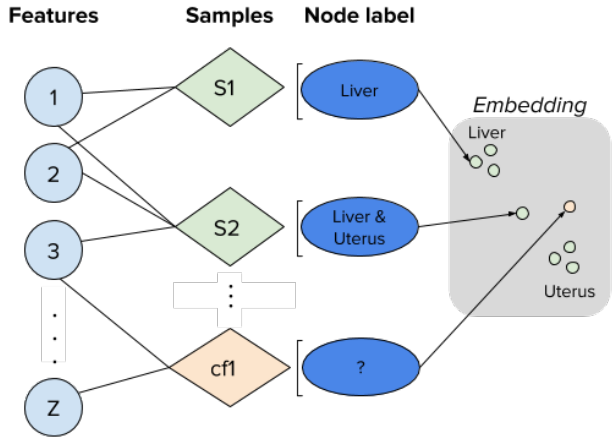

Figure 2. Example bi-partite graph where information is shared (solid edges) between samples and features. Each edge will have an associated feature vector and edge weight. Graphical structure and edge features/weights will be leveraged to recommend new tissue labels for query samples (orange). Far right shows example ideal embedding for a liver sample (S1), a known mixture (S2), and an unknown sample (cf1) with some features similar to the uterus.

4.2. Building edge feature vectors

For one feature type (Fig. 1, far right), we estimate percent spliced in (PSI, ψ for a given splice junction composed of a donor site, D, and an acceptor site, A. We calculate the following as described in [16], which forms the edge feature vector:

1. The contribution of reads from a specific pair, $C(D, A)$ relative to that of all pairs that use D:

$$\psi_D = \frac{C(D, A)}{\sum_{A' \in (D, A'} C(D, A')}$$

2. The contribution of the same pair relative to all pairs that use A,

$$\psi_A = \frac{C(D, A)}{\sum_{D' \in (D', A)} C(D', A)}$$

4.3. Examining graph properties

To inform the design and help explain the results of each recommendation method, we examine the following graph properties. We confirm the graph is bipartite, count the number of connected components, calculate the size of the giant component, and find the average connectivity of all nodes using established methods implemented in networkx. Finally, we take the sample nodes projection of the graph and look at whether samples from the same tissue cluster in the projection - that is are the raw edge vectors between samples of similar tissues more similar than those of different tissues. Note that here, we have not done any edge pruning, which may prove crucial. We calculate similarity between sample nodes by looking at the cosine similarity of the edge feature vectors defined by the common neighbors for two samples. Formally, for a node u and v, we weight $(w_{(u,v)})$ their connection in projected space by the cosine similarity between the edge weights, e, defined by u and a subset of its neighbors, N, and v and the same subset of its features (those in common with u).

$$w_{(u,v)} = \frac{e(u, (N(u) \cap N(v))) \cdot e(v, (N(u) \cap N(v)))}{||e(u, (N(u) \cap N(v)))||_2 * ||e(v, (N(u) \cap N(v)))||_2}$$

4.4. Recommendation methods

We will compare two methods - a simple baseline based on taking the cosine similarity between all node pairs and then recommending for a test node the k nodes to which it is most similar (k best cosine similarities) and GraphSAGE [10]. Here, we describe important details of each. We would like the ideal embedding method in addition to providing useful embeddings to be robust to missing features since for cfRNA measurement noise can be high and interpretable because mapping specific features to recommendations allows for stronger hypothesis generation and informs follow-up experiments.

4.4.1 Cosine similarity

As a baseline, we propose a very simple recommender system. To recommend similar samples, we can calculate cosine similarity between the vectors that describe all labeled training examples and validation or test examples. For each validation or test sample, we then choose the k top cosine similarities and recommend the samples they are associated with as labels for the query. We can then evaluate this method using the hit-rate (described in section 4.5).

Notably, because the time to calculate similarity between all sample pairs grows quadratically with the number of samples, the calculation soon becomes inefficient. To address this, for each tissue, we take a random sample of the trained examples. We then calculate pairwise cosine similarity between this labeled sample subset and validation or test queries.

4.4.2 GraphSage

We wish to learn informative embeddings for sample nodes such that samples from the same tissue are similar (positive examples) and samples from different tissues are dissimilar (negative examples). To do so, we train supervised GraphSage using a cosine embedding loss function (see below) where we attempt to minimize and maximize the distance between related and unrelated pairs defined as samples from the same tissue context versus different contexts, respectively.

Here, the loss, $\mathcal{L}$, for a pair of embeddings, (x_1, x_2), is defined as the distance from maximum similarity if the two examples are related ($y = 1$) and the max of minimum similarity (0) and the cosine similarity minus a defined margin, Δ if the two examples are unrelated ($y = -1$).

$$\mathcal{L}(x_1, x_2, y) = \begin{cases} 1 - cos(x_1, x_2) & \text{if } y = 1 \\ \max(0, cos(x_1, x_2) - \Delta) & \text{if } y = -1 \end{cases}$$

Further, to include information about which samples (as opposed to gene features) neighbor a query node, the network consists of 2 layers aggregating information from the query node and its neighbors using a skip connection followed by a ReLU transform. For improved training and ease of embedding interpretation, embeddings are normalized using the L2-norm. Finally, we tune the final embedding dimension and measure performance offline on a labeled validation set.

We evaluate performance both for tuning and testing using the hit-rate (described in section 4.5).

4.5. Evaluation

The recommendation methods will be compared using the hit rate, a common metric for recommendation system evaluation also applied by Ying et al. to evaluate PinSage [21]. For each positive example pair, (q, i) where q and i are samples from the same tissue, we embed q and select its K nearest neighbors as recommendations. The hit rate is then the fraction of queries, q, where i appeared in the top K recommendations for the test sample. We can also visually inspect embeddings using graphical projects like tSNE (similar to right side of Fig. 2) to further confirm that embeddings are sound.

5. Results

5.1. Graph building

Graph building proved especially challenging and more time-consuming than originally expected thereby limiting the number of recommendation methods we could try. Specifically extracting features from the dense matrices that typically specify genetic features to generate graph features resulted in frequent memory issues that were very hard to debug. As the authors later discovered, pandas is especially memory inefficient for large dataframes (e.g. a 12GB dataframe quickly consumes over 128GB RAM using certain pandas methods). This proved especially difficult to debug since code that worked on a reasonably sized development table (e.g. 2GB instead of 12GB for debugging) would then promptly fail on the larger dataframes we hoped to use to generate the graph. If this were repeated, the authors might have leaned more on data.table or other classes that implement procedures from databases to control memory usage and efficiently perform tabular calculations.

For this work, to address the memory issues, we first subsetted the largest dataframe, the GTEx data, to include 1447 samples instead of 17,382 in a class balanced fashion. We then also implemented several workarounds to arrive at a data object compatible with the specifications for a custom `torch_geometric.data` dataset. We split all data in a class balanced fashion (sampling from each class individually) into training, validation and test sets consisting of 70%, 15%, and 15% of the all labeled samples respectively.

We also used the data object we wrote to describe the dataset graphically to visualize the bipartite graph and obtain graph statistics using networkx.

5.2. Graph statistics

Ultimately, we were unable to generate a graph in time; however, plan to by the poster session, having debugged most of the memory issues (so we think) by now.

Graph statistics (the ones we know) are summarized in the table below. Overall, the number of features (junctions) far exceeds the number of samples. Given more time, we would prune the graph to remove junctions only observed in a small fraction (e.g. 0.1) of all samples since these are likely noisy, uninformative measurements that pollute the graph (Fig. 3). It should be noted that we have a script ready to calculate the rest of these that has been tested on a smaller development set of the data.

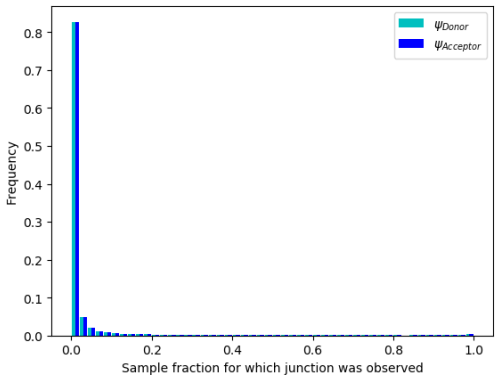

Figure 3. Statistics for usage frequency of junctions observed. Most junctions are observed less than in 0.05 of samples whereas the rest ($\sim 300,000$ are observed in more samples.

Feature	Value
N nodes	4,163,093
N edges	
Average node degree	
N connected components	
Size of giant component	

Table 1. Description of bi-partite graph statistics

Finally, for a subset of the data across all data sources, we would have plotted the sample projection and colored by distinct tissue labels. Since we were unable to generate the full graph, an example plot of this type generated using a subset of the ENCODE data and the same script that would have generated feature statistics (Table 1) is shown below. For this subsample, we do not observe clustering by tissue type; however, this may be because the number of features $f > n$ where n is the number of samples and as a result, we only observe noise. By pruning low frequency features (Fig. 3), we might observe better clustering (although likely far from desirable) in this plot. The plot represents a good sanity check that there exists some order in the data that embedding methods might pick up on.

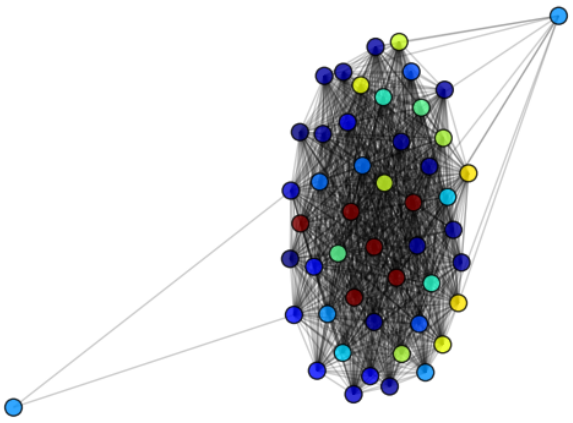

Figure 4. Example projection on sample part for subset of ENCODE samples colored by tissue

5.3. Cosine similarity performance

Calculating cosine similarity efficiently across even a subsample of all training examples proved challenging. Further, although this method presents a basic, reasonable first-pass/baseline, it does not allow for flexibility. For instance, it does not informatively incorporate new information from new samples since every new sample results in a new pairwise comparison. Notably, it is unlikely that a mixture example would be that similar to any of the tissues it is composed of and as a result, this method may not predict the correct labels as expected.

Below, we describe the results of calculating cosine similarity for a subset of data since we were unable to generate the full graph in time. Importantly, the rows are ordered by tissue such that samples from the same tissue would appear next to each other. For this small subset of features and samples, it does not appear as though raw features alone correlate with one another since we do not see any areas where the cosine similarity is particularly high. Instead, the plot appears random. Perhaps other junction counts (beside the ones plotted) better correlate with tissue.

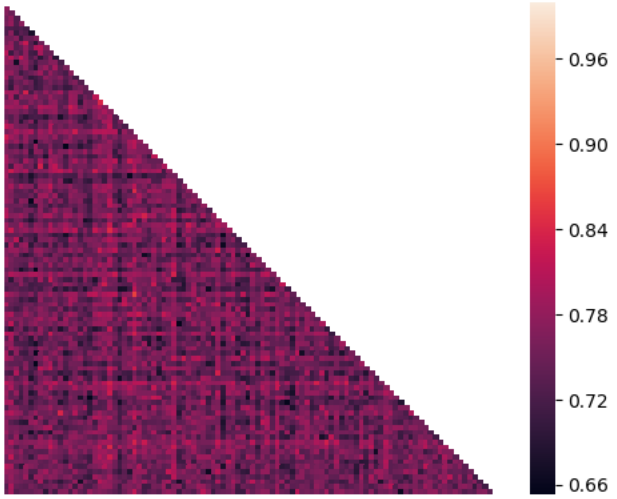

Figure 5. Example heatmap on sample data subset where rows and columns are ordered by tissue. Notice the lack of structure across the graph (no obvious areas of the same color) indicating that at least for this small subset of features, there is no agreement between samples of the same tissue.

5.4. GraphSage training and performance

With more time, we were prepared to train GraphSage using 1 GPU and had set up code to train GraphSage over 500 epochs for the whole graph. Below we describe what we would have done (and hopefully will do by the poster session). We would have tuned the embedding dimension size in a grid like fashion across several orders of magnitude, comparing performance on the validation set for embedding dimension sizes of 10, 50, 100, and 300. Using this coarse grain grid search, we sought to choose the minimal embedding dimension such that it would be expressive enough to describe all data classes, but not overfit the training data. This was particularly a concern given the number of features relative to the number of samples. Further, we wanted to limit the size of the embedding dimension since we would also like to predict mixtures for which if embeddings are too spaced apart across multiple dimensions, the curse of dimensionality may become an issue.

Finally, we would have compared this experiment to co-

Embedding dimension size	Hit-rank (k = 10)
10	
50	
100	
300	

Table 2. The effect of embedding dimension size on performance

sine similarity performance using the hit-rank metric described in methods and commented on the superior method. We hypothesize that GraphSage, which readily incorporates neighboring information from all neighbors (vs cosine similarity which only looks at pairs of samples), would have significantly outperformed cosine similarity. Further, although GraphSage is an expensive method to train, as an inductive model, it scales better to new data like new test samples and new tissue samples to incorporate in embedding space. This is particularly appealing given that there are many new biological atlases currently under collection and sample processing which could later be incorporated.

Method	Hit-rank (k = 10)
Cosine similarity	
GraphSage	

Table 3. Final comparison of embedding methods

6. Discussion & Conclusions

Extracting features to form a well-structured graph proved extremely difficult for unforeseen reasons like pandas' unwieldy memory consumption. Once the graphs are built (hopefully later tonight), we will implement the methods described and report by the poster session (hopefully). Hopefully the above makes clear our thought process and what would have been described and interpreted given results. Below, we offer our hypotheses about what we will observe as a sample interpretation of the results.

We believe that GraphSage will outperform the basic baseline. We however also think that the embeddings may not describe mixtures well since the embedding dimension size that best works for single label prediction (e.g. one tissue) may be larger than the one that works best for mixture embedding. Mixture embedding relies on the embedding of a mixture appearing proportionally in between the embeddings of two single source samples, and this becomes harder as the size of embedding dimension increases due to the curse of dimensionality. Finally, it should be noted that we have written all the code to complete this project, and are in the process of debugging graph generation.

Overall, we believe this to be a potentially promising avenue for hypothesis generation and interpretation of cfRNA results. This is particularly exciting since cfRNA presents a non-invasive method by which we can study many diseases in humans and then combined with clever network methods, we may be able to generate hypotheses that go beyond correlation to test further using an animal model or cell line.

7. Further work

We plan to further improve on the two recommendation systems described above hopefully before the poster session. We also want to implement the systems for graphs that contain different genetic features (e.g. gene counts) and compare the results to the above using the same evaluation metrics.

We also would like to see if there is improved performance if we make the graph tri-partite and include both gene counts and junction counts as sources of information or if the performance saturates. Further, due to time constraints, we were not able to explicitly test for performance on mixture samples (similar to cfRNA) above. To do so, we could simulate tissue mixtures by subsampling and combine feature measurements from labeled samples (e.g. GTEx and ENCODE) in known ratios. We could evaluate performance on these mixture samples in a similar but not identical way to that described above. Because hit-rate only accounts for one label, we would instead want to calculate the fraction of queries for which the recommended k samples contained samples that reflected each tissue in the mixture in appropriate relative proportions.

References

[1] Applying deep learning to related pins - the graph - medium.

[2] Listing embeddings in search ranking - airbnb engineering & data science - medium.

[3] Picard tools - by broad institute.

[4] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, aug 2014.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 2013.

[6] E. P. Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, sep 2012.

[7] G. Consortium. The genotype-tissue expression (GTEx) project. *Nature Genetics*, 45(6):580–585, jun 2013.

[8] C. A. Davis, B. C. Hitz, C. A. Sloan, E. T. Chan, J. M. Davidson, I. Gabdank, J. A. Hilton, K. Jain, U. K. Baymuradov, A. K. Narayanan, K. C. Onate, K. Graham, S. R. Miyasato, T. R. Dreszer, J. S. Strattan, O. Jolanki, F. Y. Tanaka, and J. M. Cherry. The encyclopedia of DNA elements (ENCODE): data portal update. *Nucleic Acids Research*, 46(D1):D794–D801, jan 2018.

[9] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras.

STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, jan 2013.

[10] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. *arXiv*, jun 2017.

[11] W. Koh, W. Pan, C. Gawad, H. C. Fan, G. A. Kerchner, T. Wyss-Coray, Y. J. Blumenfeld, Y. Y. El-Sayed, and S. R. Quake. Noninvasive in vivo monitoring of tissue-specific global gene expression in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 111(20):7361–7366, may 2014.

[12] Y. I. Li, D. A. Knowles, J. Humphrey, A. N. Barbeira, S. P. Dickinson, H. K. Im, and J. K. Pritchard. Annotation-free quantification of RNA splicing using LeafCutter. *Nature Genetics*, 50(1):151–158, 2018.

[13] Y. M. Lo, J. Zhang, T. N. Leung, T. K. Lau, A. M. Chang, and N. M. Hjelm. Rapid clearance of fetal DNA from maternal plasma. *American Journal of Human Genetics*, 64(1):218–224, jan 1999.

[14] T. T. M. Ngo, M. N. Moufarrej, M.-L. H. Rasmussen, J. Camunas-Soler, W. Pan, J. Okamoto, N. F. Neff, K. Liu, R. J. Wong, K. Downes, R. Tibshirani, G. M. Shaw, L. Skotte, D. K. Stevenson, J. R. Biggio, M. A. Elovitz, M. Melbye, and S. R. Quake. Noninvasive blood tests for fetal development predict gestational age and preterm delivery. *Science*, 360(6393):1133–1136, jun 2018.

[15] W. Pan, T. T. M. Ngo, J. Camunas-Soler, C.-X. Song, M. Kowarsky, Y. J. Blumenfeld, R. J. Wong, G. M. Shaw, D. K. Stevenson, and S. R. Quake. Simultaneously monitoring immune response and microbial infections during pregnancy through plasma cfRNA sequencing. *Clinical Chemistry*, 63(11):1695–1704, nov 2017.

[16] D. D. Pervouchine, D. G. Knowles, and R. Guigó. Intron-centric estimation of alternative splicing from RNA-seq data. *Bioinformatics*, 29(2):273–274, jan 2013.

[17] L. L. Poon, T. N. Leung, T. K. Lau, and Y. M. Lo. Presence of fetal RNA in maternal plasma. *Clinical Chemistry*, 46(11):1832–1834, nov 2000.

[18] A. Saha, Y. Kim, A. D. H. Gewirtz, B. Jo, C. Gao, I. C. McDowell, G. Consortium, B. E. Engelhardt, and A. Battle. Co-expression networks reveal the tissue-specific regulation of transcription and splicing. *Genome Research*, 27(11):1843–1858, oct 2017.

[19] M. Uhlén, L. Fagerberg, B. M. Hallström, C. Lindskog, P. Oksvold, A. Mardinoglu, . Sivertsson, C. Kampf, E. Sjöstedt, A. Asplund, I. Olsson, K. Edlund, E. Lundberg, S. Navani, C. A.-K. Szigyarto, J. Odeberg, D. Djureinovic, J. O. Takanen, S. Hober, T. Alm, P.-H. Edqvist, H. Berling, H. Tegel, J. Mulder, J. Rockberg, P. Nilsson, J. M. Schwenk, M. Hamsten, K. von Feilitzen, M. Forsberg, L. Persson, F. Johansson, M. Zwahlen, G. von Heijne, J. Nielsen, and F. Pontén. Proteomics. tissue-based map of the human proteome. *Science*, 347(6220):1260419, jan 2015.

[20] J. Vaquero-Garcia, A. Barrera, M. R. Gazzara, J. González-Vallinas, N. F. Lahens, J. B. Hogenesch, K. W. Lynch, and Y. Barash. A new view of transcriptome complexity and regulation through the lens of local splicing variations. *eLife*, 5:e11752, feb 2016.

[21] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, pages 974–983, New York, New York, USA, aug 2018. ACM Press.