Paper Review For Gene2vec
by Anurag Banerjee

Gene2vec: distributed representation of genes based on co-expression[1]

Jingcheng Du et. al.

1 Jargon Glossary

functional annotation/description - description of a function of the protein that a gene produces

gene - molecular gene is a sequence of nucleotides in DNA

gene-expression - using information from a gene to synthesize either proteins or non-coding RNA

gene co-expression - when expression of two or more genes are correlated

genome - umbrella term, includes *all* genetic information of an organism

pathways - when different genes work in different sequential steps of a biological process, it is called a *genetic pathway*

transcript - A primary transcript is the single-stranded ribonucleic acid (RNA) product synthesized by transcription of DNA, and processed to yield various mature RNA products such as mRNAs, tRNAs, and rRNAs

transcription - Both DNA and RNA are nucleic acids, which use base pairs of nucleotides as a complementary language. During transcription, a DNA sequence is read by an RNA polymerase, which produces a complementary, antiparallel RNA strand called a primary transcript

transcriptome - *set* of all transcripts (including coding and non-coding)

2 Problem Description

This paper attempts to figure out, how to represent all human **gene**s as n-dimensional vectors, such that they also capture functional relatedness of the genes. These vectors are to be the *distributed* representation of the genes, in line with word embeddings (as in Natural Language Processing or NLP).

3 Problem Relevance

A naive way to think of a **gene** is that, it is made up of multiple **transcript**s; where as all transcripts in the human **genome** have been identified, the **functional annotation**s of the gene is *discrete*, *categorical* and through *manual efforts*.

In NLP, a word's vector representation (neural embedding) depends on the co-occurrence of other words in the same sentence. Leveraging this idea, the authors have used **gene co-expression** as the basis of similarity for obtaining the ***gene embeddings***.

Such gene embeddings may be used for multiple downstream tasks such as *finding gene-gene interactions*, clustering the embeddings in some dimensional space to *group genes into some functional group*, etc.

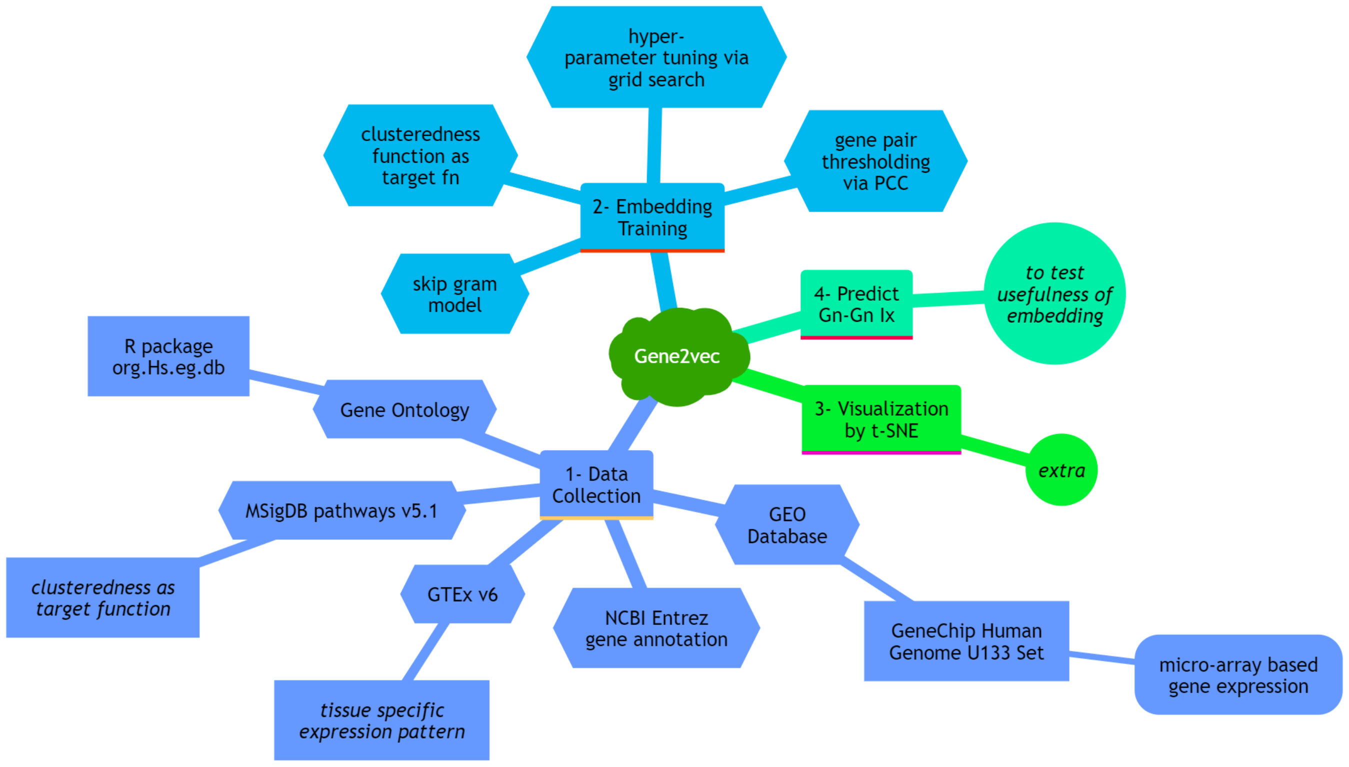
hyper-parameter tuning via grid search
clusteredness function as target fn
gene pair thresholding via PCC
2- Embedding Training
skip gram model
4- Predict Gn-Gn Ix
to test usefulness of embedding
R package org.Hs.eg.db
Gene2vec
Gene Ontology
3- Visualization by t-SNE
extra
1- Data Collection
MSigDB pathways v5.1
GEO Database
clusteredness as target function
GTEx v6
NCBI Entrez gene annotation
GeneChip Human Genome U133 Set
micro-array based gene expression
tissue specific expression pattern

Figure 1: Mindmap

4 Proposed Solution

The proposed experimental setup[2][3] mostly depends on the data collection and processing for a relatively straightforward neural network. The main components for the entire solution may be summarized in the *mindmap* in Figure 1.

The flow of the logic for each of the major components (viz., *Data Collection*, *Embedding Training*, *Visualization* & *Downstream Task*) is described next:

4.1 Data Collection

The data sources are:

1. **GEO database**'s **GeneChip Human Genome U133 Plus 2.0 Array set**, which provides gene-expression information via the older micro-array based method. Probe sets with $\geq$ 30 samples and large intra-sample variance were chosen. The gene-coexpression was measured using the *Pearson Correlation Coefficient*. The gene pairs with $\mathcal{PCC} \geq 0.9$ were used for training

2. Gene-gene interaction information is obtained from **Gene Ontology** annotations accessed via the `R` package `org.Hs.eg.db`. Gene pairs that share annotation (*GO category "Biological Process"*) form the *positive set* and those that don't, form the *negative set*. The genes were also mapped to the **NCBI Entrez Gene** knowledge base.

3. **GTEx v6** data was used to compute tissue specificity (*z-score*) by comparing avg. gene expression of all genes across 27 tissues. Apparently, this value is mapped to the gene-expression info in (1) above

4. With pathways information from **KEGG**, **Biocarta**, and **Reactome**, *clusteredness* from **MSigDB pathways v5.1** is used to setup the *Loss Function* for the optimization of the learnable parameters.

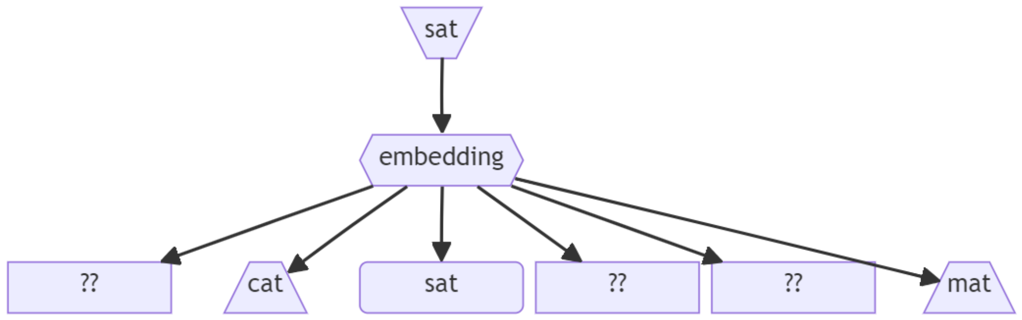

Figure 2: Skip-gram Model

4.2 Embedding Training

In general, a neural embedding aims at mapping items in a discrete space into a continuous *Euclidean* space. Embeddings enable the computation of *gradients* and thus training of a neural network.

There are two NLP methods, viz., *skip-gram model* and *continuous-bag-of-words*. Each can be quickly visualized using the following diagrams:

First let us look at ***skip-gram model*** (as in Figure 2): the target word is the input (*here, it is "sat"*) and the model has to predict the most context sensitive words (excluding stop words) (*here the words, "cat" and "mat"*). For this, the probabilities are predicted over entire vocabulary.

In the ***continuous-bag-of-words model*** (as in Figure 3): the context is the input, where the words preceding and following a target word are input, and a probability for the target word is predicted. In essence, it is just the reverse of *skip-gram*.

The **authors have used the *skip-gram model* to model the learning**. The idea involves creating an ordered vocabulary of all unique genes, and represent each gene by a one-hot vector. These vectors are input to the neural network. The network calculates loss and thus gradients based on whether or not the correct probability of co-expression was predicted by the network. Further, a grid search is performed for hyper-parameters such as ***number of iterations*** and ***dimensions of embeddings***.

Once the parameters and hyper-parameters reach apparent convergence, the vector generated by the last hidden layer forms the gene embedding. The entire process may be understood as in the Figure 4.

The cross-entropy loss attempts to minimize the following (*as per the paper*):

$$-\sum_{(g_i,g_j)\in T} Pr(g_i|g_j) \tag{1}$$

where,
T : set of highly co-expressed gene pairs
$Pr(g_i|g_j) = \frac{exp(v_i^T v_j)}{\sum_{j'} exp(v_i^T v_{j'})}$

In other words, the cross-entropy loss will maximize the likelihood of highly co-expressed genes and reduce the likelihood of unrelated ones.

> This is my understanding. Seems the paper has some discrepancy in the formula though!

For the grid search that is used to find the most useful values for the hyper-parameter, the following clusteredness formula is used:

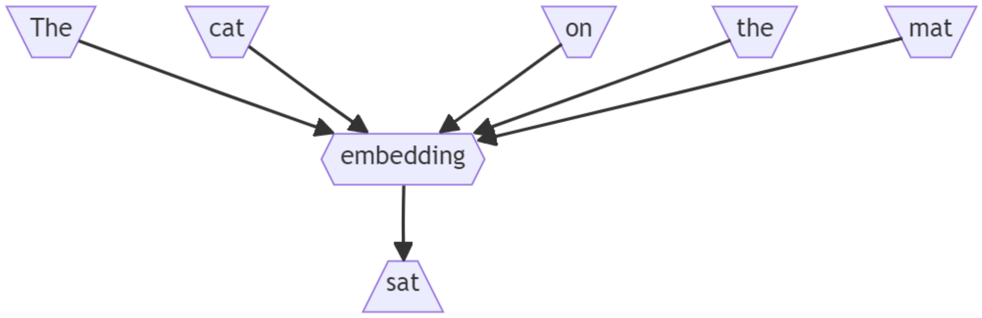

Figure 3: Continuous Bag of Words Model

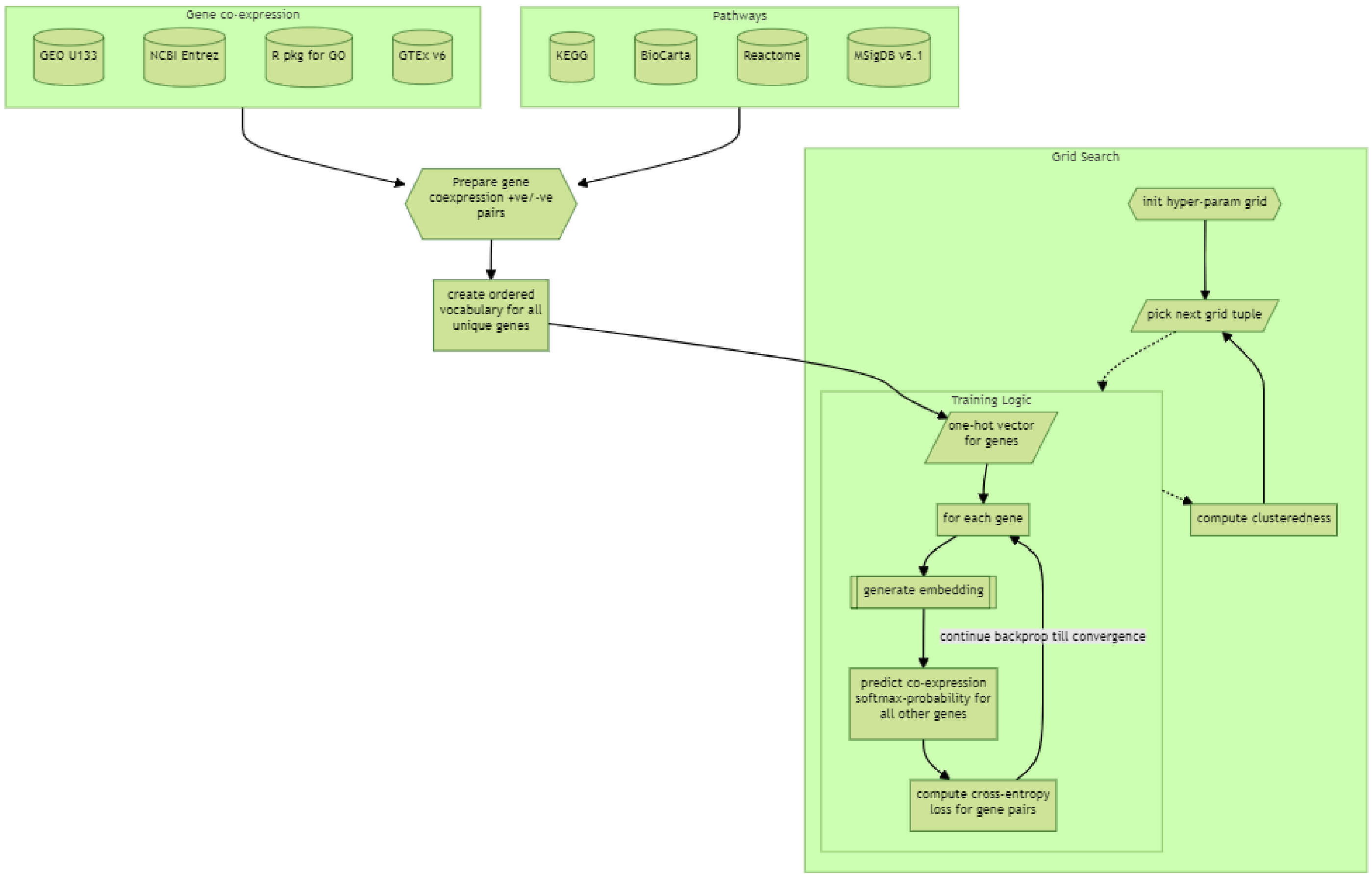

Figure 4: Embedding Training Logic

$$clusteredness = \frac{\frac{1}{|Q|}\sum_{P \in Q} \frac{1}{\#gene\ pairs\ in\ P} \sum_{g_i, g_j \in P} (v_i^T v_j)}{\frac{1}{\#gene\ pairs\ in\ Q'} \sum_{g_i, g_j \in Q'} (v_i^T v_j)} \tag{2}$$

where, the symbols have the following meaning:
Q : set of pathways in MSigDB (*have co-expression*)
Q' : set of random gene pairs
$g_i \in \mathbb{R}^d$: one-hot vector for i^{th} gene in ordered gene vocab of size d
$v_i \in \mathbb{R}^k$: embedding for the i^{th} gene, of dimension k
Further,
g_i and v_i are related as: $v_i = W \cdot g_i,\ where\ W \in \mathbb{R}^{d \times k}\ is\ the\ learnable\ parameter.$

4.3 Visualization by t-SNE

The authors utilize PCA to reduce the dimensionality of the generated gene embeddings and plot the t-SNE charts. The visual clusters are used for verifying the viability of the embeddings in terms of gene annotation similarity.

4.4 Predict Gn-Gn Ix

The downstream task of gene-gene interaction prediction is setup to establish the effectiveness of the generated embeddings in some real scenarios.

> I am not delving into this experiment since that is not the focus of my study.

5 Positive Points

- The paper utilizes concepts present in established NLP pipelines to create learned gene embeddings that are shown to be useful in downstream ML tasks.

6 Negative Points

- The description of the embedding generation could have been better
- The data sources may not reflect updated knowledge in the domain

7 Questions

1. The use of micro-array based co-expression info could be outdated - will that affect the embedding generation?
2. The actual steps in utilizing the data sources for creation of input data is not clear. What are the steps?
3. The exact description of the loss function is slightly dubious. Can it be improved?

References

[1] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, and D. Zhi, "Gene2vec: distributed representation of genes based on co-expression," *BMC Genomics*, vol. 20, p. 82, Feb 2019.

[2] J. Du, "Gene2vec github repository." Available: `https://github.com/jingcheng-du/Gene2vec`, Aug 2019. Accessed: 14 June 2024.

[3] D. Cox, "Gene2vec github repository (unofficial)." Available: `https://github.com/david-r-cox/Gene2vec`, Nov 2016. Accessed: 14 June 2024.