INTEGRATIVE APPROACHES FOR LARGE-SCALE BIOMEDICAL DATA ANALYSIS

by

ASHIS KUMER BISWAS

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at
The University of Texas at Arlington
December, 2016

Arlington, Texas

SUPERVISING COMMITTEE:
Jean X. Gao, Supervising Professor,
Chris Ding,
Gautam Das,
Chengkai Li

ABSTRACT

INTEGRATIVE APPROACHES FOR LARGE-SCALE BIOMEDICAL DATA ANALYSIS

Ashis Kumer Biswas, Ph.D.

The University of Texas at Arlington, 2016

Supervising Professor: Jean X. Gao

Advancement of the Next Generation Sequencing (NGS), also known as the High Throughput Sequencing (HTS) technologies allow researchers investigate genome, transcriptome, or epigenome of any organism from any perspective, thereby contributing to the enrichment of the biomedical data repositories for many of the lesser known phenomena. The regulatory activities inside genome by the non-coding RNAs (ncRNAs), the transcribed product of the long-neglected “junk DNA” molecules is one such phenomenon. While large-scale data about the ncRNAs are becoming publicly available, the computational challenges are being imposed to the bioinformaticians for efficient mining to get reliable answers to few subtle questions. Given the fact that a huge number of transcript sequences are retrieved every day, how can one distinguish a coding transcript from an ncRNA transcript? Can the structural patterns of the ncRNAs define their functions? Finally, from the accumulating evidences of dysregulations by ncRNAs leading to their association with a wide variety of human diseases, can one devise an inference engine to model the existing disease links as well as deduce unexplored associations?

Most prior works on ncRNA data analysis are not applicable for addressing the challenges due to the size and scope of the available datasets. In this dissertation, we present efficient *in silico* integrative methods to mine biomedical data pertaining to answering aforementioned questions. We design CNCTDiscriminator method for reliably classifying the coding and non-coding RNAs coming from any part of the genome. This is achieved through an extensive feature extraction process for learning an ensemble classifier. We design algorithm, PR2S2Clust, to characterize functional ncRNAs by considering their structural features. For this, we formulate the problem as a clustering of the structures of the patched RNA-seq read segments, which is first of its kind in literature. Finally, we propose three algorithms to deal with the disease-ncRNA association inference problem. The first algorithm formulates the inference as a modified Non-negative Matrix Factorization (NMF) problem that can handle additional features of both the entities. The second algorithm formulates the problem as an Inductive Matrix Completion (IMC) problem presenting a generalized feature integration platform overcoming the cold-start issue common to most of the prior works including the NMF strategy. The final algorithm, Robust Inductive Matrix Completion (RIMC) is presented to solve two major issues with the IMC formulation pertaining to data outliers and sparsity. For all the problems, we provide rigorous theoretical foundations of the proposed algorithms and conduct extensive experiments over real-world biomedical data available in the public domains. The performance evaluation validates the utility and effectiveness of the proposed algorithms over existing state-of-the-art methods.

ACKNOWLEDGEMENTS

First I want to thank my advisor Dr. Jean X. Gao. It has been an honor to be her Ph.D. student. I appreciate her contributions of time, ideas, and support to make my Ph.D. experience stimulating and productive. The enthusiasm she has for her research was contagious and motivational for me. I am also thankful to her for teaching me etiquette in official communication, reviewing my resume for my job applications, and foremost holding the weekly research meetings for the last five years from which, besides brain storming on the projects, I learned discipline, patience and perseverance in fulfilling project requirements from start to finish. I also want to thank my committee members Dr. Chris Ding, Dr. Gautam Das and Dr. Chengkai Li for their interest in my research, their helpful comments, and for taking time to serve in my dissertation committee.

The past and present members of the Biomedical Computing and Intelligent Systems (BioMeCIS) lab have contributed immensely to my personal and professional time at UTA. The lab group has been a source of friendships as well as good advice and collaboration. Especially, I recall countless discussions with Dr. Mingon Kang, Dr. Dongchul Kim, Dr. Shuo Li, Minh Nguyen, and Piraporn Jangyodsuk which were really encouraging for me surviving through the Ph.D. track.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. I was funded by the Enhanced Graduate Teaching Assistantship (EGTA) for the past five years except the summers and the Science Technology Engineering and Math (STEM) Graduate Teaching Assistantship for the past four summers and the current “graduating” semester. I would also like to thank the government of the Peoples’

Republic of Bangladesh and its people for supporting my studies up to the Masters level, almost for free.

During my Ph.D. studies I lost few of my family members who were always a great source of motivation for me. Among them my dadu (grand-father), borda (most senior cousin Joydeb da) would have been very happy for me today. I miss them and always pray for their departed souls.

I would like to thank my father Dr. Sunil Kumer Biswas and mother Satyabati Biswas for their love, encouragement, sacrifice they made (and still are making) to raise me and support me. My sister Sipra Biswas and her husband Dr. Sudip Kumar Karmoker were always great source of inspiration for me; and I want to thank them for regularly keeping me posted about two of my cutest niece and nephew Ira and Shoumil.

Finally, my lovely wife Madhuri Debnath: you have been there from the beginning, for this tumultuous but fun ride. This journey was as much yours as it is mine. I strongly believe I am the luckiest in the world to be your husband. Your support in all my pursuits, your unwavering trust on me, amazing patience, and compassionate encouragement are one of a kind. Thank you Madhuri.

November 21, 2016

To my parents.

RELATED PUBLICATIONS

- **Ashis Kumer Biswas**, Dong-Chul Kim, Mingon Kang, and Jean X. Gao, "Robust Inductive Matrix Completion Strategy to Explore Associations between LincRNAs and Human Disease Phenotypes," *in 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Shenzhen, China, Dec 15–18, 2016.
- **Ashis Kumer Biswas**, and Jean X. Gao, "PR2S2Clust: Patched RNA-seq Read Segments' Structure-oriented Clustering," *Journal of Bioinformatics and Computational Biology (JBCB)*, World Scientific Publishing, vol. 14, no. 5, pp:1650027, 2016.
- **Ashis Kumer Biswas**, Mingon Kang, Dong-Chul Kim, Chris H. Ding, Baoju Zhang, Xiaoyong Wu, and Jean X. Gao, "Inferring disease associations of the long non-coding RNAs through non-negative matrix factorization," *Network Modeling Analysis in Health Informatics and Bioinformatics (NetMAHIB)*, Springer Vienna, vol. 4, no. 1, pp. 1–17, 2015.
- **Ashis Kumer Biswas**, Jean X. Gao, Baoju Zhang, and Xiaoyong Wu, "NMF-based LncRNA-Disease Association Inference and Bi-clustering," *In the 14th IEEE International Conference on Bioinformatics and BioEngineering (BIBE)*, IEEE, pp. 97–104, Boca Raton, Florida, USA, 2014. Received the best paper award in the Bioinformatics category.
- **Ashis Kumer Biswas**, Baoju Zhang, Xiaoyong Wu, and Jean X. Gao, "CNCT-Discriminator: Coding and Non-coding Transcript Discriminator – An Excursion through Hypothesis Learning and Ensemble Learning Approaches," *Journal*

of Bioinformatics and Computational Biology (JBCB), Imperial College Press, vol. 11, no. 5, pp:1342002, 2013.

- **Ashis Kumer Biswas**, Baoju Zhang, Xiaoyong Wu, and Jean X. Gao, "An Information Integration Approach for Classifying coding and non-coding genomic data," *In the International Conference on Communications Signal Processing and Systems (CSPS)*, Springer, pp. 1085–1093, Tianjin, China, 2013.
- **Ashis Kumer Biswas**, Baoju Zhang, Xiaoyong Wu, and Jean X. Gao, "QLZCClust: Quaternary Lempel-Ziv Complexity based Clustering of the RNA-seq Read Block Segments," *In the 13th IEEE International Conference on Bioinformatics and BioEngineering (BIBE)*, IEEE, pp.10–13, Chania, Greece, 2013.
- **Ashis Kumer Biswas**, Jean X. Gao, Xiaoyong Wu, and Baoju Zhang, "Integrating RNA-seq transcript signals, Primary and Secondary Structure Information in Differentiating coding and non-coding RNA transcripts," *In the 5th International Conference on Bioinformatics and Computational Biology (BICoB)*, ISCA, Honolulu, Hawaii, USA, 2013.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

1.1 Non-coding RNAs (NcRNAs) and Preliminaries

In early 2003, when researchers officially[1] completed sequencing the human genome, they were surprised to find out that only about 21,000 protein coding genes are scattered along the 3 billion DNA bases and in between are megabases of "junk", or so it seemed to the researchers back then. It was until the ENCODE projects published their decade-long investigations and uncovered the truth that the human DNA is not actually littered with useless bases, rather most of these have functional importance [1]. The project revealed that the regulation of the genes, which is more complicated than what was previously thought, is influenced by multiple stretches of regulatory DNA located both near and far from the gene itself and by strands of RNAs, that never are translated into proteins, the so-called non-coding RNA (ncRNA) molecules.

With the advancement of the High Throughput Sequencing (HTS)[2] platform, researchers can sequence DNA and RNA much more quickly and inexpensive way than the previously used Sanger sequencing and as such it revolutionized the study of ncRNAs and other genomic molecules. The platform includes the RNA-seq tool that can quantify expression scores of any stretch of RNA transcript under investigation, especially the stretches from the ncRNA parts of the transcripts, more effectively and

[1] `https://www.genome.gov/11006929/2003-release-international-consortium-completes-hgp/`, Last accessed: 12-07-2016 9:12AM

[2] *a.k.a.* Next Generation Sequencing (NGS), the term deliberately avoided throughout the dissertation to promote the usage of HTS in place of NGS to refer to the decade old platform.

accurately which otherwise would not be possible to calculate with the almost obsolete microarrays [2]. Whereas the other tool under the HTS umbrella, known as the ChIP-seq that can map the interactions between the transcription factor proteins and the DNA sequences comprehensively across the entire genome, that essentially have influence on the rate of transcription of genetic information from DNA to messenger RNA to coding and non-coding RNA transcripts as well as forming gene network scaffolds irrespective of their coding potentials [3]. Essentially this method uses an antibody to home in on a particular DNA-binding protein (called the transcriptor factor, TF) and helps pinpoint the locations (called the transcriptor factor binding sites, TFBS) where that particular protein works. A typical HTS experiment produces around 100 gigabytes of short-read sequence data that at first needs to go under alignment with respect to the reference genome before retrieving the corresponding information about specific ncRNA sites.

NcRNA transcripts like other RNA transcripts are single-stranded molecules and most functional ncRNAs can fold to itself by forming base pairings. These pairings occur between the bases G and C, U and A and sometimes G and U. Such pairings are the building blocks of the structural identity of the ncRNA transcripts. There are *in silico* structure prediction tools being used. For example, Vienna RNA Package which predicts the secondary structure of a given ncRNA with minimum free energy [4]. Whereas the Single nucleotide polymorphisms (SNPs), the most frequent genetic variations among people link to abnormalities in gene expressions, their dysregulations, impacts on the stability of the secondary structures have become a *de facto* standard dataset to utilize in ncRNA function association problems [5].

The datasets and packages described above can be accessed through publicly available repositories. As the datasets are heterogeneous to each other, these open up a broad spectrum of analysis pipelines for the bioinformaticians involving one or multiple sources to reliably retrieve desired information about the ncRNAs. Most of the datasets are large-scale and require special computational treatments during analysis, while others are inherently sparse that stymie the applications of straightforward machine learning algorithms to accomplish a goal. Since all the datasets are coming from actual problem environments, e.g., either from sequencing of tissues of classified patients, or genome-wide association study to identify SNPs, any new inferences from the models built in this dissertation can not be readily evaluated (at least computationally verified) unless supported by any independent and experimentally obtained data or findings.

1.2 Motivation and Challenges

Computational challenges are being imposed to the bioinformaticians as plethora of ncRNA datasets are becoming available in the public repositories on a regular basis. Of the many challenges, developing an *in silico* classification of the non-coding and coding RNA transcripts will help the researchers explore these two entities separately, determine activities where a group of ncRNAs are always involved or even study closely related ncRNAs in terms of their functions. There are two categories of classifiers for this purpose. Firstly, the alignment-based approaches try computationally predict which possible protein the given transcript would produce, and then exhaustively align the protein sequence over a protein database. While the approach may look seemingly intuitive, as it shows low sensitivity as it declares the given transcript as non-coding if it misses in the protein database. The problem narrows down to the choice of protein database. The alignment-free approach extracts features of the given transcript and train a classifier. As discussed in earlier section that there are many heterogeneous data available about a transcript, researchers face two questions while developing the discriminatory models: which feature-set would be the best choice, and what classification algorithm would reliably classify the two types of transcripts.

The RNA-seq platform offers researchers deep understanding about the transcriptomics of any organisms. The analysis pipeline begins by first mapping the raw short-read sequences to the reference genome. Then clusters of overlapping short-read sequences are grouped into blocks of reads with similar start and end positions. Finally, the overlapping and closely spaced read blocks are further grouped to form read-block groups that will be called read-segments. The segments are representative fragment for the constituent raw short-read sequence blocks, and considering these instead of the raw short-read set greatly reduces size of the dataset and also allows the application of computationally sophisticated algorithms. However, since most of the functional ncRNAs form secondary structures, and their corresponding functions can be characterized through the structural signatures, the challenge now is to efficiently characterize functions of the ncRNA read-segments through these signatures, as well as group together similar ncRNAs in that perspective.

Accumulating evidences show that the long ncRNAs (lncRNAs), a subclass of the ncRNAs having length more than 200 bases are discovered to be associated with a variety of human diseases through many the dysregulations and mutations. Thus, a comprehensive understanding of potential disease-related lncRNAs can facilitate development of our current knowledge-base in a way capable of explaining accurately the molecular mechanisms of human diseases, their implications and also facilitate the diagnosis, treatment, prognosis and prevention. Most prior work on lncRNA-disease association inference do not offer data integration interface where varieties of data about both the entities can be provided and utilized to predict association scores. Some others suffer the *cold-start* problem, where the entire inference model requires re-training in order to predict association between novel lncRNA and diseases. This issue hinders the applicability of the method by biologists and researchers from the relevant field.

1.3 Dissertation Organization

In Chapter 2, we presented the CNCTDiscriminator method to classify coding and non-coding RNA transcripts [6, 7, 8]. Our key contribution is to consider RNA-seq expression scores of the transcripts along with other standard features to discriminate the transcripts. Prior works focused on evaluating their methods on a small dataset lacking diversity. We provide a robust large-scale benchmark dataset covering a large population of transcripts in each of the two groups. In addition to employing the traditional hypothesis model, we also proposed a hierarchical ensemble strategy to accomplish the task of classification by building up a group of hypothesis models at the first level based on different feature categories, then the outputs of the groups are fed as input features to build a hypothesis model at the next level thereby extending the reliability of the overall prediction performance of each individual models. We provide extensive experimental results over the benchmark datasets that illustrate the effectiveness of our technique.

In Chapter 3, we defined the problem of clustering the patched read-segments through secondary structure perspective [9, 10]. We present a platform to tackle the clustering problem. We proposed several pairwise distance measures for the patched read segments based on the perspective. Hierarchical clustering algorithms are employed to cluster the segments through the respective pairwise distance matrices. We offer methods to extract features from the predicted secondary structures of the patched read-segments, so that we can represent the patched read-segments as vectors. Classical partitional clustering algorithms are employed to cluster the vectors. We present ensemble approach to aggregate multiple clustering results that the ability to boost clustering performance. Comprehensive experiments on real datasets show the significant improvement of our methods over the state-of-the-art methods.

In Chapter 4,we propose a computational framework for constructing the lncRNA-gene co-modules based on the integration of prior knowledge we have [11, 12]. We incorporated the predicted lncRNA-disease association and experimentally validated gene-disease association, gene-gene interaction data, expression profiles of both lncRNAs and genes in a non-negative matrix factorization framework. We also applied the similar matrix factorization approach on each of the association data alone to further cluster the lncRNAs in terms of meaningful disease groups.

In Chapter 5, we demonstrate that the integration of diverse features of the lincRNAs and the diseases available through publicly available data-servers can overcome worse predictive performance issue faced by the inference tools which occurs due to the extreme sparsity inherent to the lincRNA-disease association dataset. We provide an application of the Inductive Matrix Completion (IMC) method and show induction on novel diseases and novel lincRNAs that are not seen at the training time, unlike the traditional matrix factorization methods and network-based inference methods that are transductive by nature. We present extensive experiments and comparative study to show effectiveness of our proposed method.

In Chapter 6, We present RIMC algorithm, its correctness and proof for its convergence [13]. We apply RIMC to the available association data between human lincRNAs and OMIM disease phenotypes as well as a diverse set of side information about the lincRNAs and the diseases. We demonstrate the performance of our method in terms of $precision@k$ and $recall@k$ at the top-k disease prioritization to the subject lincRNAs. We also provide a comparative study considering the state-of-the-art lincRNA-disease ranking solutions. Finally, we present results focusing on the effectiveness of the induction property of RIMC along with the standard IMC approach.

CHAPTER 2

CNCTDiscriminator: The Coding and Non-coding Transcript Discriminator

2.1 Introduction

2.1.1 The Coding and Non-coding Transcript Discrimination Problem

The central dogma of Molecular Biology states that the portions of a DNA are first transcribed into RNA transcripts (i.e., the mRNAs) that later get translated to different proteins. This pipeline was proved incomplete after the discovery of transcipts that never get translated to any protein. These are the non-coding RNA (ncRNA) transcripts, and they form another tier of gene expression with many different cellular activities including gene silencing, replication, gene expression regulation, transcription, chromosome stability, protein stability, translocation, localization and RNA modifications, processing and their stability [14]. Classifying the non-coding and coding transcripts that produce the RNA transcripts will help the researchers explore the two types of genes separately, determine activities in which a group of ncRNAs are always involved or even study closely related functions by different groups of ncRNAs.

Among the *in silico* strategies to discriminate coding and non-coding transcripts, two broad categories were being observed – the alignment-free and the alignment-based approaches. Each of these approaches employed either sequence based or structure based features or both from the transcripts. For instance, the Open Reading Frame (ORF) length is one of the most intuitive feature used to distinguish non-coding RNAs from the mRNAs. Because short putative ORFs can be occurred by chance within a long non-coding transcript and minimum ORF cutoffs are applied to reduce the likelihood of falsely classifying non-coding transcript as mRNAs [15]. Although straightforward to apply across large datasets, ORF length is an unreliable feature. When ORF cutoff length is big, very long non-coding RNAs having putative ORFs may be misclassified as mRNAs, and also when ORF cutoff length is small, many mRNAs are misclassified as non-coding RNA.

Discriminating long non-coding RNAs from the mRNAs can also be accomplished by assessing putative ORFs for similarity to known proteins using an alignment-based approach[16]. But the approach is limited by the number of genomes available for comparison.

Programs like RNAz [17] employed the presence of conserved predicted RNA secondary structure to identify non-coding RNAs that have different functional properties. But, using these programs to detect transcripts as non-coding RNAs is likely to lead to significantly false positive and false negative discoveries, since conserved secondary structures are also commonly found in mRNAs, especially at the 3' UTRs. The functional non-coding RNAs may contain secondary or tertiary structures with non-canonical base interactions [18] that are not considered by most structure prediction programs. Moreover, Rivas et al. [19] outlined that secondary structures are not sufficiently different from the predicted stability of a random RNA sequence that make it harder to build stable discriminating strategy utilizing structure information only. Therefore, researchers emphasize on multi-domain combination strategies to solve the problem.

2.1.2 Existing Combination Strategies

Among the combination strategies, CPC (Coding Potential Calculator) [20] employed a few features of the transcripts pertaining to the ORFs and the possible proteins for the transcript which was accomplished by doing an exhaustive alignment of the transcript sequences over the entire UniProt database of proteins using BLAST. The approach is not suitable because it is prone to be biased for classifying as non-coding those transcripts that do not have good hits from the protein databases. Thus, CPC is not robust in situations where the researchers are looking for a reliable annotation of a novel RNA transcript.

Alignment-free strategy like PORTRAIT [21] was developed with a goal to screen the non-coding RNAs which might have been caught during the transcriptome sequencing process. The method first translates the ESTs from the given input sequences into possible proteins. The SVM induced protein coding and non-coding transcript discriminating models were built by employing nucleotide composition, length, amino acid composition of the translated protein, ORF length, isoelectric point, entropy and hydropathy features. It classifies only the transcripts which are at least eighty nucleotides long is one of it's drawback. But there may be cases where it would be required to work with shorter RNA sequences having length less than eighty nucleotides.

While such methods have been widely used, recently another alignment-free combination approach was employed in CPAT [22] to classify non-coding and coding transcripts which is fast as it extracts only four ORF-based features out of the transcripts to build the supervised logistic regression model. Although very fast and seemingly straightforward approach, CPAT is not reliable in large scale datasets where transcripts from both classes do not exhibit any consistent discriminatory behaviors at the ORF feature space.

2.1.3 Outlines

The existing methods discussed so far employed either sequence, or ORF based features from the transcripts. However, with the advent of the High Throughput Sequencing platform – RNA-seq [2], expression scores of a large number of transcripts become easy to calculate. Expression scores of both coding and non-coding transcripts can be quantified from the RNA-seq short-read sequences more effectively and accurately than using microarrays. The structural patterns of the blocks of many such short-read sequences obtained from typical RNA-seq experiments were previously shown promising in novel ncRNA transcript discovery [23]. But such expression scores were never employed as features to address the transcript classification problem. In this paper, we have considered integration of such expression scores and presented applicability of the scores alongside other features to discriminate transcripts.

Again, preparation strategies of benchmark dataset by the existing three methods focused extracting only a very small portion of the annotation data sources as compared to the total number of annotations available in the sources, thereby the methods lost, to some extent, the reliability of the prediction performance when applied to predict a large scale robust benchmark. Here, we addressed the issue of benchmark construction.

All of the existing *in silico* strategies used traditional hypothesis learning scheme – that is trained with a set of instances each having a set of features and a model is built from it which can be used later to predict class labels of new and unknown instances. One limitation of this scheme is feature selection. If a large number of features per instance are employed, without feature selection, while building such models, the scheme may show poor performance. Whereas selecting prominent features out of a large number of features is not an easy task. In addition to employing the traditional hypothesis model, we also proposed a hierarchical ensemble strategy to accomplish this task by building up a group of hypothesis models at the first level based on different feature categories, then the outputs of the groups are fed as input features to build a hypothesis model at the next level thereby extending the reliability of the overall prediction performance of the each individual models.

Finally we performed a comparative analysis of our proposed discriminating scheme with the current state-of-the-art systems – CPC [20], PORTRAIT [21] and CPAT [22].

2.2 Methods

2.2.1 Obtaining the Dataset

We collected with the RNA-seq experiment data which was generated from a High Throughput Sequencing (HTS) of RNAs from brain and other cell lines of human (*homo sapiens*) samples [24]. We retrieved the RNA-seq short-read aligned BED/BigWig formatted files associated with the GEO accession `GSE30222` from the NCBI's Gene Expression Omnibus.

We retrieved the annotations of the protein-coding transcripts and long non-coding RNAs found in the main chromosomes of human from GENCODE (version 16) [25]. The total number of annotated protein coding and non-coding transcripts found in the annotation dataset were 94,847 and 22,444, respectively. We further retrieved species specific 47,250 protein coding annotations from NCBI RefSeq [26] (March 10, 2008 update). Short non-coding RNA transcripts were obtained from Ensembl (`ftp://ftp.ensembl.org/pub/release-71/fasta/homo_sapiens/ncrna/Homo_sapiens.GRCh37.71.ncrna.fa.gz`). This dataset contains 19,878 non-coding transcripts. For the four datasets, we kept only those transcripts which have length less or equal to

1000nt. Table 2.1 summarizes the number of transcripts of each class from the four datasets.

Table 2.1: Summary of the transcript annotations considered in this study

Data Source Name	**Transcript type**	**Number of Annotations**[a]
GENCODE (ver 16)	Protein coding	37,640
RefSeq (rel 3/10/08)	Protein coding	8,436
GENCODE (ver 16)	Non protein coding	17,345
Ensembl (ver GRCh 37.71)	Non protein coding	17,651

[a] after the size filter was applied to the original data source.

We randomly selected 80% transcripts from each of the four datasets for training the supervised learning method, and the remaining 20% of each of the datasets were stored as an independent benchmark test set. This test dataset will be used to assess performances of our method and existing three methods. From Table 2.1 we noticed that there are 11,080 more coding transcripts than the non-coding transcripts. This uneven distribution of the two classes would have subtle impact on the quality of training models. To remedy this problem, we used random oversampling [27] of the minor dataset (i.e, the non-coding dataset in our case) before the training phase. Thus, the size of the training set became 73,720 transcripts and that of the test set was 18,432 transcripts. The dataset is made available in public (please read Section 5 about the retrieval).

2.2.2 Features for the Supervised Learning

There are four categories of features we selected in our study – (i) base compositions, (ii) open reading frame (ORF) statistics, (iii) transcript expression scores and (iv) properties of the secondary structure of the transcripts. There are many nucleotide composition-based measures that can be obtained from the bare transcript sequences. Longest ORFs can be obtained through application of probability modeling from the given transcript sequences. The third category of features comprise of Read Per Kilobases Per Millions short reads (RPKM) scores that are the normalized expression scores for a given transcript in several different cell lines observed in RNA-seq experiments. Finally, the properties of the predicted secondary structures of the transcripts also have potential in the transcript discrimination problem. In the next four sections we discuss the extraction of these features in detail.

2.2.3 Extracting Composition based Features

From the nucleotide sequences of transcripts we computed global G+C content, G+C content in the first, second and third positions of the codon bases. The compositions of unigrams, bigrams and trigrams for the entire length of the transcripts were also calculated. Finally after including the length of the transcript as another composition based feature, the total number of composition features become 89 (4 G+C content related, 4 unigrams, 16 bigrams, 64 trigrams and 1 length measure).

2.2.4 Extracting the ORF based Features

Protein coding transcripts have very well defined properties – a 5' cap, 5' and 3' untranslated regions (UTR), an open reading frame (ORF) and a poly-A tail. The ORF is mostly unique feature for protein coding transcripts. We extracted the longest ORF within a given transcript sequence using `getorf` tool from the EMBOSS-6.5.7 package [28]. Once an ORF is predicted, we extracted different features of the probable translated proteins. For instance, isoelectric point, mean hydropathy, polarity scores, base and acidity profiles, amino acid compositions. Fickett scores of the ORF, ORF coverage scores [22] were also calculated. Finally, after the inclusion of the total number of predicted ORFs from each transcript, we could extract 35 ORF based features in this category.

2.2.5 Extracting the Expression Scores of the Transcripts

From a typical RNA-seq experiment we get short read sequences which are first aligned to a reference genome before the expression analysis could be started. The alignment results are stored in a variety of file formats. We dealt with two of them – BigWig and BED. The bigwig is a binary indexed file containing coverage scores for RNA transcripts with one nucleotide span under specific condition. Suppose a bigwig file contains all the scores of n different chromosomal positions, and we have m different transcripts for which we want to compute the total expression scores. Since n is much larger than m, we need a fast way to sum up all the scores that map to the m transcripts rather than doing a linear scan. We proposed Algorithm 1 to solve this problem.

In Algorithm 1, we called several functions in lines 2, 7, 10. In line 2, the function **ranges** extracts the start and end coordinates of the scores of the chromosome under consideration fom the given bigwig file BW. Let, the total number of such ranges be n_1. As the bigwig files are binary-indexed, the cost of the extraction of the ranges is $O(n_1)$. In line 7, we created an interval tree having m intervals (ranges). The complexity to build the interval tree here is $O(m \lg m)$. Then in line 10, the function **searchTree** searches a query range of the chromosome C into the subject interval tree for a full overlap. This complexity for this search is $O(\lg m)$, while the total complexity of the **for** loop of lines 9–11 is $O(n_1 \lg m)$. So overall complexity of Algorithm 1 is $O(n_1 \lg m)$ which is more efficient than the brute-force linear scan of the query ranges over the subject transcript ranges that would have an exhaustive

Algorithm 1 Computing expression scores of m transcripts of chromosome C from a bigwig file

Input: (i) A BigWig file, BW containing n short read map entries, (ii) a list of transcript annotations, T containing m entries of them. Each of the T entry has a field *id* and two coordinates – *start* and *end* representing the transcript start and end positions with respect to the chromosome C.

Output: A list of transcripts containing all the information of T with an additional information– "score" of each transcript that will represent the expression scores of the transcripts.

```
if $BW$ contains scores of chromosome $C$ then
   $queryRanges \leftarrow \textbf{ranges}(BW[C])$
   $RangeList \leftarrow \emptyset$
   for $i = 1$ to $m$ do
      $RangeList[i] \leftarrow (T[i].start, T[i].end)$
   end for
   $IVT \leftarrow \textbf{BuildIntervalTree}(RangeList)$
   $hits \leftarrow \emptyset$
   for $j = 1$ to $\textbf{count}(queryRanges)$ do
      $hits[j] \leftarrow \textbf{SearchTree}(queryRanges[j], IVT)$
   end for
   for $k = 1$ to $\textbf{count}(hits)$ do
      if $hits[k]$ hits on the subject $T[i]$ then
         Add the $hits[k].score$ to $T[i].score$
      end if
   end for
end if
return $T$
```

complexity of $O(n_1 m)$. We used Algorithm 1 to retrieve the nucleotide coordinates along with the scores from each of the bigwig files.

Expressions of RNA transcripts from an RNA-seq experiment is often quantified with a measure "RPKM"[29], which is defined in Equation 2.1.

$$RPKM = \frac{n(ER)}{n(TR)(\text{millions}) \times Len(\text{kB})} \tag{2.1}$$

In the Equation 2.1, $n(ER)$ is the total number of short-reads mapped to a specific transcript under consideration, $n(TR)$ is the total number of short-reads mapped in the experiment (in Million unit), and Len is the length of the transcript (in kilobase unit).

In order to compute RPKM score of a transcript, we added all the one-nucleotide span scores of all the coordinates that map in between the start and the end coordinates of the transcript. The summation is the $n(ER)$ in the above equation. For an RNA-seq experiment mapping a total of 50 million short-reads would have $n(TR)$ value 50 for instance. Finally, the absolute difference between the start and end coordinates of the transcript is the length of the transcript (Len). By plugging in the three values into the RPKM equation (shown above), we can compute the RPKM score of a transcript. By following this approach, we computed the RPKM scores of all the transcripts in different bigwig files. The RPKM scores of the transcripts from aligned BED formatted file can be computed using the python script `rpkmforgenes.py` [30].

2.2.6 Extracting Features from the Predicted Secondary Structures

RNA transcripts are single-stranded molecules and can fold to itself by forming base pairings. These pairings often occur between the bases G and C, U and A, and sometimes G and U. The base pairings form the structural components of the RNA transcript. The components are shown in Figure 2.1.

The important components of secondary structures are – (i) helix or stem, which is a consecutive stacking of base pairs; (ii) loop, that is a region of unpaired bases; (iii) hairpin loop, which is a loop enclosed by a helix or stem; (iv) multi-loop, which is a loop region from which three or more helices arise; and (v) internal loop, which is a loop inside a helix.

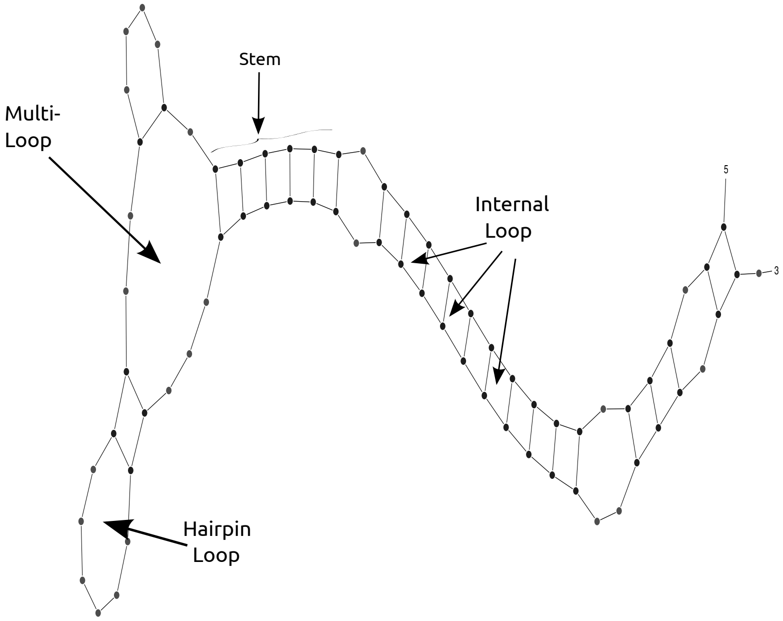


Figure 2.1: Components of the secondary Structure of the Non-coding transcript (`transcript id ENST00000390187`) from the ENSEMBLGRCh v37.71 data source.

Among the several approaches to predict RNA secondary structures, the Vienna RNA package “RNAfold” [4] is the most widely used. It is based on a dynamic programming algorithm with a quadratic computational complexity, aiming to predict the secondary structures of a given RNA sequence with minimum free energies. We applied the program to predict the secondary structures of all the transcripts listed in Table 2.1. The program reported the minimum free energy (MFE) of each of the transcript and generated a text file containing the predicted secondary structures in bracket notation. Then we extracted the following features of the predicted secondary structures: (i) number of paired bases, (ii) number of hairpin loops, (iii) number of multi-loops and (iv) minimum free energy (MFE).

2.2.7 Hypothesis Classifier

We used SVM^{perf} that is an implementation of the Support Vector Machine (SVM) formulation for optimizing multivariate performance measures [31]. Since our training data set contains 73,720 instances, training with a non-linear SVM would be expensive due to its quadratic computational complexity and in practice does not show good performance. Whereas the SVM^{perf} enables training the conventional linear classification SVMs by optimizing error rate in time that is linear in the size of the training data set. During the training, we set the trade-off between training error and margin to be 20, the loss function to be the error rate, and we did not use any bias feature for the particular classifier.

2.2.8 Feature Specific Ensemble Classifiers

Ensemble learning refers to a collection of methods that learn a target function by training a number of individual weak learners and combining their predictions. The features are first grouped into four broad categories – sequence composition features, ORF features, RPKM features and secondary structure features. We then build four separate hypothesis models given the four categories of features. Then we applied the stacked generalization approach [32] at the combiner, i.e., the output patterns of the first level experts serve as input to the second-level expert that is also a hypothesis learner. We used SVM^{perf} to build learners of each level of the Ensemble classification.

2.2.9 Training and Testing using the Classifier

All feature values of the training and test instances were normalized to fit into range [-1, 1]. The training and test steps by the two types of classifications are described in following two subsections.

2.2.9.1 Steps in the Hypothesis Learning

- Step 1: For feature combination "`f`" from the 15 possible combinations of the 4 feature categories, get the training set containing the particular feature values.
- Step 2: Do a 10-fold cross validation with the training dataset and report the average performance scores. Finally, train with all the entries in the training dataset and store the model.
- Step 5: Repeat steps 1 and 2 for the remaining 14 possible combinations.
- Step 6: Predict the independent benchmark set using each of the final training models and report the performance scores.

2.2.9.2 Steps in the Feature Specific Ensemble Learning

- Step 1: Randomize the instance ordering of the training dataset for feature "`f`" from the 4 feature categories and split the set into 10 blocks having equal number of training instances.
- Step 2: Do the exact same ordering of instances for the training datasets of remaining 3 feature categories and split into 10 blocks in a similar fashion as step 1.
- Step 3: For the feature category "`f`", do the 10 fold cross validation using the 10 blocks prepared in previous steps, and report the average performance scores. Moreover, the predicted labels for each of the 10 blocks were stored.
- Step 4: Repeat step 3 for the remaining 3 feature categories.
- Step 5: The four predicted labels of each of the training instances reported by each of the four category classifiers are then combined into a new training file having the four predicted labels as four features with the true labels.

- Step 6: The new training file is given to a hypothesis classifier and applied 10-fold cross validation. The performance scores were recorded.
- Step 7: At the end, re-train the second level classifier with all the instances of the new training file and store the model.
- Step 8: Re-train all the four first level classifiers with all the instances of the training set and store the respective models.
- Step 9: Predict the independent benchmark set by first extracting the 4 categories of features of all the transcripts in the test set and fed these to the four level-1 classifiers. The prediction results by each of the classifiers were then combined to form a new test file which is given to the level-2 classifier for the final prediction. Then the performance scores were reported.

2.2.10 Evaluating the Classifiers

The test datasets were given to the classification models for prediction and based on the true class labels and the predicted class labels the confusion matrices containing four frequency scores – True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for the classification experiments were prepared. Here, TP denotes the number of positive samples corrected predicted as positive, TN denotes the number of negative samples correctly classified as negative, FP represents the number of negative samples misclassified as positive, and FN denotes the number of positive samples incorrectly classified as negative. In this binary classification problem we followed the convention to represent a non-coding RNA transcript to be in the "negative" class and a coding RNA transcript in the "positive" class. For all the classification runs, we computed the four performance measures of binary classification: precision, recall, accuracy and F_1-score which are defined in equations 2.2–2.5.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.3}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.4}$$

$$\text{F}_1\text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.5}$$

2.3 Experimental Results and Discussion

We prepared fifteen bar plots of performance scores by the fifteen classifiers we built by training with each of the fifteen possible feature combinations which is shown in Figure 2.2. Each plot represents accuracy (A), precision (P), recall (R) and F1-score (F1) in both 10-fold cross validation (CV) and testing with the independent benchmark (Test) by a single hypothesis classification model trained with certain combination of features. It can be observed in the plots that a single feature category classifier alone can not outperform any classifier which is trained with more than one feature categories.

The RPKM feature alone shows the worst discrimination performance. In search of the reason behind this, we identified that not all coding and non-coding transcripts exhibited expressions in the RNA-seq experiment we dealt with. Thus, the training data with the RPKM features became sparse and are not suitable alone. However, applications of this feature in conjunction with other category of features enhances the classification performance.

We further ranked the feature combinations in the decreasing order of ROC area and found that the combination of composition and ORF based features yields the

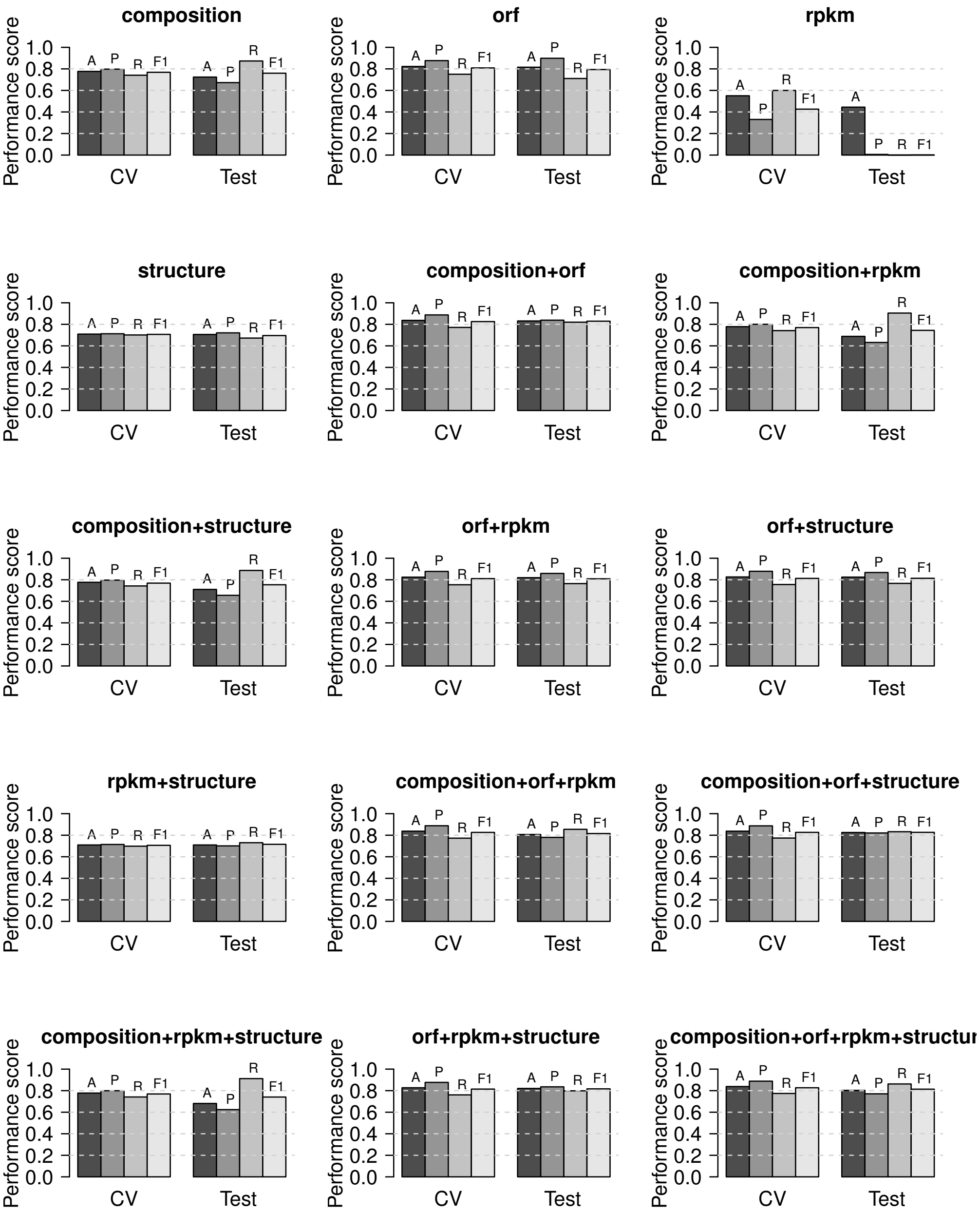


Figure 2.2: Separate single hypothesis classification performance built by all the 15 possible combinations of the 4 feature categories. Left most four bars of each plot represent accuracy (A), precision (P), recall (R) and F1-score (F1) respectively from a 10-fold cross validation (CV), whereas the rightmost four bars represent the same four performance measures when the corresponding classifier is given the independent test set for prediction.

most area under the ROC curve. Table 2.2 lists the ranks along with the proportion of the Area Under the ROC curve (AUC) by the classifiers built with the combination.

Table 2.2: Ranks of the combination of feature categories to build single hypothesis classifiers

Rank	Combination of Feature Categories	AUC
1.	composition+orf	0.8917
2.	composition+orf+structure	0.8915
3.	composition+orf+rpkm+structure	0.8907
4.	composition+orf+rpkm	0.8906
5.	orf+rpkm+structure	0.8844
6.	orf+structure	0.8838
7.	orf+rpkm	0.8784
8.	orf	0.8776
9.	composition	0.8343
10.	composition+structure	0.8335
11.	composition+rpkm+structure	0.8315
12.	composition+rpkm	0.831
13.	rpkm+structure	0.7716
14.	structure	0.7705
15.	rpkm	0.4724

Figure 2.3 illustrates the 10-fold cross validation as well as the test performance of our feature-specific ensemble approach. From the figure it is evident that both the training performance and the test performances were consistent. However, this approach falls behind few feature combination specific single hypothesis classifiers as reported in Table 2.2 in terms of “recall”.

Next, we compared the three existing state-of-the-art systems – CPC (Coding Potential Calculator) [20], PORTRAIT [21] and CPAT [22] with our top ranked 3 hypothesis classification systems built with combinations of feature categories as well as the one built with the ensemble based approach. All these systems have publicly accessible web interface where we can perform the predictions based on the inputs

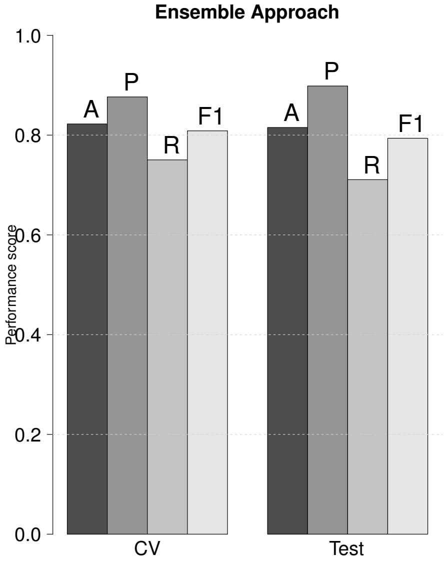


Figure 2.3: Performance of the CNCTDiscriminator – ensemble approach. Left most four bars represent accuracy (A), precision (P), recall (R) and F1-score (F1) respectively from a 10-fold cross validation (CV). And the rightmost four bars represent the same four performance measures when the Ensemble classifier is given the independent test set for prediction.

we provide. We evaluated our system and the existing three systems with the independent test dataset we prepared as discussed in Section 2.1. We measured the prediction performances of each of the systems after computing the confusion matrices. However, the PORTRAIT prediction system could not predict 634 input RNA transcripts which were less than eighty nucleotides long. In this regard, we evenly added the number of coding and non-coding transcripts from these non-predicted test examples into respective cells in the confusion matrix for the PORTRAIT.

Table 2.3 summarizes the comparison statistics of the three existing systems with our proposed three single hypothesis classification and one ensemble classification approaches. From the results it can be noticed that the metric "precision" for each of the existing three systems is higher as compared to ours. This is because of the fact that the CPC, PORTRAIT and CPAT systems show bias towards predicting a transcript “non-coding” than “coding”. Other than this metric, our system is superior than these. And after analyzing the prediction results of these three existing systems we found that many coding transcripts (1600 out of 4838 false negatives in case of CPAT, 3129 out of 6824 false negatives) with lengths between 400nt to 600nt were falsely predicted as non-coding.

Table 2.3: Comparing classification performance of our system with CPC[20], PORTRAIT[21] and CPAT[22] on the independent benchmark dataset

Systems	**Accuracy**	**Precision**	**Recall**	**F1-score**
CPC[20]	0.627530	0.983141	0.259468	0.410578
PORTRAIT[21]	0.798258	0.843730	0.732005	0.783907
CPAT[22]	0.756402	0.977374	0.524957	0.683044
CNCTDiscriminator-I [a]	0.8311	0.8386	0.8201	0.8292
CNCTDiscriminator-II [b]	0.8254	0.8209	0.8324	0.8266
CNCTDiscriminator-II [c]	0.8028	0.7706	0.8622	0.8138
CNCTDiscriminator-IV [d]	0.8153	0.8985	0.7108	0.7937

Note:
[a] Classifier built with feature categories: composition and ORF
[b] Classifier built with feature categories: composition, ORF and structure
[c] Classifier built with feature categories: composition, ORF, RPKM and structure
[d] Feature specific ensemble based classifier

2.4 Conclusion

The proposed CNCTDiscriminator is a collection of models trained with different features of the transcripts, where each model is capable of discriminating two classes of transcripts. We assessed the performance of each one of these and argued which to pick in practice. We also introduced a feature specific ensemble approach in building a specialized model which outperforms prediction performances of those models that were built only with the individual category of features. Compared with the existing supervised systems – CPC [20], PORTRAIT [21] and CPAT [22] methods, the proposed CNCTDiscriminator system operates more accurately and reliably.

There are several future research directions worth pursuing. The proposed model is a binary classifier, it is only applicable if the given input is a valid RNA transcript from a reference organism's genome. One can impose different strategy to detect the transcripts which are neither coding nor non-coding. Moreover, among the four categories of features, calculating the RPKM scores requires the user to know the coordinate of the input transcripts with respect to the reference genome of the organism. One limitation of our system is that we selected transcripts having length of 1000*nt* or less from the data sources. So, the performance of the CNCTDiscriminator could be suffered while classifying very long transcripts.

2.5 Availability

CNCTDiscriminator source codes, training and test datasets are available at the URL `http://biomecis.uta.edu/~ashis/res/cnctdiscriminator/suppl`

CHAPTER 3

PR2S2Clust: The Patched RNA-seq Read Segments' Structure-oriented Clustering

3.1 Introduction

RNA-seq Read Segments' Clustering Problem: RNA-seq is a revolutionary technology for profiling transcriptomes with which a very precise measurement of expression levels of transcripts and their isoforms can be accomplished [2]. The expression data reveal a vast number of opportunities to investigate the transcriptomic details of an organism that may or may not have a well-studied genome. In most cases the reference genome is available and the analysis pipeline starts by mapping the RNA-seq short-read sequences to the genome, generally using the tool TopHat [33]. Then clusters of overlapping short-reads are grouped into blocks of reads with similar start and end positions using the tool blockbuster [23]. Finally, the overlapping and closely spaced read blocks are further grouped to form read-block groups which we call the read-segments. The RNA-seq read segments are the representative fragment for the constituent raw short-read sequence blocks as introduced in earlier studies [34, 35]. The strategy of considering the read segments rather than blocks of raw short-read sequences greatly reduces the size of the dataset, that allows the application of more computationally expensive algorithms (e.g., pairwise structural distance metric based clustering algorithms, etc.) can be applied as well as preserving structural properties, such as position, length and approximate read start sites and end loci, which otherwise would not have been computationally feasible. By looking at their genomic positions the segments can be labelled. However, clustering the segments might group together segments which share common traits. The commonalities include either sequence level patterns, or structural patterns, or read mapping patterns. We emphasize that our technique will necessarily consider both the sequence and secondary structure patterns of the segments to find meaning clusters of the read-segments. The reason behind this is that RNA structures are responsible for specific biological functions [36, 37]. For example, there are many well-characterized regulatory RNAs in the UTRs of mRNAs that act in *cis* as receivers of other *trans*-acting signals, by forming secondary structures that bind regulatory proteins or small molecular weight ligands [38], whereas mechanisms of miRNA biogenesis were investigated through the corresponding experimentally determined secondary structures [39, 40]. However, as experimentally predicted secondary structures may be expensive and time consuming to get, the Minimum Free Energy (MFE) based secondary structure modeling tools have been applied for the prediction of functional non-coding RNAs [19, 41]. The sequence patterns of the segments have been investigated by earlier studies [34, 35]. Since, we incorporated both sequence and secondary structure patterns of the segments, our method should be efficient in the following manners – (i) it should be able to produce biologically meaningful clustering results, and (ii) computational complexity of the method should be comparable to the existing methods.

Limitations of existing methods: There are several sequence and structure based RNA clustering algorithms available. The LocARNA [42] employs a pairwise local sequence-structure alignment based strategy to cluster RNAs. The FoldAlign [43] uses an energy model and sequence similarity to simultaneously fold and align the given RNA sequences. Both the LocARNA and FoldAlign use variant of Sankoff algorithm for alignment and folding, resulting high computational complexity. Whereas, the GraphClust [44] is an alignment-free method that first encodes the secondary structures in graphs, then extracts kernel features from the graphs and finally leverages the idea of locality sensitive hashing to perform clustering in terms of approximate nearest neighbor queries. The aforementioned methods can only be employed to cluster the segments, but not the read-mapped patched segments. However, the DeepBlockAlign [34] method considers patched processing patterns, but does that at the sequence level of the segments making it impossible to analyze whether structural information at all have any significance on the generated clustering results.

Outline of our results: In this paper, we investigate the significance of using secondary structure information of the patched read-segments in the ncRNA annotation. We propose the annotation of the ncRNAs as a clustering problem using which we tend to know groups of read-segments having similar structural properties. This grouping of the read-segments also provides confidence in the novel ncRNA annotation. We proposed seven pairwise structural distance metrics that take patched RNA-read segments as input. The metrics are (i) Patched Binary Lempel-Ziv complexity distance, (ii) Patched Quaternary Lempel-Ziv complexity distance, (iii) Patched Tree-Edit Distance, (iv) Patched Damerau-Levenshtein string edit distance, (v) Patched Euclidean Distance and (vi) Patched Cosine distance and (vii) Patched Random walker termination based distance. Hierarchical clusterings from each of the distance matrices were performed. Since we have several clustering results for each dataset due to multiple distance matrices and clustering parameters, we employed ensemble strategy to combine the clustering results to come to a consensus partitioning. For this purpose, we applied co-association based [45], majority voting based and Meta-Clustering Algorithm (MCLA) [46] based aggregation strategies to perform the consensus merging of the results. Moreover, we presented a method to represent the read-segments as vectors and applied classical partitional clustering algorithms. Depending on the parameters of the clustering algorithms, we obtained multiple results where again the ensemble clustering strategies were used to retrieve the consensus partitions.

Summary of contributions:

- We define the problem of clustering the patched read-segments through secondary structure perspective (§3.2). We present a platform to tackle the clustering problem (§3.3).
- We proposed several pairwise distance metrics for the patched read-segments based on the structural perspective. Hierarchical clustering algorithms are employed to cluster the segments through the matrices (§3.4).
- We show methods to extract features from the predicted secondary structures of the patched read-segments, so that we can represent the patched read-segments as vectors. classical partitional clustering algorithms are employed to cluster the vectors (§3.5).
- We present ensemble approach to aggregate multiple clustering results that has ability to boost clustering performance (§3.6).
- We present comprehensive experiments on real datasets that show the significance improvement of our proposed methods over the state-of-the-art methods (§3.7).

3.2 Problem Definition

In this section, we start with describing the patched RNA-seq read segments and then define the problem of clustering the segments. We also present clustering performance measures that we will be using in the experiments.

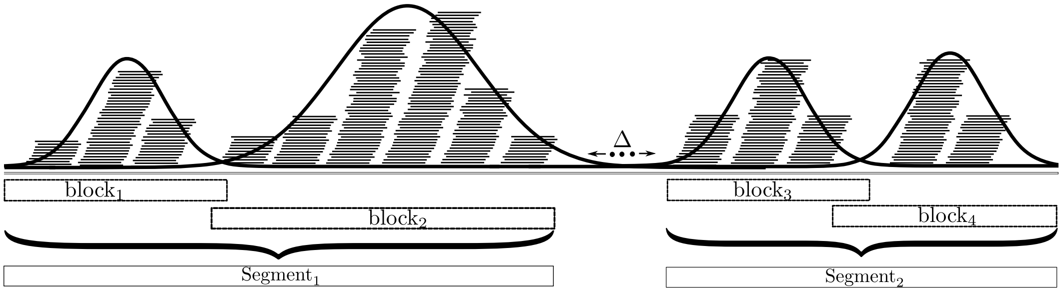


Figure 3.1: RNA-seq read segments.

3.2.1 RNA-seq Read Segments

Expression profile of a given RNA transcript from an RNA-seq experiment is measured in terms of number of short read sequences mapped to the genomic loci of the transcript. The profiles can be compressed by grouping the reads into blocks using the blockbuster [23] tool. The tool performs peak detection on the signal obtained through counting the number of short reads mapped per nucleotide. The signal, across adjacent loci, is then modeled with a mixture of Gaussian distributions. A greedy algorithm is applied to it to extract the reads that belong to the same block, beginning with the largest Gaussian component and removing them in successive iterations. It further assembles a sequence of adjacent blocks into a block-group if the constituent blocks are either overlapping or separated by not more than 30 nucleotides (Δ). The representative nucleotide sequence spanning the first block to the last block will be termed as "segment" sequence throughout this article. Figure 3.1 shows two such segment sequences: $\text{segment}_1, \text{segment}_2$, representing the two block-groups $\{\text{block}_1, \text{block}_2\}$, and $\{\text{block}_3, \text{block}_4\}$ respectively. The two segments are separated by the threshold distance, Δ nucleotides.

3.2.2 Patched RNA-seq Read Segments

An n length segment $\mathbf{s} = (a_1 : f_1, a_2 : f_2, \cdots, a_n : f_n)$ is called the patched representation of the RNA-seq read segment, where a_i is the i^{th} nucleotide position of the segment, and f_i denotes number of short-read sequences overlap at position a_i. Now, if we have stable secondary structure associated with the segment, we can convert the structure into a patched structure leveraging the patched representation of the segment. Figure 3.2 illustrates a patched RNA secondary structure of a certain segment. The structure can also be represented as a weighted undirected graph $G(V, E)$, with n nodes denoting the number of bases in the segment, and m edges forming the secondary structure (both base-pairs and sugar backbone), $c_{ij} = c_{ji} = f > 0$ is the edge weight between nodes i and j referring the number of short-read overlaps between the two.

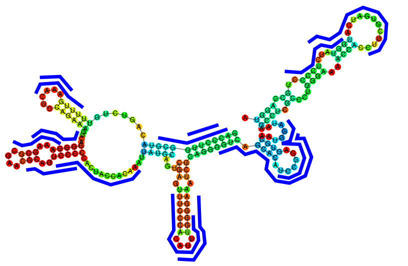


Figure 3.2: Patched representation of a Secondary structure of the representative read segment with the read blocks aligned with the structure (shown in the overlapping blue-lines).

3.2.3 Problem Setup

Let $S = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_N\}$ be a collection of N RNA-seq read segments. We construct the corresponding set P, the patched secondary structures of the read segments from the set S, where $P = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$. Our goal is to resolve a clustering of these N segments in such a way that segments in the same group (called a cluster) are more similar to each other than to those in other groups (clusters) in terms of their secondary structural properties. We adapt three categories of clustering: hierarchical clustering, partitional clustering and ensemble clustering.

Definition 1. *A cluster K is defined as any subset of P. A collection of clusters H is called a* ***hierarchical clustering*** *if $\cup_{K_i \in H} K_i = P$ and for any $K_i, K_j \in H$, only one of the following is true (i) $K_i \subset K_j$, (ii) $K_j \subset K_i$, (iii) $K_i \cap K_j = \emptyset$*

The hierarchical clustering H forms a tree, where each internal node corresponds to a particular cluster. Let $D = \{d_{i,j}\}$ denotes the collection of all pairwise similarities (or dissimilarities) between the segments in P, with $d_{i,j}$ denoting the similarity score (or dissimilarity score) between $\mathbf{p}_i$ and $\mathbf{p}_j$ assuming $d_{i,j} = d_{j,i}$. The hierarchical clustering algorithm requires the complete D matrix to be able to identify the clustering result H.

Definition 2. *A collection of cluster T is called a* ***partitional clustering*** *if $\cup_{K_i \in T} K_i = P$ and $K_i \cap K_j = \emptyset$ holds for any i, j.*

Most of the partitional clustering algorithms optimize a criterion function, minimize the intra-cluster sum of squares distances between the segments for example, to compute the partitioning. The hierarchical clustering H can be converted into a partitional clustering through using a cut at specified height of the tree.

Definition 3. *Given a set of t partitional clustering results $\Pi = \{\pi_1, \pi_2, \cdots, \pi_t\}$ of the set of N patched segments $P = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$, where $\pi_i = \{\phi_i(\mathbf{p}_1), \phi_i(\mathbf{p}_1), \cdots, \phi_i(\mathbf{p}_N)\}$, and $\phi_i(\cdot)$ denotes a cluster number, $x \in \{1, 2, \cdots, C\}$, assigned by the i^{th} clustering result, and C being the total number of clusters in each of the clustering result. Now, a collection of cluster Γ is called a* ***consensus (i.e., ensemble) clustering*** *of the set Π if $\cup_{\gamma_i \in \Gamma} \gamma_i = \Gamma$ and $\gamma_i \cap \gamma_j = \emptyset$ holds for any pair of i, j,* *and* $\Gamma = \Psi(\Pi)$*, where* $\Psi(\cdot)$ *being the consensus function that combines the* t *separate partitioning results and computes the single final cluster.*

Among many cluster aggregation strategies (i.e., the $\Psi(\cdot)$ functions), we adapted the three state-of-the-art strategies, namely co-association matrix, meta-clustering and majority voting. In this manuscript, we present a comparative study of the consensus clustering over the individual (either hierarchical or partitional) clustering algorithms.

3.2.4 Performance Measures

Given a set of N segments $G = \{S_1, \cdots, S_N\}$ and suppose $\mathbf{X} = \{X_1, \cdots, X_M\}$, and $\mathbf{Y} = \{Y_1, \cdots, Y_P\}$ be the two partitioning results of the N segments in G, where $\cup_{i=1}^{M} X_i = G = \cup_{j=1}^{P} Y_j$ and $x_i \cap X_j = \varnothing = Y_k \cap Y_l$ for $1 \leq i \neq j \leq M$ and $1 \leq j \neq k \leq P$. The information on the overlap between the two partitions $\mathbf{X}$ and $\mathbf{Y}$ can be summarized in form of a $M \times P$ contingency table $C = [n_{ij}]_{j=1 \cdots P}^{i=1 \cdots M}$ as illustrated in Table 3.1, and n_{ij} denotes the number of segments that are common to partition X_i and X_j. The $\binom{N}{2}$ segment pairs in G can be classified into four types – N_{11}: the number of pairs that are in the same cluster in both $\mathbf{X}$ and $\mathbf{Y}$; N_{00}: the number of pairs that are in different clusters in both $\mathbf{X}$ and $\mathbf{Y}$; N_{01}: the number of pairs that are in the same cluster in $\mathbf{X}$ but in different cluster in $\mathbf{Y}$; and N_{10}: the number of pairs that are in different cluster in $\mathbf{X}$ but in the same cluster in $\mathbf{Y}$. All of these four values can be computed using the n_{ij}'s from the contingency table (Table 3.1). Since, the dataset we will be using is associated with ground-truth labels, we will perform two external validations to assess one partitioning result (say, $\mathbf{X}$) with the ground truth partition ($\mathbf{Y}$). The validations are namely Adjusted Rand Index (ARI)

Table 3.1: The notations for the contingency table for comparing two partitions X and Y.

$\mathbf{X} \backslash \mathbf{Y}$	Y_1	Y_2	$\cdots$	Y_P	Sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1P}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2P}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_M	n_{M1}	n_{M2}	$\cdots$	n_{MP}	a_M
Sums	b_1	b_2	$\cdots$	b_P	$\sum_{ij} n_{ij} = N$

[47] and Normalized Mutual Information (NMI) [46], which are defined in Equations 3.1 and 3.2.

$$ARI(\mathbf{X}, \mathbf{Y}) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})} \tag{3.1}$$

$$\begin{aligned} NMI(\mathbf{X}, \mathbf{Y}) &= \frac{I(\mathbf{X}, \mathbf{Y})}{\sqrt{H(\mathbf{X})H(\mathbf{Y})}}, \text{ where: } I(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{M}\sum_{j=1}^{P} \frac{n_{ij}}{N} \log \frac{n_{ij}/N}{a_i b_j / N^2} \\ &\text{and, } H(\mathbf{X}) = -\sum_{i=1}^{M} \frac{a_i}{N} \log \frac{a_i}{N}, H(\mathbf{Y}) = -\sum_{j=1}^{P} \frac{b_j}{N} \log \frac{b_j}{N} \end{aligned} \tag{3.2}$$

The ARI is bounded by 1 and equals to 0 only when the Rand Index (RI), that is $(N_{00} + N_{11})/\binom{N}{2} = \mathbb{E}[RI]$ which happens if X and Y partitions are picked at random. The NMI is also bounded by 1 and equals 0 when the two partitions are truly random.

Besides the external validations, we will also use one internal quality metric namely Sum of Squared Error (SSE) that measure the intra-cluster homogeneity and the inter-cluster separability while only considering the data itself without external ground true labels, and is defined in Equation 3.3.

$$SSE(\mathbf{X}) = \sum_{i=1}^{M} \sum_{v \in X_i} ||v_i - \mu_i||^2, \tag{3.3}$$

where X_i is the set of items in the cluster i; μ_i is the mean vector of cluster i.

3.3 Overview of PR2S2Clust System

This section overviews PR2S2Clust, our system for clustering of the patched secondary structure of the representative patched RNA-seq read segments. We start by presenting the key idea of the PR2S2Clust: how patched secondary structures can be clustered through introduction of various distance measures and applications of partitional, hierarchical and the consensus clustering strategies to reliably partition the segments. While §3.4, §3.5, and §3.6 describe each of the parts in detail, we discuss at the end of this section how we prototyped the PR2S2Clust system over real-world RNA-seq experiment datasets for its evaluation.

3.3.1 System Architecture

Figure 3.3 illustrates architecture for PR2S2Clust which has two three main steps: (1) Pre-processing that builds pairwise distance matrices to be used by the hierarchical clustering algorithms, and the vectorized representation of the structures that will suitably be applied to any partitional clustering algorithm, (2) Clustering of the segments according to the inputs from the previous step and (3) Clustering aggregation of all the partitions obtained in the previous step. The system works as follows:

- **Step 1, Pre-processing:** It receives as input a set of patched segment sequences (as defined in §3.2). RNA Secondary structures of the sequences are predicted through using the-state-of-the-art tools. Then, the patched version of the structures are prepared. We introduce several pairwise structural distance measures that takes the patched representation into consideration. The corresponding distance matrix for each of the measure is prepared for the next step of hierarchical clustering. Additionally, we also introduce feature extraction

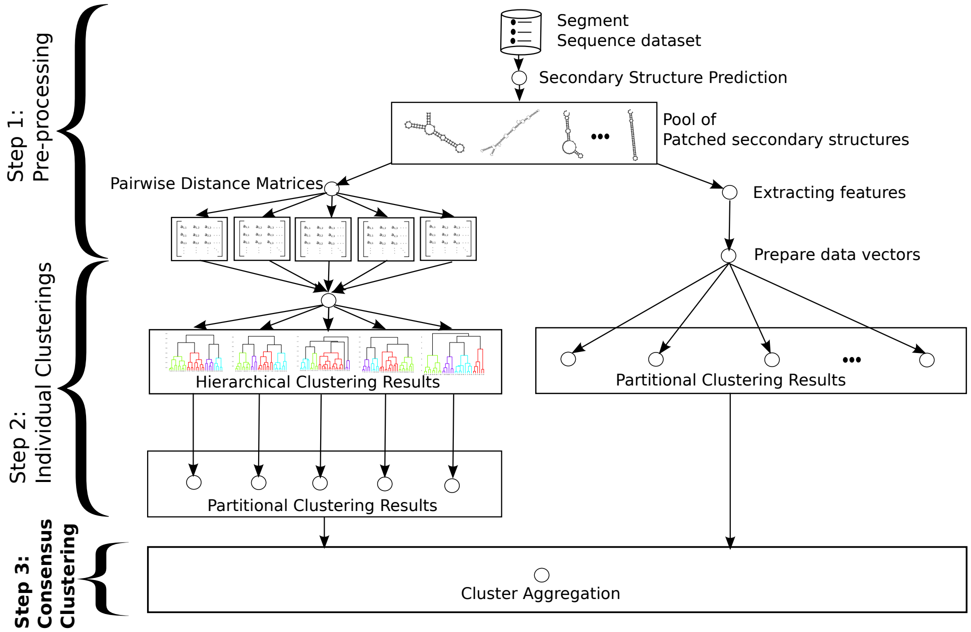

Step 1:
Pre-processing
Segment
Sequence dataset
Secondary Structure Prediction
Pool of
Patched seccondary structures
Pairwise Distance Matrices
Extracting features
Prepare data vectors
Step 2:
Individual Clusterings
Hierarchical Clustering Results
Partitional Clustering Results
Partitional Clustering Results
Step 3:
Consensus
Clustering
Cluster Aggregation

Figure 3.3: System architecture of the PR2S2Clust.

processes to convert the patched secondary structures into feature vectors that will be suitable for use as input for the partitional clustering algorithm.

- **Step 2, Individual Clusterings:** Given a distance matrix, we perform hierarchical clustering. Later, we cut the generated dendrograms to retrieve a linear partitional clustering results. Again, the feature vectors prepared in the previous step are also fed into a partitional clustering algorithm to generate clustering results. All of these clustering results are individually validated for their performance and reported in §3.7.
- **Step 3, Consensus Clustering:** Given, all the different clustering results of the segments obtained from the different clustering methods in the previous step, in this step we combine all the individual results to get a single consensus clustering result (§3.6). This is due to the fact that sometimes individual clustering results may lead to incorrect and unstable partitioning of the data points,

and aggregating the clustering results overcomes the issue. We extended our system to addressed this issue in this step. We presented three cluster aggregation strategies and reported a comparative analysis of the techniques along with the individual clustering methods.

3.3.2 Prototype Design for the Morin_EB dataset

Before delving into the detail design of PR2S2Clust in §3.4, §3.5 and §3.6, we would like to briefly discuss how we prototyped over Morin_EB dataset [48], a representative dataset used by deepBlockAlign [34] method that we contrasted with our system. Note that while we focus the rest of the manuscript on this particular dataset, the adaptation to other RNA-seq datasets is pretty straightforward; e.g., we present experiments with two additional datasets – GSM450598 [49] and GSM450605 [49] in §3.7.2.2.

We retrieve the segment sequences from the dataset through application of the blockbuster tool [23]. The sequences along with the frequency of short-reads mapped on the entire lengths (i.e., their patched representations) are then used to predict corresponding secondary structures. Two predictor algorithms were used, namely Vienna RNAfold [4] and CentroidFold [50]. Several distinctive features were extracted from the predicted structures, and pair-wise distance matrices were prepared according to our proposed distance criteria. The feature-represented structure vectors are used in k-means clustering algorithm, and the distance matrices are applied to hierarchical agglomerative clustering algorithm. Finally, the results are aggregated.

We used the same ground-truth labels for each of the segments as the deepBlockAlign [34] project. Originally, the labels are determined based on the genomic positions of the segments and pre-existing annotations of the region encompassed by the segments.

3.4 Hierarchical Clustering of the Segments

The key to hierarchical clustering is the pairwise distance matrix among the read-segments. Here, we first define several pairwise structural distance metrics, with which we can build the distance matrices. We then perform hierarchical clustering on each of the distance matrix, and aggregate the results. By applying the pairwise structural distance measures, we can define one pairwise distance matrix per measure. Then we perform the hierarchical clustering with complete linkage of the distance matrices, and applied a cut to obtain k number of clusterings. We report each of the individual clustering performances using external evaluation criteria – Adjusted Rand Index and Normalized Mutual Information (NMI), and the internal evaluation criterion – Sum of Squared Error (SSE).

3.4.1 LZ-Complexity based Distance Criterion

The Binary LZ-complexity distance [10] is the direct extension to the very well known Lempel-Ziv (LZ) sequence comparing algorithm [51]. Since the LZ complexity is applicable only to finite linear sequences, the secondary structures denoted by dot-bracket notation of the RNA segments are first required to convert to a dot-plot matrix, from which a corresponding binary sequence can be extracted. A dot plot is a two-dimensional graph in which there is a dot (or symbol “1”) at position (i,j) if a base at position i pairs with the base at position j in the secondary structure, otherwise there is no dot present in the plot (denoted by symbol “0” in the matrix). Fig. 3.4 shows both the predicted secondary structure and corresponding dot plot representation of the read block segment 405 of the morin_EB dataset. In the dot plot, if scanned diagonally from left to right downward fashion and stopping at the symmetric border line and re-scan from the next column or row, we will get a binary sequence of 0s and 1s. In the binary sequence a block of consecutive 1s represent a stem of the secondary structure and block of consecutive 0s between two stems represent loop. We further replaced each block of 0s by a single “0” for simplicity. Thus, the characteristic binary sequence for the structure of segment 405 is “0111101011111110”.

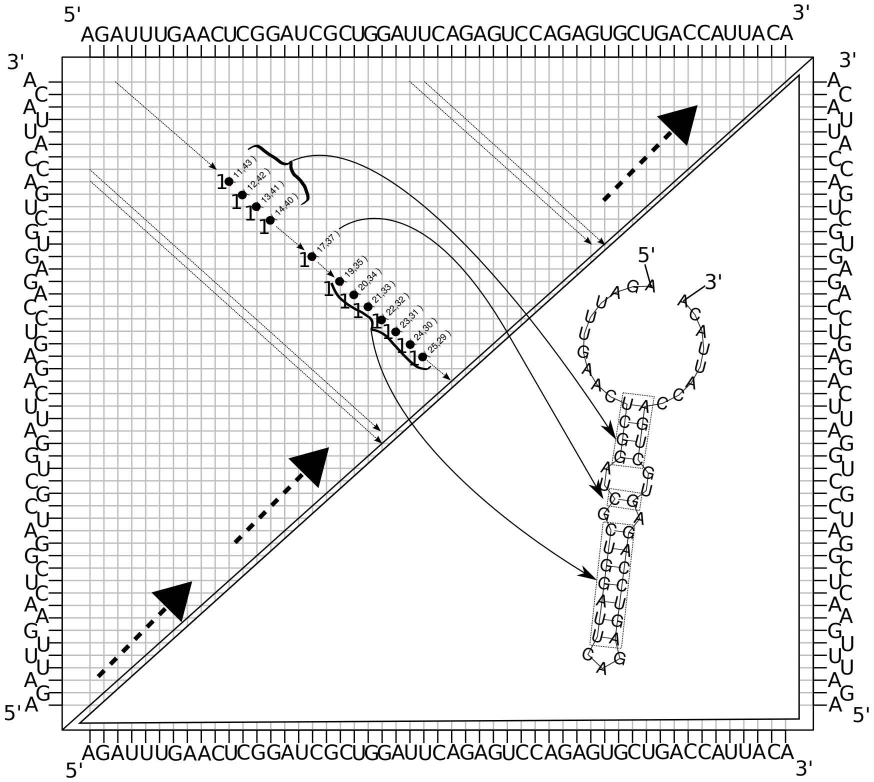


Figure 3.4: Dot plot representation of the secondary structure of the read block segment (ID: 405). The lower right triangle contains the secondary structure and the upper left triangle is its dot plot representation. The mapping of the stems (consecutive base-pairs in the structure) are shown using the arrows from the dot plot to the secondary structure plot. The scanning direction starts from the lower left part of the upper triangle to its upper right part (shown as the dotted triangular arrow heads).

After the conversion of the RNA secondary structure into the binary sequence, the LZ-complexity of the sequence can be computed. There are several distance measures between two linear sequences S and Q based on the LZ-complexities of the S and Q sequences [52]. In our study we used the following normalized distance function $d(S, Q)$:

$$d(S,Q) = \begin{cases} \frac{c(SQ)-c(S)+c(QS)-c(Q)}{\frac{1}{2}\left[c(SQ)+c(QS)\right]} & : \text{ if } Q \neq S \\ 0 & : \text{ otherwise} \end{cases}$$

Here, the function $c(\cdot)$ returns the LZ-complexity of a given linear sequence, SQ is a new sequence when Q is appended to the sequence S. The LZ complexity, $c(\cdot)$ of a finite sequence is related to the number of steps required by a production process that builds the original sequence. Let S, Q and R be sequences defined over an alphabet Σ, where $|S|$ denotes number of symbols in sequence S, $S[i]$ denotes the i^{th} symbol of sequence S and $S[i : j]$ denotes the substring of S composed of the elements of S between index i and j (inclusive). An extension $R = SQ$ of S is reproducible from S (denoted $S \rightarrow R$) if there exists an integer $p \leq |S|$ such that $Q[k] = R[p + k - 1]$ for $k = 1, \ldots, |Q|$. That is, R can be obtained from S by copying elements from the p^{th} location in S to the end of S. As each copy extends the length of the new sequence beyond $|S|$, the number of symbols copied can be greater than $|S| - p + 1$. Thus, this is a simple copy process of S starting from position p, which can carry over to the added part, Q.

A sequence S is producible from its prefix $S[1 : j]$, which is denoted by $S[1 : j] \Rightarrow S$, if $S[1 : j] \rightarrow S[1 : |S| - 1]$. Thus, the production allows an extra different symbol at the end of copy process which is not permitted in reproduction. Any sequence S can be built using a production process where at its i^{th} step $S[1 : h_{i-1}] \Rightarrow S[1 : h_i]$, assuming an empty symbol produces the first symbol of S. An m-step production process of S results in a parsing of S in which $H(S) = S[1 : h_1] \cdot S[h_1 + 1 : h_2] \cdot \ldots \cdot S[h_{m-1} + 1 : h_m]$ is called the history of S and $H_i(S) = S[h_{i-1} + 1 : h_i]$ is called the i^{th} component of $H(S)$. If $S[1 : h_i]$ is not reproducible from $S[1 : h_{i-1}]$, then $H_i(S)$ is called the exhaustive history. A history is called exhaustive if each of its components (except possibly the last one) is exhaustive. Let $c_H(S)$ be the number of components in the history of S. Then the LZ complexity of S is $c(S) = min\{c_H(S)\}$ over all histories of S. It can be shown that $c(S) = c_E(S)$, where $c_E(S)$ is the number of components in the exhaustive history of S

The Binary LZ complexity based distance measure does not consider base compositions into account in the stem sites, that is, it treats characteristic sequences of AU or UA, GC or CG, GU or UG pairs without their order of occurrences. In Quaternary LZ complexity, the order of the base-pair compositions is taken into consideration. The dot plot from the secondary structure is prepared in much the same way as in the binary case, except that in the $(i, j)^{th}$ cell a 1 is assigned if (i, j) base pair is a AU or UA, a 2 is assigned if it is a GC or a CG base pair, a 3 is assigned if it is a GU or a UG base pair, and otherwise 0 to represent a no base pair. Then the characteristic sequence out of the dot plot is extracted as before, and LZ-complexity algorithm is applied to it to deduce the pairwise normalized distance score between two RNA structures.

3.4.2 Tree Edit Distance Criterion

SimTree is an application available at the url `http://bioinfo.cs.technion.ac.il/SimTree/` (Last accessed: 12-07-2016 9:13AM) for computing and analyzing the similarity between two RNA secondary structures. The method transforms the two RNA secondary structures into labeled trees and then computes the distance between the two trees resulting in a similarity score. The values of the score range between 0 and the size of the smaller tree. However, the raw similarity score can be normalized to ignore the effect of tree sizes. The normalized score is between 0 and

1, where 1 denotes a perfect match. In our problem, SimTree normalized similarity score can be used to derive a pairwise structure distance, $d(P,Q)$ for the two RNA secondary structures P and Q:

$$d(P,Q) = 1 - \text{simTree\_}score(P,Q). \tag{3.4}$$

3.4.3 Damerau-Levenshtein String-Edit Distance Criterion

The Damerau-Levenshtein distance is a distance [53] between two strings denoting secondary structures of two read-segments represented in dot-bracket notation. The distance is given by counting the minimum number of operations required to transform one string into the other, where an operation is defined as an insertion, deletion or substitution of a single character, or a transposition of two adjacent characters. The distance between two strings a and b is given by $d_{a,b}(|a|,|b|)$ where:

$$d_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{case 1} \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \\ d_{a,b}(i-2,j-2)+1 \end{cases} & \text{case 2} \\ \min \begin{cases} d_{a,b}(i-1,j)+1 \\ d_{a,b}(i,j-1)+1 \\ d_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases}$$

where the cases are:

- Case 1: $\text{if}\min(i,j) = 0$
- Case 2: if $i,j > 1$ and $a_i = b_{j-1}$ and $a_{i-1} = b_j$

3.4.4 Euclidean Distance Criterion

The distance between two secondary structure vectors $\vec{u}$ and $\vec{v}$ is given by the following equation:

$$d(\vec{u},\vec{v}) = \frac{1}{2}\frac{||(\vec{u}-\mu_{\vec{u}})-(\vec{v}-\mu_{\vec{v}})||^2}{||\vec{u}-\mu_{\vec{u}}||^2+||\vec{v}-\mu_{\vec{v}}||^2}$$

where the $\mu_{\vec{a}}$ is the mean of $\vec{a}$. The distance value of 1 denotes the most distant vectors, where a value close to 0 denotes the two vectors essentially are the same. Preparation of each of the vectors given the predicted secondary structures are discussed in §3.5.1.

3.4.5 Cosine Distance Criterion

The cosine distance between two structure vectors $\vec{u}$ and $\vec{v}$ is given by the following equation:

$$d(\vec{u},\vec{v}) = 1 - \frac{\vec{u}\cdot\vec{v}}{||\vec{u}||||\vec{v}||}$$

The distance measure is also bounded in the range [0,1].

3.4.6 Random Walker Termination based Distance Criterion

Consider a random walker on the patched secondary structure graph G walks from node i to node j according to the transition probability p_{ij} and pays the cost c_{ij}. Fang et al. [54] developed k-Random Walker Termination algorithm (k-RWT) for undirected and unweighted graph. Based on an intuition from the thermodynamic system, the authors set k random walkers with a fixed initial energy for a walk on the graph, and by walking they lose their energy, and eventually stop when the energy becomes zero.

```
COMPUTE_RWTWG($G, k, \phi$)
// $G(V, E)$: input graph, $k$: number of iterations,
// and $\phi$: sink threshold
for $i = 1$ to $V.length$
    // assigning number of random walkers on each node
    $walkers[i] = i.degree$
    $walkers\_next[i] = 0$
$total\_walkers = \text{SUM}(walkers)$
while $k > 0$
    for $i = 1$ to $V.length$
        for each walker on node $i$
            for each $j$ be the neighbor of node $i$
                // transition probability
                $p(i, j) = \frac{w(i,j)}{\sum_a w(i,a)}$
                $j$ = pick a neighbor node with probability $p(i, j)$
                if $walkers[j] > \phi$
                    $walkers\_next[j] = walkers\_next[j] + 1$
                else $total\_walkers = total\_walkers - 1$
    ZERO($walkers$)
    SWAP($walkers, walkers\_next$)
    $rate$ =termination rate of walkers in this iteration
    $RWT.append(rate)$
    $k = k - 1$
return $RWT$
```

This way, just as the cooling process of two objects with similar shape have similar heat transfer pattern, thereby reveal the fact that the random walker termination patterns should be similar between the two graphs if they are structurally similar.

We need to extend the k-RWT algorithm for a general setup, that is, on weighted undirected graphs. Then, the modified k-Random Walker Termination algorithm on a graph of patched RNA Secondary Structure would get us a score, and scores of several structures can be compared to identify similarities among structures. The similarity scores can then be employed to perform hierarchical clustering. We have implemented Random Walker based termination algorithm on patched (i.e., weighted) RNA secondary structure graph. Algorithm `Compute_RWTWG` returns the time series rate vector given a patched RNA secondary structure graph. In the algorithm we computed the transition probabilities from the edge-weights. The k element time series rate vector RWT is considered as a representative signature of each RNA secondary structure graph.

Parameter Selection k and ϕ: Since each node are assigned number of walkers equal to its degree, and we see that in a RNA secondary structure graph a node can not have more than degree 3, we can tune ϕ from 0 to 3. Sink threshold close to the degree of the nodes will lead to early walker terminations, whereas threshold close to 0 will keep the walker running for a while. Empirically we found that all the walkers die after about 15-20 iterations. Thus, $k = 20$ would be a reasonable choice. Applying the algorithm on all the patched transcripts of the dataset of segments, we retrieved RWT signatures, and computed the pairwise alignments.

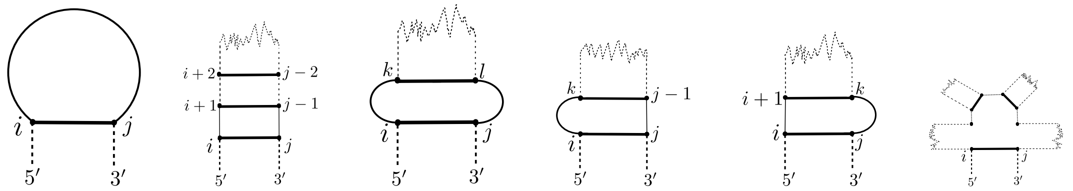
Then, we measure pairwise distance between two graphs (i.e., patched segments) using Equation 3.5.

$$RWT_{score}(U, V) = \sqrt{\sum_{i=1}^{k}(u_i - v_i)^2} \tag{3.5}$$

3.5 Partitional Clustering of the Segments

In order to apply a partitional clustering algorithm (k-means for instance), we are required to transform the secondary structure of the segment into vector of features. We applied k-means algorithm on the vectors containing the read-segment vectors. We generated clustering results by running several runs of k-means with different seeds each time with the distance functions: (1) squared Euclidean distance, (2) cityblock distance, (3) cosine distance, (4) correlation distance and (5) hamming distance. In this section, we describe the process of feature extraction from the structural segments.

3.5.1 Feature Representation of the Segments



(a) Hairpin Loop (b) Stem (c) Interior Loop (d) Left Bulge Loop (e) Right Bulge Loop (f) Multi-Loop

Figure 3.5: Structure elements occurred in a RNA secondary structure

Most RNA molecules are single stranded that fold back onto itself to form double helical regions stabilized by the Watson-Crick base pairs (A-U and C-G) and wobbling base pair (G-U).

Given a primary structure, i.e., a sequence S, a secondary structure is defined as a set $P = \{(i, j)|1 \leq i < j \leq n\}$ of base-pairs represented as tuples of positions in the sequence of length n such that for any two base-pairs $(i_1, i_2), (j_1, j_2) \in P$ with $i_1 < j_1$ either

1. $i_1 < i_2 < j_1 < j_2$ or,

2. $i_1 < j_1 < j_2 < i_2$

The two conditions imply that a base participates in at-most one base-pair. The tuple (S, P) describes an RNA as a sequence of nucleotides provided with a secondary structure formed by base-pairs. A secondary structure can be drawn in a plane such that base-pairs are designated by arcs whose ends connect the two bonded bases, and all arcs can be drawn in one half-plane such that they do not cross.

The base-paired structure formed by the Watson-Crick base-pairs A-U and C-G and the wobbling base-pair G-U can be divided into loops, also known as structure elements. A loop is a formation of a base-pair (i, j) that encloses a chain nucleotides or other base-pairs. A free energy contribution can be assigned to each loop. The method commonly used for the energy calculation of a complete secondary structure is based on the nearest neighbor model in which the thermodynamic stability of a base-pair is dependent on the adjacent base-pairs. The loops are assumed to contribute additively to the overall free energy of the secondary structure.

Figure 3.5 demonstrates the common structural elements of secondary structures. If all the internal nucleotides in the sequence interval $[i + 1, \cdots, j - 1]$ with base-pair (i, j) are contiguous and non-binding, then we call this element a hairpin. If the base-pair (i, j) is adjacent to another base-pair (k, l) such that $i < k < l < j$, then various structure element formations are possible:

1. if $k > i + 1$ and $j = l + 1$, then we call this structure element a left bulge;
2. if $k = i + 1$ and $j > l + 1$, then it is a right bulge;
3. if $k > i + 1$ and $j > l + 1$, then it is an internal loop;
4. if $k = i + 1$ and $j = l + 1$, then it is a stem (or a stack).

A multi-loop consists in addition to the base-pair (i, j) of at least two base-pairs from which several stems radiate.

Minimum Free Energy (MFE) based RNA secondary structures can be predicted using the well known Vienna RNA package [4]. However, it was later discovered that the non-coding RNA sequences do not always form MFE based secondary structures that may lead to a wrong predictions of the structure by Vienna RNA package. We, therefore, in conjunction with the Vienna RNA package, applied a non-MFE based secondary structure predictor, CentroidFold [50]. The later package applied a posterior decoding method including the γ-centroid estimator that can provide more reliable RNA structures, especially non-coding RNA structures.

Both the Vienna RNA and CentroidFold packages provide a dot-bracket notation of the predicted secondary structures. From the results we first extracted the following 10 features: (1) $n(AU)$, the total number of A-U, or U-A base pairs (Watson & Crick pair) present in the predicted structure. The frequency is normalized by total number of base pairs present in the structure. (2) $n(CG)$, the normalized number of C-G or G-C base pairs (Watson & Crick pair) present in the structure, (3) $n(GU)$, the normalized number of G-U or U-G base pairs (Wobbling pair) present in the structure, (4) $n(BP)$, base pair ratio of total number pairs present in the predicted structure to the length of the sequence, (5) $n(HP)$, average number of bases forming a hairpin, (6) $n(LB)$, average number of bases forming a left bulge, (7) $n(RB)$, average number of bases forming a right bulge, (8) $n(IL)$, average number of bases forming an interior loop, (9) $n(Stem)$, average number of base-pairs forming a stem, (10) $n(ML)$, average number of bases forming a multi-loop.

However, the Vienna RNA package also reports the MFE of the predicted secondary structure that we also consider as a feature associated with the Vienna RNA prediction. Thus, a total of twenty-one structural features form a twenty-one dimensional data vector per predicted structure. The $21 \times N$ data matrix containing the 21 features of the N read-segment vectors were prepared. Figure 3.6 illustrates two examples of the secondary structure profile extraction of the RNA-seq read block groups.

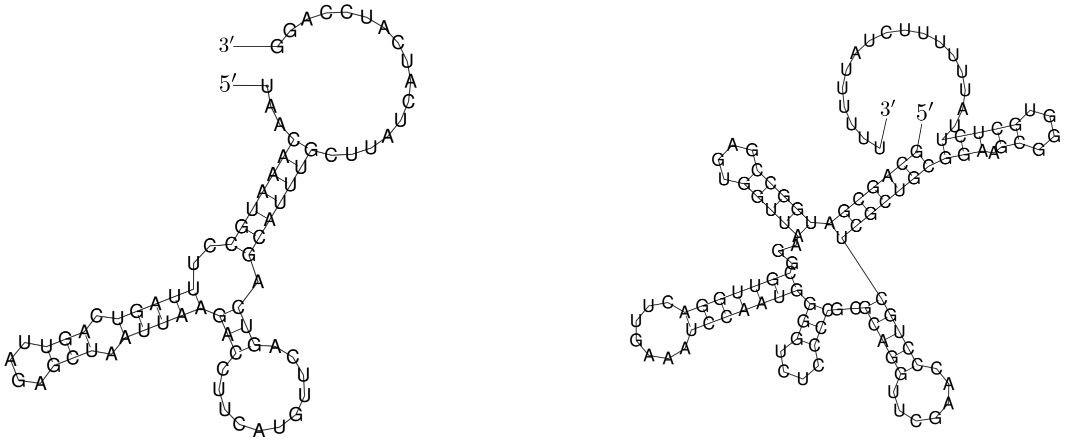


Figure 3.6: Secondary Structures of the two read block segments (segment id 43 on the left and 608 on the right) from the morin_EB dataset (predicted by VienaRNA package) . Here, for the read-segment id 43 (left): (i) average hairpin length is $\frac{4+11}{2} = 7.5$, (ii) average left and right bulges length are 0 and 0 respectively since there is none present in the structure, (iii) average interior loop length is $\frac{2}{1} = 2$, (iv) average length of stems is $\frac{7+3+5+3}{4} = 4.5$, (v) average size of the multi-loop is $\frac{9}{1} = 9$. And for id 608 (right): (i) average hairpin length is $\frac{4+7+4+7+4}{5} = 5.2$, (ii) average left and right bulge length is $\frac{1+1}{2} = 1$ and 0 respectively, (iii) average interior loop length is 0, since none is present, (iv) average length of stems is $\frac{2+3+7+5+7+3+5}{7} = 4.57$, (iv) average size of multi-loop is $\frac{15}{1} = 15$

3.6 Ensemble Clustering of the Segments

Recently, the ensemble clustering has emerged as an important extension to the classical clustering problem since it shows potential to overcome the clustering instability imposed by the classical approaches and thereby improve performance. In this section, we briefly summarize the three ensemble clustering approaches that aggregates various partitional clustering results of the same number of segments, and draw conclusion by providing us a consensus result.

3.6.1 Co-association matrix based cluster aggregation

Combining clustering results (i.e., evidences) using the co-association matrix employed a voting mechanism [45]. The key intuition behind this aggregation strategy is that read-segments which are similar are very likely to be co-located in the same cluster. Thus, the co-occurrences of the pairs of read-segments in the same cluster can be considered as votes for their association. Thus the n data partitions of N read-segments are mapped into an $N \times N$ co-association matrix C:

$$C(i, j) = \frac{n_{ij}}{n}, \tag{3.6}$$

where N_{ij} is the number of times the segment pair (i, j) is assigned to the same cluster among the n partitions. This corresponds to a non-linear transformation of the original feature space into a new representation, summarized in the similarity matrix, C, induced by inter-segment relationships present in the clustering ensemble [45]. Now by applying a hierarchical clustering on the new similarity matrix C, a consistent partition can be retrieved.

3.6.2 Meta-Clustering Algorithm (MCLA) for cluster aggregation

The approach is based on clustering the clusters [46]. The first step of the clustering algorithm is to form a meta-graph with a vertex for each base cluster. The edge weights of the graph are proportional to the similarity between the vertices, computed using the binary Jaccard measure:

$$W_{i,j} = \frac{|c_i \cap c_j|}{|c_i \cup c_j|}$$

where c_i and c_j are the two clusters. This similarity matrix W can be treated as a graph with clusters as nodes. And this graph is partitioned into meta-clusters using the METIS algorithm. Then, all the clusters in each meta-cluster are collapsed to yield an association vector for the meta-cluster. The vector is computed by averaging the association instances to each of the constituent clusters of the corresponding meta-cluster. The instance is then clustered into the meta-cluster that it is associated to.

3.6.3 Majority Voting based cluster aggregation

Majority voting is an intuitive cluster aggregation method. It chooses the cluster for a data point that is chosen by the majority of the independent clustering results [55, 56]. Given a set of data points, every clustering algorithm tries to minimize the total intra-cluster distances and maximize total inter-cluster distances among the data points. However, the choice of optimization criterion, that is, the distance measures used for a particular clustering algorithm may lead to unstable partitioning of the data points if the distribution of the points is totally unknown. In such cases, it may not be obvious which criterion to use to obtain stable clustering results. Majority voting based clustering aggregation method can help solve this issue. With this method the output of several single clustering algorithms can be combined to reduce the variance of the error between the different results and to get an overall decision made by the combined clustering algorithms.

Assume that we have T independent clustering results of N data points, each having C number of partitions. Now, for a given data point x, we first prepare a decision profile matrix $D_x \in \{0,1\}^{T\times C}$, where $D_x(t,i) = 1$ if the t^{th} clustering results chooses the i^{th} cluster for the data point x, otherwise $D_x(t,i)$ is set to 0. Majority voting result in an ensemble decision for cluster i of a data point x would be formulated as in Equation 3.7.

$$\text{Cluster assignment of the data point } x = \underset{i}{max}\left\{\sum_{t=1}^{T} D_x(t,i)\right\} \tag{3.7}$$

As can be seen in Equation 3.7, if there is no majority, the tie is broken by assigning the data point x to the first cluster, which would be the assignment number returned by the *max* function.

The technique was successfully applied to characterize the event related potentials of the EEG signals that help in the early diagnosis of the Alzheimer's disease [57]. It was also reported to solve problems from molecular biology. It was applied to recognize, given a sequence of DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA that are spliced out) [56].

3.7 Experimental Evaluation

In this section, we present details of the datasets, programming platforms, and the experiments along with the results.

3.7.1 Experimental Setup

Hardware and Platform: All our experiments were conducted on a computer with Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz with 3MB of L2-cache, 8GB of RAM, 500GB of SATA hard-drive with 5400RPM. The algorithms were implemented in R programming language (version 3.2.0) in R-Studio development environment installed in a 64-bit Ubuntu 14.04 LTS operating system.

Dataset: We considered the using the Illumina sequenced 15 days old human Embroyoid cell dataset [48] (morin_EB). Additionally we considered two other RNA-seq datasets: GSM450598 and GSM450605, which are Illumina sequenced dissected 34 days and 5105 days old human post-mortem superior frontal gyrus tissue sample datasets respectively [49].

Table 3.2: The dataset containing the read block groups used in our study.

Dataset name	Number of RNA-seq read segments per category of ncRNAs									
	miRNA	tRNA	rRNA	scRNA	snoRNA CD	snoRNA HACA	snoRNA scaRNA	snRNA	un-labeled	Total (Σ)
morin_EB [48]	193	157	24	7	42	4	1	9	18	455
GSM450598 [49]	193	74	28	13	30	2	1	6	30	377
GSM450605 [49]	194	161	35	56	56	16	3	52	113	686

Ground Truth: We retrieved the annotations of RNA-seq segments of the dataset from the website of deepBlockAlign (`http://rth.dk/resources/dba/supplementary.php` (Last accessed: May-20-2015)) [34]. We list the labels in Table 3.2, where we can see that the morin_EB dataset contains 455 segments representing eight different categories of non-coding RNAs (ncRNAs). Statistics of the additional two datasets are also listed in the table.

Algorithms Evaluation: We evaluated hierarchical clustering approaches applied on the distance criteria we proposed in §3.4 in terms of Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). We also evaluated ensemble partitional clustering approaches in terms of ARI, NMI and SSE. We compared four algorithms with our ensemble based approach, which are (1) LocARNA [42], (2) FoldAlign [43], (3) GraphClust [44], (4) DeepBlockAlign [34].

3.7.2 Experimental Results

3.7.2.1 Hierarchical Clustering Results of the Segments

According to the flow diagram illustrated in Figure 3.3, we first obtained the secondary structures of all the segments of the dataset (§3.5.1). Then we prepared seven pairwise distance matrices for the dataset (§3.4.1, §3.4.1, §3.4.2, §3.4.3, §3.4.4, §3.4.5, §3.4.6). The CPU time requirement for preparing each of the matrix is illustrated in Table 3.3. It is clear that the Euclidean distance, Damerau-Levenshtein string edit distance and the Cosine distance calculations are the fastest because of their linear computational complexity, whereas the implementations of the preparation of the other three distance matrices quite complicated, and each runs in quadratic order.

Table 3.3: CPU time (in seconds) required to prepare each of the pairwise distance matrix for the Morin_EB dataset.

Distance Metric Name	**CPU Time**
1. Binary LZ distance	23125.80
2. Quaternary LZ distance	20621.92
3. SimTree edit distance	14860.30
4. Damerau-Levenshtein String Edit distance	44.66
5. Normalized Squared Euclidean distance	48.22
6. Cosine distance	41.93
7. Random Walker Termination	101.22

Once we got the pairwise distance matrices, we performed hierarchical agglomerative clustering with complete linkage. We applied a cut into the dendrograms at $k = 8$, and computed the evaluation scores – Adjusted Rand Index and NMI. We set k to 8 only because the data-set has ground truth labels and there were eight different types of ncRNA annotations, namely – (i) miRNA, (ii) tRNA, (iii) rRNA, (iv) scRNA, (v) snoRNA_CD, (vi) snoRNA_HACA, (vii) snoRNA_scaRNA, (viii) snRNA. There were few non-annotated segments which we discarded in all our evaluation (Table 3.2). Table 3.4 (rows 1–10) summarizes the results of all the experiments of our proposed PR2S2Clust system through the hierarchical clustering strategies applied on the Morin_EB dataset.

Furthermore, from the seven dendrograms retrieved from the hierarchical clusterings, we prepared seven linear partition vectors of the class membership by applying a cut in specified height to have eight clusters (i.e., $k = 8$). Table 3.4 (rows 8–10) summarizes the three different cluster aggregation strategies. We notice the consensus function MCLA performs the best and is the most appropriate to use in retrieving the final ensemble partition vector than the other two consensus functions.

Table 3.4: Performance of different hierarchical clustering approaches. The top scores are shown in bold-faces.

	Hierarchical Clustering Approach	**ARI**	**NMI**
PR2S2Clust System	1. Binary LZ distance	0.6216	0.0777
	2. Quaternary LZ distance	0.6385	0.1609
	3. SimTree edit distance	0.6458	0.1694
	4. String Edit distance	0.6959	0.3124
	5. Normalized Squared Euclidean distance	0.5827	0.2240
	6. Cosine distance	0.6589	0.2444
	7. Random Walker Termination distance	0.7036	0.2897
	8. Co-association based cluster aggregation	0.5861	0.0369
	9. Meta-clustering based cluster aggregation	**0.7183**	**0.3159**
	10. Majority Voting based cluster aggregation	0.5014	0.0527
	11. LocARNA [58] distance	0.3583	0.0642
	12. FoldAlign [43] distance	0.4069	0.1091
	13. GraphClust [44] distance	0.3520	0.0743
	14. deepBlockAlign [34] distance	0.3586	0.0699

We also applied the four existing algorithms (LocARNA [58], FoldAlign [59], GraphClust [44] and deepBlockAlign [34]) on the same dataset to retrieve the pairwise distance scores, and we performed similar hierarchical clustering strategy as applied in PR2S2Clust to obtain clustering results. We evaluated the performance of each of the methods in terms of ARI and NMI, and listed in Table 3.4 (rows 11–14). The clustering performance based on the pairwise distance scores obtained through the deepBlockAlign [34] is found to underperform. The alignment of block groups (i.e., the read-segments) only considers pairwise sequence level changes, wherein the pairwise segment distances computed by our proposed PR2S2Clust system considers both sequence and secondary structure level features of the segments that can be seen as the key player for the performance difference. Although, the LocARNA [58], FoldAlign [43] and GraphClust [44] methods considered both sequence and structural similarities during the computation of the pairwise alignment scores, none did not consider the patched representation of the RNA-seq read segments like in the PR2S2Clust system. This empirically justifies the performance boost by using all the pairwise distance variations used in our proposed system. Again, from the table we identify the strength of Meta-clustering based cluster aggregation (row 9 in the table) among all the available methods over the two cluster aggregation methods as well as any other individual clustering schemes. We also can see that the performance of the Random Walker Termination based strategy on the patched RNA secondary structures introduced in this study is comparable with other strategies in PR2S2Clust system.

3.7.2.2 Partitional Clustering of the Read-Segments

The set of experiments were conducted to emphasize on the investigation about the applicability of the ensemble partitional clustering over the ensemble hierarchical clusterings. The data matrices of dimension $21 \times N$ of the N read-segments in each of the datasets were generated by extracting twenty one features from the predicted secondary structures were the input to the several runs of k-means with combinations of varying initial centroid and distance metrics, where k were set to 8. The three consensus functions were once again applied to the generated partitions for aggregation. Table 3.5 summarizes the clustering performances of the three consensus functions in each of the three data-sets.

We can see the MCLA based consensus function consistently shows superior performance than the other two – co-association scheme and the majority voting algorithm in terms of all the three evaluation criteria (two external and one internal). If we compare the performance of MCLA from Table 3.5 with the result presented in Table 3.4, we realize that the ensemble partitional clustering strategy is the best to apply in practice than the ensemble hierarchical clustering.

Table 3.5: Ensemble partitional clustering of the dataset of RNA read block structural profiles.

Dataset	**Consensus Function**	**ARI**	**NMI**	**SSE**
Morin_EB	Co-association based aggregation	0.5067	0.0386	0.4637
	Meta-clustering based aggregation (MCLA)	**0.6918**	**0.3021**	**0.7341**
	Majority Voting based aggregation	0.377	0.0462	0.4637
GSM450598	Co-association based aggregation	0.5579	0.0354	0.5915
	Meta-clustering based aggregation (MCLA)	**0.6398**	**0.3096**	**0.7321**
	Majority Voting based aggregation	0.4032	0.0429	0.5942
GSM450605	Co-association based aggregation	0.4644	0.0246	0.4534
	Meta-clustering based aggregation (MCLA)	**0.7301**	**0.2541**	**0.6166**
	Majority Voting based aggregation	0.3440	0.0873	0.4475

3.7.2.3 Comparing Hierarchical and Partitional Clustering Results

In terms of experimental running time, partitional clustering algorithm is much faster than that of hierarchical clustering. The most exhausting part of the hierarchical clustering in our experiments was the preparation of pairwise distance matrices for N read-segments per dataset, that requires $O(N^2)$ running time as the pre-processing time before the we can call the hierarchical cluster function. In contrast, k-means algorithm, the partitional clustering we used in our study does not require this expensive pre-processing step. However, it requires vectorized representation of each of the segments. That is why we extracted twenty one structural features and formed a 21-dimensional vector for each of the read-segments. The running time of extracting all the 21 features from a given pool of secondary structures of the N read-segments is $O(N)$.

If we take a look at both Table 3.4 and Table 3.5, we will find that performance scores of hierarchical clustering is better than those of the partitional clustering which underlines the superiority of the hierarchical clustering over the partitional clustering with the cost of CPU time.

3.7.2.4 Comparing with Ensemble Clustering Results

Different classical clustering algorithms produce different clustering results because they impose different structures on the data. And, not a single clustering algorithm is optimal; moreover different realizations of the same algorithm may generate different results. However, the ensemble clustering exploits the complementary nature of the different clustering results of the same dataset to aggregate all into a consensus clustering result that agree to all the different input clusterings to some extent. Thus, the ensemble strategies are popular way of overcoming instabilities in each of the individual clustering algorithms. If we take a look at the seven classical hierarchical clusterings entries in Table 3.4, and compare with the three ensemble results just below the seven rows (rows 8–10), we can realize the phenomena. For instance, not a single classical clustering algorithm topped in all the evaluation scores for all the datasets. This underlines the fact that the performance of the individual clustering is somehow inconsistent than that of ensemble clusterings (specifically of the Meta-clustering based aggregation algorithm).

3.8 Conclusions and Future Works

In this article we explored various ensemble clustering strategies and application of these to various RNA-seq non-coding segment data-sets based on their structural dimensions. We presented justification for picking the appropriate clustering strategy to use in similar studies – classical individual clustering vs. ensemble clustering, or partitional clustering vs. hierarchical clustering. We showed that the use of the ensemble clustering will provide an extra level of confidence in the clustering results.

However, the ensemble partitional clustering would be best in practice than the ensemble hierarchical clustering strategies in terms of the standard evaluation criteria and computational cost and time. The results presented in this manuscript will help researchers to adopt the best strategy while they do intend to cluster their own segment data-sets and seek to focus on the structural perspective. One possible future research direction would be to aggregate the dendrograms without destroying the hierarchy in order to perform the ensemble hierarchical clustering. Investigating the application of the multi-dimensional structural profile vectors of the RNA-seq read segments in the multi-class classification framework would also reveal some structural properties that might be more discriminative than the others to solve the annotation problem.

CHAPTER 4

NMF based LncRNA-Disease Association

4.1 Introduction

With the advent of the High Throughput Sequencing (HTS) platform it is experimentally verified that the protein-coding genes account for only a small fraction of the human genome ($\sim 1.5\%$). In other words, more than 98% of the human genome do not code any protein; the fact implies that the traditional central dogma of molecular biology [60] is incomplete. There exists another branch along with the "traditional" dogma that explains a huge number of the non-protein coding genes that undergo transcription but never translate proteins [61, 62, 14]. Accumulating evidences reported over the past decade shed lights on many these non-coding RNAs (ncRNAs) and their functionalities in biological processes. The long non-coding RNAs (lncRNAs), a subclass of the ncRNAs having length more than 200 bases are discovered to be associated with many biological processes, such as imprinting control, epigenetic regulation, cell cycle control, nuclear and cytoplasmic trafficking, cell differentiation, immune responses and chromosome dynamics [63]. It is rather not surprising to discover the fact that the dysregulations and mutations of the lncRNAs are implicated in variety of human diseases [64, 65, 66]. That is why, a comprehensive understanding of potential human disease-related lncRNAs can facilitate development of our current knowledge-base; essentially that could explain accurately the molecular mechanisms of human diseases, their implications and also facilitate the diagnosis, treatment, prognosis and prevention [67, 68].

There are plenty of research efforts that have contributed into characterizing lncRNAs by generating the corresponding sequences, expression profiles and genomic annotations. But, only a few studies have been conducted to infer lncRNA-disease associations, indicating that we still are far from understanding the hidden functional associations of the lncRNAs. Of the few, Liao et al.[69] proposed the concept of coding-non-coding gene co-expression (CNC) network which was constructed from several gene expression dataset of coding and non-coding genes. The authors then conducted prediction of similar functional characteristics of lncRNAs from the CNC networks using a graph analytical approach. Guo et al. [70] developed a long non-coding RNA global function predictor (lnc-GFP) to predict probable functions for lncRNAs at large scale by integrating gene expression data and protein-protein interaction data. They also employed the concept of CNC network by Liao et al. [69]. But here the weighted CNC network was constructed using both the co-expression data and the protein-protein interaction data. Once the CNC network is built, a global propagation algorithm that is guaranteed to converge to a local minimum. The algorithm outputs the rank of all un-annotated genes with respect to a query function category. Finally, the top-ranked genes are functionally annotated with the function category of interest.

Yang et al. [71] presented a method to analyze lncRNA-disease associations, that can be used to predict lncRNA implicated diseases. Based on the available lncRNA-disease associations, two biological networks were constructed – an lncRNA-implicated disease network (lncDN) and disease-associated lncRNA network (DlncN). In lncDN, a vertex represents a disease, and a link between two vertices indicates the two corresponding diseases shared at least one lncRNA as their disease-causing lncRNA. However, in DlncN, a vertex represents an lncRNA, while a link between two nodes represents the fact that the two corresponding lncRNAs were implicated in at least one common disease. A graph analytical approach was applied to extract the similar lncRNAs and disease from these projected networks. Moreover, a propagation algorithm was applied on a weighted bipartite network of the lncRNA-disease associations to predict potential. Thus, by modeling the lncRNA-disease association as a bipartite network, and by mining the resultant network lncRNA and disease association scores were predicted. Chen et al. [68] developed a method of Laplacian Regularized Least Squares for LncRNA-Disease Association (LRLSLDA) that considered integrating the intergenic lncRNA (lincRNA)-tissue expression profiles obtained from the Human BodyMap LincRNA project [72], although only a few lncRNAs in their dataset were found to be intergenic that makes such an integration non-contributing towards the inference. However, the method prioritized the entire lncRNAome for disease of interest by integrating known phenome-lncRNAome network obtained from the existing database of lncRNA-disease associations.

All of these studies focused on solving and ranking lncRNA-disease associations. The prior knowledge of the coding gene - disease associations pops up a question: is there any possibility that there exists lncRNA-gene regulation in order to perform a particular function and or infer a disorder in the biological system? In fact this has been puzzled quite a few researchers and apparently there exists such lncRNA-gene relationship network inside cell [73, 74]. It is yet to explore the mysterious modular organization of the lncRNAs and coding genes, and understanding of their complex phenomena that are causing human diseases. In this article, we propose a computational framework for constructing the lncRNA-gene modules based on the integration of prior knowledge we have, so far. We incorporated the predicted lncRNA-disease association, and experimentally validated gene-disease association, gene-gene interaction data, expression profiles of both lncRNAs and genes in a non-negative matrix factorization (NMF) framework. We also applied the similar factorization approach on each of the association data alone to further cluster the lncRNAs in terms of meaningful disease groups.

The rest of the article is organized as follows: section 2 presents the details of the dataset used in this study, along with the preprocessing step. The lncRNA-disease association and the lncRNA-gene co-module discovery problem are formulated later in the section. Then, at section 3 we present experimental setup along with results. Finally, conclusions and future research directions are drawn at the section 4.

4.2 Materials and Methods

Figure 4.1 describes all our proposed framework to deal with four problems: (i) lncRNA-disease association inference, and corresponding bi-clustering, (ii) lincRNA-disease association inference, (iii) Categorizing lincRNAs by factoring the corresponding expression profile, (iv) lincRNA-gene co-module discovery. Each of the framework are discussed below.

4.2.1 Datasets and pre-processing

The Human BodyMap project provides us a catalog of 8,194 lincRNAs and 28,473 assembled from RNA-seq data from around 4 billion RNA-seq reads across 22 tissues [72]. The catalog contains transcript expression scores estimated through measuring corresponding abundances across the tissues using the tool Cufflinks [75].

We have collected the list of 1,028 experimentally validated associations among 322 long non-coding RNAs (lncRNAs) and 221 diseases from the LncRNADisease data server [67] on January 13, 2015.

The association dataset of long intergenic non-coding RNAs (lincRNAs) and diseases were extracted from the same data server on the same date. There listed was 1,564 lincRNAs and their associations with a pool of 1,641 diseases. The disease

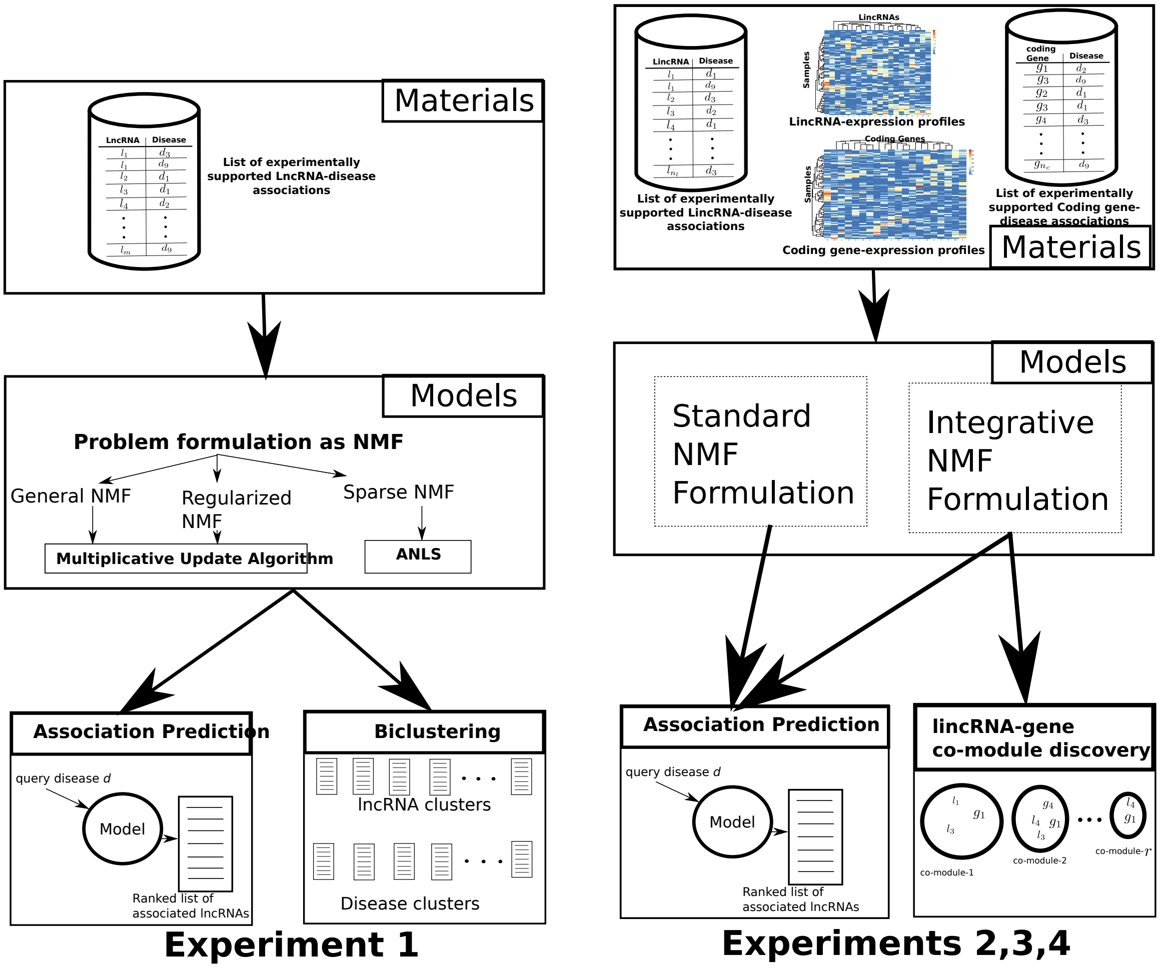

Materials
LncRNA
Disease
List of experimentally supported LncRNA-disease associations
Models
Problem formulation as NMF
General NMF
Regularized NMF
Sparse NMF
Multiplicative Update Algorithm
ANLS
Association Prediction
query disease d
Model
Ranked list of associated lncRNAs
Biclustering
lncRNA clusters
Disease clusters
Experiment 1
LincRNA
Disease
List of experimentally supported LincRNA-disease associations
LincRNAs
Samples
LincRNA-expression profiles
Coding Genes
Samples
Coding gene-expression profiles
coding Gene
Disease
List of experimentally supported Coding gene-disease associations
Materials
Models
Standard NMF Formulation
Integrative NMF Formulation
Association Prediction
query disease d
Model
Ranked list of associated lncRNAs
lincRNA-gene co-module discovery
co-module-1
co-module-2
co-module-r
Experiments 2,3,4

Figure 4.1: Overview of the proposed frameworks.

names that were used in the association set were not-canonical and had no index through using standard naming (e.g., RefSeq, ENSEMBL, and so on). We first tabulated all the disease names from the dataset, and then employed an OMIM API function call [76] to retrieve closely matched phenotype IDs (i.e., the entries prefixed with character # or %, or none), resulting a set of 684 OMIM phenotypes (mainly diseases) associated with the lincRNAs. We removed all the lincRNA entries from the association dataset that did not participate in the expression measurement, and removed all the disease entries that could not be matched with any valid OMIM phenotype ID and disjoint of the disease entries from the coding-gene association data-set. Finally, the lincRNA-disease association data-set contains associations among 562 lincRNAs and 645 OMIM diseases.

The associations among the coding-genes and diseases were pulled from DisGeNET web-server [77]. There were 415,681 associations among 16,666 coding genes and 13,135 diseases. Again, the disease names found were non-indexed through standard naming. We, therefore, applied OMIM function call again to map the disease names into closely matched phenotype IDs. We retain the coding-gene entries that were found expressed in the RNA-seq experiment. We retain disease entries in this case if valid OMIM phenotype IDs were found, and is also associated with at least one lincRNAs in the pool we selected above. Thus, we extracted the coding gene and disease association dataset among 13,425 coding-genes and 645 OMIM diseases.

Gene-gene genetic interaction network were extracted from [78]. Of the 4,836,794 genetic interactions between pairs of genes, we kept only the interactions among the genes from our gene pool, resulting 3,264,923 remaining interactions. We associate the genes with their Entrez identifier throughout the study.

Given the disease associations among lncRNAs and genes, we propose a way to build a lncRNA-gene co-association network from the disease association perspective. An lncRNA l is co-associated with a gene g if both implicates the same disease, d. Using this simple process we could extract 1,775,735 co-associative edges among the 562 lincRNAs and 13,425 coding genes.

4.2.2 Association Inference through Standard NMF Formulation

The lncRNA-disease association matrix $A \in \mathbb{R}_{+}^{m \times n}$, where m and n are the number of lncRNAs and diseases respectively, and $A_{i,j} = 1$ denoting there is at least one experimental evidence present that support association between lncRNA i

with disease j, otherwise the cell value would be 0. Each column of the matrix A corresponds to a data point in the m-dimensional space. The non-negative matrix factorization (NMF) [79, 80] technique divides such a matrix into two non-negative matrices: a basis matrix of lower rank $W \in \mathbb{R}_+^{m \times r}$ and a coefficient matrix $H \in \mathbb{R}_+^{r \times n}$, where the rank $r < \min\{m, n\}$, so that

$$A \approx WH. \tag{4.1}$$

An NMF solution is not unique, because of the fact that, for any diagonal matrix $D \in \mathbb{R}_+^{r \times r}$

$$A \approx WH = WDD^{-1}H = (WD)(D^{-1}H) = VG,$$

where, $V = WD$ and $G = D^{-1}H$. Both the matrices V and G are not necessarily equal to W and H respectively, implies the non-uniqueness property of the solution to the equation 4.1.

A solution to the NMF problem, however, can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_{W,H} \quad \mathcal{F}(A, W, H) \equiv ||A - WH||_F^2 \\ s.t., W \geq 0, H \geq 0, \end{aligned} \tag{4.2}$$

where $W \in \mathbb{R}_+^{m \times r}$ is a basis matrix, and $H \in \mathbb{R}_+^{r \times n}$ is a coefficient matrix. $W, H \geq 0$ means that all elements of W and H are non-negative. Since $r < m$ and $r < n$, dimensionality reduction is achieved, and a lower dimensional representation of A in a r-dimensional space is given by $H.|| \cdot ||_F^2$ is the square of the Frobenius norm and is defined as

$$||A - WH||_F^2 \quad = \quad \mathrm{tr}((A - WH)(A - WH)^T),$$

where tr is the matrix trace operator.

The fact that W and H are non-negative guarantees that parts of the matrix can be combined additively to form the given association matrix as a whole; NMF is a useful technique for obtaining a part-based representation of the data. In other words, factorization allows us to easily identify substructures in the data [81]. Several approaches to solve NMF by iteratively updating W have been presented in earlier studies [82]. Additional Bioinformatics applications of NMF are presented in a review article by Devarajan [83]. Several variants of NMF have been proposed by incorporating various kinds of constraints: discriminative constraints [84], locality-preserving or network-regularized constraints [85, 86] and sparsity constraints [87, 88].

One non-negative matrix factorization algorithm developed by Lee and Seung [89] is based on the multiplicative update rules of W and H, and is shown in Algorithm 2. The approximations of W and H remain non-negative during the updates.

Algorithm 2 Standard NMF based on Euclidean Distance [89]. Calculate W, H such that $A \approx WH$

Input: $A \in \mathbb{R}_+^{m \times n}$, rank r, and the two initial seed matrices $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$

Step 1: Normalize columns of A.

Step 2: Scale columns of W to sum to 1.

Step 3: Update H and W matrices using the following update rules:

$$H_{qj} \leftarrow H_{qj} \frac{(W^T A)_{qj}}{(W^T W H)_{qj} + \epsilon}, \quad (1 \leq q \leq r, 1 \leq j \leq n)$$

$$W_{iq} \leftarrow W_{iq} \frac{(A H^T)_{iq}}{(W H H^T)_{iq} + \epsilon}, \quad (1 \leq i \leq m, 1 \leq q \leq r)$$

Step 4: Scale columns of W to sum to 1.

Step 5: Repeat steps 3–5 until convergence

It is generally best to update W and H “simultaneously”, instead of updating each matrix fully before the other [90]. That is, after updating a row of H, we update the corresponding column of W. In the implementation, we added a small quantity $\epsilon = 2.2204 \times 10^{-16}$ to the denominators in the approximations of W and H in each iteration. However, Kullback-Leibler divergence based NMF formulation avoids similar multiplicative update rules, and is able to avoid numerical underflows to some extent (Algorithm 3).

Algorithm 3 Standard NMF based on Kullback-Leibler divergence [91]. Calculate W, H such that $A \approx WH$

```
**Input**: $A \in \mathbb{R}_+^{m \times n}$, rank $r$, and the two initial seed matrices $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$
Step 1: Normalize columns of $A$.
Step 2: Scale columns of $W$ to sum to 1.
Step 3: Update $H$ and $W$ matrices using the following update rules:
    $$H_{au} \leftarrow H_{au} \frac{\left( \sum_i \frac{W_{ia} A_{iu}}{(WH)_{iu}} \right)}{\sum_k W_{ka}}, \quad (1 \le a \le r, 1 \le u \le n)$$
    $$W_{ia} \leftarrow W_{ia} \frac{\left( \sum_u \frac{H_{au} A_{iu}}{(WH)_{iu}} \right)}{\sum_v H_{av}}, \quad (1 \le i \le m, 1 \le a \le r)$$
Step 4: Scale columns of $W$ to sum to 1.
Step 5: Repeat steps 3–5 until convergence
```

Pauca et al. [90] proposed a constrained NMF (CNMF) formulation,

$$\begin{aligned} \min_{W,H} \quad & ||A - WH||_F^2 + \alpha ||W||_F^2 + \beta ||H||_F^2 \\ & s.t., W \ge 0, H \ge 0, \end{aligned} \tag{4.3}$$

where α and β are regularization parameters. Algorithm 4 can be used to retrieve the two factors W and H. The regularization parameters $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are used to balance the trade-off between the accuracy of the approximation and the smoothness of the computed solution.

Algorithm 4 CNMF/Regularized NMF. Calculate W, H such that $A \approx WH$

```
**Input**: $A \in \mathbb{R}_+^{m\times n}$, rank $r$, and the two initial seed matrices $W \in \mathbb{R}_+^{m\times r}$ and $H \in \mathbb{R}_+^{r\times n}$
Step 1: Normalize columns of $A$.
Step 2: Scale columns of $W$ to sum to 1.
Step 3: Update $H$ and $W$ matrices using the following update rules:
        $H_{qj}^{(t)} \leftarrow H_{qj}^{(t-1)} \frac{((W^{(t-1)})^T A)_{qj} - \beta H_{qj}^{(t-1)}}{((W^{(t-1)})^T W^{(t-1)} H^{(t-1)})_{qj} + \epsilon}$
        for $1 \le q \le r, 1 \le j \le n$
        $W_{iq}^{(t)} \leftarrow W_{iq}^{(t-1)} \frac{(A(H^{(t)})^T)_{iq} - \alpha W_{iq}^{(t-1)}}{(W^{(t-1)} H^{(t)} (H^{(t)})^T)_{iq} + \epsilon}$
        for $1 \le i \le m, 1 \le q \le r$
Step 4: Scale columns of $W$ to sum to 1.
Step 5: Repeat steps 3–5 until convergence
```

Sparseness constraints can be enforced on W or H in the NMF formulation (Equation 4.2). Kim and Park [88] introduced two formulations and the corresponding algorithms for sparse NMFs – SNMF/L for sparse W, and the SNMF/R for sparse H. The following is the formulation of SNMF/L:

$$\begin{aligned} &\min_{W,H} \frac{1}{2}\{||A - WH||_F^2 + \eta||H||_F^2 + \beta \sum_{i=1}^{m} ||W(i,:)||_1^2\} \\ &s.t., W \geq 0, H \geq 0. \end{aligned} \tag{4.4}$$

Here, the parameter β is used to adjust the sparsity in W while the parameter η is used to preserve accuracy in H. And the formulation of the SNMF/R is:

$$\begin{aligned} &\min_{W,H} \frac{1}{2}\{||A - WH||_F^2 + \eta||W||_F^2 + \beta \sum_{j=1}^{n} ||H(:,j)||_1^2\} \\ &s.t., W \geq 0, H \geq 0. \end{aligned} \tag{4.5}$$

Again, the parameter β is used to adjust the sparsity in H and the parameter η is used to preserve the accuracy in W. Each of these two sparse NMF formulations that imposes the sparsity either on W or H utilizes L_1-norm minimization and the corresponding algorithms are based on Alternating Non-negativity constrained Least Squares (ANLS) [88]. The ANLS problem for SNMF/L is shown below:

$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\eta} I_r \end{pmatrix} H - \begin{pmatrix} A \\ \mathbf{0}_{r\times n} \end{pmatrix} \right\|_F^2, s.t. H \geq 0. \tag{4.6}$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\beta} e_{1\times r} \end{pmatrix} W^T - \begin{pmatrix} A^T \\ \mathbf{0}_{1\times m} \end{pmatrix} \right\|_F^2, s.t. W \geq 0. \tag{4.7}$$

Similarly, the ANLS problem for the SNMF/R formulation is given below:

$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\beta} e_{1\times r} \end{pmatrix} H - \begin{pmatrix} A \\ \mathbf{0}_{1\times n} \end{pmatrix} \right\|_F^2, s.t. H \geq 0. \tag{4.8}$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\eta} I_r \end{pmatrix} W^T - \begin{pmatrix} A^T \\ \mathbf{0}_{k\times m} \end{pmatrix} \right\|_F^2, s.t. W \geq 0. \tag{4.9}$$

4.2.2.1 LncRNA-Disease Association Inference

Non-negative Matrix Factorization models map both the lncRNAs and diseases to a joint latent factor space of dimensionality r, such that lncRNA-disease association are modeled as the inner products in the latent feature space $(f_1, f_2, \cdots, f_r)$. Accordingly, each lncRNA i is associated with a vector $\mathbf{l_i} \in \mathbb{R}^r$, and each disease j is associated with a vector $\mathbf{d_j} \in \mathbb{R}^r$. Thus, for a given lncRNA i, the elements of the vector $\mathbf{l_i}$ measure the extent to which the lncRNA possesses those factors, whereas for a given disease j, the elements of $\mathbf{d_j}$ measure the likelihood of association of the disease with corresponding factors. The dot product $\mathbf{l_i}^T\mathbf{d_j}$ captures the association between lncRNA i and disease j. This approximates the overall association of disease j with lncRNA i, that is denoted by $\hat{a}_{ij}$ leading to the estimate

$$\hat{a}_{ij} = \mathbf{l_i}^T \cdot \mathbf{d_j} \tag{4.10}$$

Once the NMF factorization is complete on matrix A, the inference system can easily estimate the likelihood of association of an lncRNA with a disease using equation 4.10. Figure 4.2 illustrates the inference process.

4.2.3 Bi-clustering

Many traditional clustering algorithms such as Hierarchical clustering have been applied for the purpose of clustering gene micro-array data which is an association between genes and samples to some extent [92, 93]. These strategies have a significant limitation: the approaches assign samples into some specific classes based on the genes' expression levels across all the samples. Sometimes, it is necessary to develop clustering methods that can identify the local structures, instead of the global phenomenon. Moreover, it has been shown in molecular biology that only a small number of genes or lncRNAs are involved in a pathway or biological process on most cases.

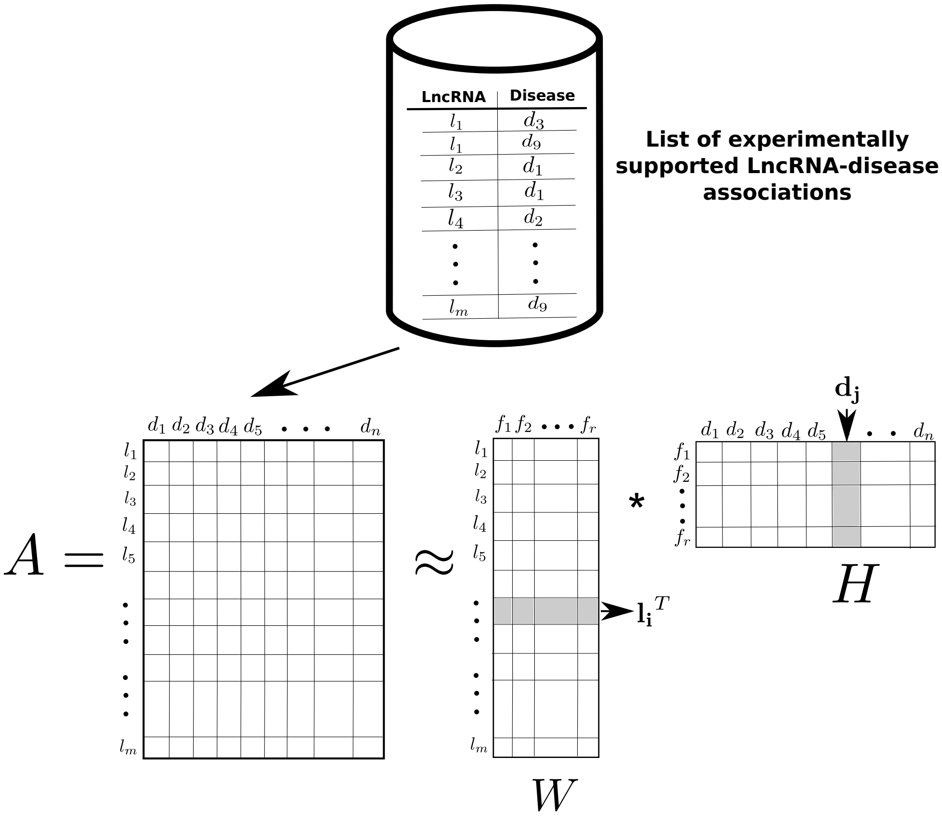


Figure 4.2: An abstract view of the lncRNA-disease association inference process. At first from the list of experimentally supported lncRNA-disease associations, the original association matrix A is formed, where $A_{ij} = x$, and $x \geq 0$ is a positive integer denoting number of experimental evidences that support the association between i^{th} lncRNA and j^{th} disease. Then NMF is applied to factor A into two matrices W and H. The corresponding rows of W and columns of H are then used to estimate the likelihood of the association betwen lncRNAs with diseases.

Specifically, only a small subset of lncRNAs are active for one cancer type, or one dysfunction, so generating sparse bi-clustering structures (i.e., the number of genes in each bi-clustering structure is small) is of great interest [94]. Many bi-clustering algorithms have been developed to explore the correlations between genes and samples and to identify the local gene-sample structures in the micro-array data, and some other association data [95]. However, the idea of bi-clustering is to characterize each lncRNA by a subset of diseases and to define each disease in a similar way. As a consequence, bi-clustering algorithms can select the groups of lncRNAs that show similar expression behaviors in a subset of diseases that belong to some specific classes such as some specific cancers, or disorders, and thus identify local structures of the association data.

Several bi-clustering algorithms have been proposed including BiMax, ISA, SAMBA, OPSM, which are evaluated in the review by Prelic et al [95]. However, bi-clustering can also be performed using NMF. The NMF factors can be used to perform bi-clustering analysis of the data matrix. The rows of the association matrix A represent lncRNAs, and the columns represent diseases. We can use the basis matrix W to divide the m lncRNAs into r lncRNA-clusters, and the coefficient matrix H can be used to divide the n diseases into r disease-clusters. Typically the following rules are used to assign membership:

- i^{th} lncRNA is assigned to the lncRNA-cluster q if the W_{iq} is the largest in $W(i,:)$, i.e., the i^{th} row of the matrix W.
- j^{th} disease is assigned to the disease-cluster q if the $H_{q,j}$ is the largest in $H(:,j)$, i.e., the j^{th} column of the matrix H.

4.2.4 Association inference through Data Integration in NMF Formulation

Here we defined the NMF objective function with three components: (i) The non-negative lincRNA and coding gene expression matrices $L \in \mathbb{R}_+^{n_t \times n_l}$ and $C \in \mathbb{R}_+^{n_t \times n_c}$, where n_l, n_c, n_t represent number of lincRNAs, number of coding genes and number of tissue samples considered in the expression datasets. (ii) the coding gene-coding gene genetic interaction network, $X \in \mathbf{R}_+^{n_c \times n_c}$, and (iii) lincRNA-coding gene co-association network, $Y \in \mathbf{R}_+^{n_l \times n_c}$ that represents the relationship among sets of lincRNAs and genes that are co-associated for similar diseases.

Here in this problem our goal is to identify lincRNA-gene co-modules. We further assume that there is a common basis matrix W for both the lincRNA and coding gene expression matrices L and C. These two expression matrices need to be factored into the basis W and two coefficient matrices H_l and H_c. Thus, the objective function becomes:

$$\min_{W,H_l,H_c} f(W, H_l, H_c) = ||L - WH_l||_F^2 + ||C - WH_c||_F^2, \tag{4.11}$$

where W, H_l, H_c have dimensions $n_t \times r$, $r \times n_l$ and $r \times n_c$ respectively. The parameter rank, r is chosen prior to the optimization. Solutions to the equation 4.11 is not unique [81] and demonstrate integrating prior knowledge to the above objective function to retrieve biologically significant results. In addition to that, such an integration can narrow down the large search space of lincRNA-gene co-modules.

If the gene-gene interaction network is defined as X and the lincRNA-gene co-association network as Y. We first define the objective function where we try to maximize the prior knowledge of the gene-gene interactions as possible. Thereby maximizing the following function:

$$\max_{W,H_l,H_c} \sum_{ij} x_{ij} (h_i^c)^T h_j^c = \mathrm{tr}(H_c X H_c^T). \tag{4.12}$$

This term ensures that the coding genes with known genetic interactions have similar coefficient profiles.

The co-association relationships between lincRNAs and coding genes can also be integrated similarly into the following function:

$$\max_{W,H_l,H_c} \sum_{ij} y_{ij} (h_i^l)^T h_j^c = \mathrm{tr}(H_l Y H_c^T). \tag{4.13}$$

We now combine the three objective functions from Equations 4.11, 4.12, 4.13 into the following single optimization function:

$$\min_{W,H_l,H_c} ||L - WH_l||_F^2 + ||C - WH_c||_F^2 - \lambda_1 \mathrm{tr}(H_c X H_c^T) - \lambda_2 \mathrm{tr}(H_l Y H_c^T), \tag{4.14}$$

where the parameters λ_1 and λ_2 are weights for the integration constraints defined in X and Y.

However, to avoid sparser NMF representations, we further imposed L_1-norm constraints on the H_l and H_c matrices [88], along with growth limiting constraints on the matrix W. Finally, our objective function for this problem becomes:

$$\begin{aligned} \min_{W,H_l,H_c} ||L - WH_l||_F^2 + ||C - WH_c||_F^2 - \lambda_1 \mathrm{tr}(H_c X H_c^T) - \lambda_2 \mathrm{tr}(H_l Y H_c^T) \\ +\gamma_1 ||W||_F^2 + \gamma_2 \left(\sum_j ||h_j||_1^2 + \sum_k ||h_k||_1^2 \right), \end{aligned} \tag{4.15}$$

where h_j and h_k are the j^{th} and k^{th} columns of the coefficient matrices H_l and H_c respectively. The term $\gamma_1 ||W||_F^2$ limits the growth of W, and $\gamma_2 \left(\sum_j ||h_j||_1^2 + \sum_k ||h_k||_1^2 \right)$ encourages sparsity.

The objective function in Equation 4.15 is not convex in W, H_l, H_c that kept us from obtaining the global minimum. Algorithm 5 can be adapted to retrieve reasonable local solutions to this optimization problem [81].

4.3 Experiment Results and Discussions

4.3.1 Experiment 1: NMF on the LncRNA-disease association

To evaluate the performances of the models, we preferred three widely used metrics, namely Mean Absolute Error (MAE), Accuracy and Root Mean Squared Error (RMSE) [96], which are defined as follow:

$$\mathrm{MAE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{|\tau|} \sum_{(i,j)\in\tau} |\hat{y}_{ij} - y_{ij}|, \tag{4.16}$$

$$\begin{aligned} \mathrm{Accuracy}(\hat{\mathbf{Y}}, \mathbf{Y}) &= \frac{1}{|\tau|} \sum_{(i,j)\in\tau} (1 - |\hat{y}_{ij} - y_{ij}|) \\ &= 1 - MAE(\hat{\mathbf{Y}}, \mathbf{Y}), \end{aligned} \tag{4.17}$$

Algorithm 5 Integrative NMF. Calculate W, H_l, H_c.

Step 1: Initialize W, H_l, H_c with non-negative random values drawn from uniform distribution.

Step 2: Fix H_l and H_c, and solve the following optimization problem:

$$\min_W ||L - WH_l||_F^2 + ||C - WH_c||_F^2 + \gamma_1 ||W||_F^2$$

This can be accomplished by updating W with the following update rule:

$$w_{ij} \leftarrow w_{ij} \frac{(LH_l^T + CH_c^T)_{ij}}{(WH_lH_l^T + WH_cH_c^T + \frac{1}{2}\gamma_1 W)_{ij}}$$

Step 3: Fix W, and solve the following optimization problem:

$$\begin{aligned} \min_{H_l,H_c} ||L - WH_l||_F^2 &+ ||C - WH_c||_F^2 \\ -\lambda_1 \mathrm{tr}(H_c X H_c^T) &- \lambda_2 \mathrm{tr}(H_l Y H_c^T) \\ +\gamma_2 &\left(\sum_j ||h_j||_1^2 + \sum_k ||h_k||_1^2 \right) \end{aligned}$$

This can be accomplished by updating H_l and H_c using the following update rules:

$$\begin{aligned} h_{ij}^l &\leftarrow h_{ij}^l \frac{(W^T L + \frac{1}{2}\lambda_2 H_c Y^T)_{ij}}{[(W^T W + \gamma_2 e_{r\times r})H_l]_{ij}} \\ h_{ij}^c &\leftarrow h_{ij}^c \frac{(W^T C + \lambda_1 H_c X + \frac{1}{2}\lambda_2 H_l Y)_{ij}}{[(W^T W + \gamma_2 e_{r\times r})H_c]_{ij}} \end{aligned}$$

Step 4: Repeat Step 2-3 until convergence.

$$\mathrm{RMSE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \sqrt{\frac{1}{|\tau|} \sum_{(i,j)\in\tau} (\hat{y}_{ij} - y_{ij})^2}, \tag{4.18}$$

where $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ are the computed association matrix and the observed association matrix respectively, while τ is the set of lncRNA-disease pairs for which we want to predict the ratings, that is, τ can be considered as the test set. The preference between the above two metrics depends on the particular application. In practice, MAE is popular for many collaborative filtering algorithms, while RMSE is still popular for the similar problems that generate real valued output.

Table 4.1: Evaluation of the three NMF Algorithms – Standard NMF, Regularized NMF and Sparse NMF in terms of mean absolute error (MAE) and root mean squared error (RMSE) by varying ranks (r).

r	MAE			RMSE		
	Standard NMF	Regularized NMF	Sparse NMF	Standard NMF	Regularized NMF	Sparse NMF
2	0.84	0.85	0.85	20.87	21.23	21.23
10	0.54	0.55	0.54	13.48	13.64	13.56
20	0.39	0.39	0.38	9.67	9.71	9.59
30	0.30	0.30	0.31	7.50	7.58	7.66
40	0.24	0.24	0.23	6.02	6.06	5.66
50	0.19	0.20	0.18	4.70	4.90	4.57

The lncRNA-disease associations are first split into five random folds. Then we performed five-fold cross validation to evaluate the model. Table 4.1 demonstrates the predictive performance of lncRNA-disease associations by using the three NMF models. The association matrix is first factored into W and H matrices using the three NMF algorithms. We performed several runs of NMF by varying rank of W and H, which are $r = 2, 10, 20, 30, 40, 50$. Then the original matrix is reconstructed by multiplying the computed W and H matrices. The estimated matrix is then compared with the original matrix for errors, in terms of RMSE and MAE scores.

Since, the accuracy and MAE scores are exactly complement to each other, it is evident from the Table 4.1 that once we increase the rank of the NMF factorization, the error decreases, as well as accuracy increases. As the rank of the NMF in our current dataset can only be less than 221 (that is, the minimum of the two dimensions of the association matrix), we showed here only the effect of choosing rank less than 50. The trend of accuracy and MAE can be equally observed in all of the NMF-based.

If we look at the trend of RMSE scores as the increment of rank in the various NMF implementations, we find all of the three algorithms show almost similar RMSE trend. However, since the input association matrix possesses sparsity property, it is better to use the sparse NMF considering the sparsity property into account. Thus, from Table 4.1 results we can conclude that the sparse NMF (SNMF/R) performed better than the other two NMF algorithms.

As explained in the previous section, a bi-clustering algorithm enables us to explore groups of entities that are similar within a small locality. Essentially, we are more interested to identify groups of lncRNAs that are associated with a very similar group of diseases, or disorders. Since, other than the lncRNA-disease association information we did not use any other characteristics of lncRNAs in our NMF-based formulations in order to understand similarities between lncRNA-pairs, it is not evidently interesting to perform clustering on the diseases that would reveal similar diseases groups. However, on the contrary, grouping lncRNAs reveals a number of useful characterization of lncRNAs in terms of the implication of diseases.

Table 4.2 lists out 10 significant clusters of lncRNAs that we retrieved after we performed a generalized NMF on the association matrix, and sought for two factors W and H of rank 322×10, and 10×221, meaning we expected a $r = 10$ rank approximation of the association matrix. Using the bi-clustering strategy described in the earlier section, we assigned membership scores for each of the 322 lncRNAs to any of the 10 disease classes. Here the latent feature space is 10-dimensional.

We then put the class major disease associations to the lncRNAs in Table 4.2 and found interesting lncRNA groups. For instance, there we see a prominent group (cluster # 5) of lncRNAs which are associated with heart diseases. All the lncRNAs in cluster #7 are associated with neurological disorders to some extent. Cluster # 6 contains all the lncRNAs which are mostly associated with hereditary disorders.

Cluster #1 is representing mostly the gastro-intestinal dysfunctions. All the remaining clusters are representative to several cancer categories and associated lncRNAs. A similar approach can also be employed to cluster the 221 diseases in the pool, according to the 10 latent features.

4.3.2 Experiment 2: NMF on the LincRNA-disease association

One of the essential parameter in NMF method is the rank r. It defines the number of meta-lincRNAs used to approximate the target association matrix. There are several approaches in picking the optimal value of r: (i) taking the first value of r for which the cophenetic correlation coefficient starts to decrease [91], (ii) picking the first value where the Residual Sum of Squares (RSS) curve present an inflection point [97]. The measures are plotted in Figure 4.3. Here, the cophenetic correlation coefficient decreased from rank 2 till 4, and then were raised for the increment of ranks. The explained variance (evar) went down after the rank 5. It is also important to notice that the basis W and the coefficient matrix H were sparse, and our framework was able to handle sparsity that makes the framework robust.

We further investigated the convergence speed of the NMF algorithms: brunet [91], lee [89], non-smooth NMF (nsNMF) [98], snmf/r and snmf/l [88]. The results are shown in Figure 4.4. Each curve reports the trajectory of the approximation residuals, computed with the corresponding method's loss function. Each track is normalized separately over its maximum value and stops at the number of iterations required to achieve the convergence criterion. We see that lee, snmf/l and snmf/r converged

Table 4.2: 7 prominent clusters of lncRNAs that were retrieved from our NMF models

ID	Mostly associated diseases	lincRNAs
1	gastric cancer, liver related cancer, kidney injury	AIR, CCAT1, DQ786243, Dreh, ENST00000513542, GNAS-AS1, HEIH, HOTTIP, HULC, IGF2-AS, KCNQ1OT1, LALR, LDMAR, LINCMD1, lncRNA-ATB, lncRNA-MVIH, MINA, MIR7-3HG, NPTN-IT1, np_17856, np_5318, RNA polymerase III-dependent lncRNAs, RNase MRP, VL30 LTRs
2	Esophageal squamous cell cancer, type II diabetes, melanoma	1B FGF-antisense transcripts, Alu lncRNAs, CDKN2B-AS1, CDKN2B-AS10, CDKN2B-AS11, CDKN2B-AS13, CDKN2B-AS2, CDKN2B-AS3, CDKN2B-AS5, CDKN2B-AS7, CDKN2B-AS8, D4Z4, ESCCAL-1, ESCCAL-5, ESRG, Gm20748, HI-LNC25, HYMAI, KUCG1, LINC00032, LINC01262, NPPA-AS1, NRON, PDZRN3-AS1, PISRT1, PTHLH, SPRY4-IT1
3	Angelman syndrome, Prader-Willi syndrome, Silver-Russell syndrome	116HG, AK023948, anti-NOS2A, BDNF-AS1, C15orf2, H19, IPW, KCNQ1DN, MAP3K14, MESTIT1, MIR100HG, MKRN3-AS1, SCAANT1, SLC7A2-IT1A/B, SNHG11, Ube3a-as, UBE3A-AS1, UBE3A-ATS
4	prostate cancer, enterovirus infection, autoimmune disease	AC002511.1, AP000688.29, ATXN8OS, C1QTNF9B-AS1, CBR3-AS1, CCND1 promoter-derived lncRNAs, CDKN2B-AS9, CTBP1-AS, DAPK1, DLEU1, DLEU2, DNM3OS, GAS5, Kcna2 antisense RNA, LINC00162, Linc00963, LOC728606, LSINCT5, MIR155HG, NAMA, PCA3, PCGEM1, PCNCR1, PRNCR1, PVT1, RP4-620F22.3, RP5-843L14.1, SCHLAP1, SNHG5, SRA1, TCL6, TERC, ZFAT-AS1
5	Heart Failure	5730458M16Rik, AK038798, AK044955, AK049728, AK137898, AK144081, AK153778, BX118339, DMPK, DMPK 3'UTR, ENSMUST00000022467, ENSMUST00000041159, ENSMUST00000117372, ENSMUST00000117393, ENSMUST00000119855, ENSMUST00000120925, ENSMUST00000127230, ENSMUST00000127429, ENSMUST00000130025, ENSMUST00000142855, ENSMUST00000143888, ENSMUST00000160947, ENSMUST00000167632, FADS1, Fendrr, Gm12839, Gm6644, LIPCAR, LOC102635190, Scarb2, Trpm3, uc.115-, uc.184+, UCH1LAS, Zim3
6	Hereditary Haemorrhagic Telangiectasia, fragile X syndrome	B1 SINE RNA, ENSG00000135253.9, ENSG00000147753.5, ENSG00000196096.3, ENSG00000197251.3, ENSG00000203325.3, ENSG00000206129.3, ENSG00000215231.3, ENSG00000215374.4, ENSG00000215808.2, ENSG00000226496.1, ENSG00000229563.1, ENSG00000230133.1, ENSG00000230544.1, ENSG00000231133.1, ENSG00000231185.2, ENSG00000232021.2, ENSG00000232046.1, ENSG00000232956.3, ENSG00000233154.1, ENSG00000233251.3, ENSG00000235285.1, ENSG00000237036.3, ENSG00000237548.1, ENSG00000240453.1, ENSG00000241269.1, ENSG00000245910.3, ENSG00000248176.1, ENSG00000249364.1, ENSG00000249772.1, ENSG00000250195.1, ENSG00000250608.1, ENSG00000254154.3, ENSG00000255471.1, ENSG00000256218.1, ENSG00000259150.1, ENSG00000259334.1, ENSG00000259484.1, ENSG00000259758.1, ENSG00000263753.1, ENSG00000264772.1, ENSG00000266952.1, FMR4, FMR6, RNA-a
7	Alzheimer's disease, bipolar disorder, Huntington's disease, schizophrenia, depression, DiGeorge syndrome	51A, 7SL, BACE1-AS, BCYRN1, BDNF-AS, DAOA-AS1, DGCR5, DISC2, DLG2AS, FGF10-AS1, GDNFOS, HAR1A, HAR1B, HCP5, HELLPAR, HLA-AS1, HTTAS, HTTAS_v1, IFNG-AS1, LINC00271, LINC00299, LOC389023, NEAT-1, PRINS, PSORS1C3, PTCSC, PTCSC3, REST/CoREST-regulated lncRNAs, SNHG3, SOX2-OT, TRAF3IP2-AS1, TUG1

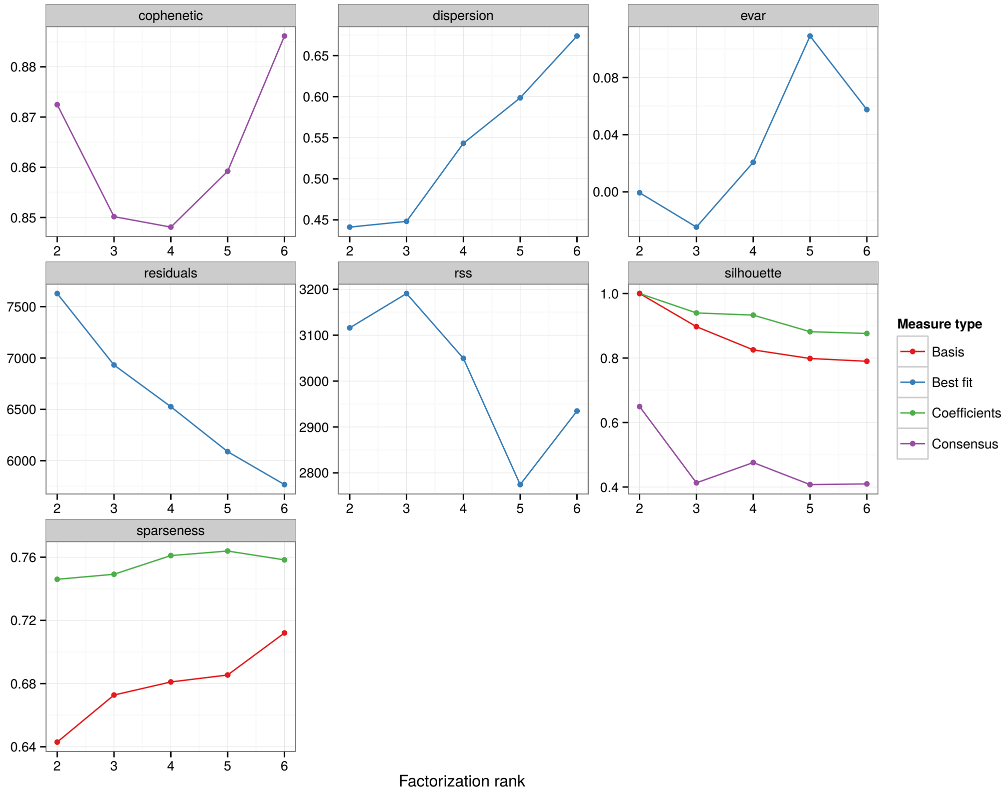


Figure 4.3: Estimation of the rank parameter. Each of the quality measures were computed from 10 runs for each value of the rank. Here, we estimate the rank to be 4 (according to [91]).

early than the brunet and nsNMF algorithms. Although the brunet and nsNMF kept moving towards more minimum solutions, than the rest of the algorithms.

Table 4.3 illustrates the comparison between the five NMF codes we applied on our formulation. We performed leave-one-out-cross validation (LOOCV) to measure the accuracy of each of the formulation. We found that SNMF/R performed superior than the other four algorithms. Moreover, SNMF/R algorithm is fast since it converges quickly and it also is robust since it takes into account the sparsity constraints. All the other parameters, like the Silhouette for the basis and coefficient matrices of the five algorithms are almost similar.

Table 4.3: Comparison results of running several NMF algorithms on the LincRNA-disease association dataset.

Method	**Sparseness (basis)**	**Sparseness (coefficient)**	**Silhouette (coeffient)**	**Silhouette (basis)**	**Residuals**	**iterations**	**Accuracy**
brunet	0.71	0.73	0.90	0.85	6573.31	2000	0.63
lee	0.55	0.64	0.85	0.80	1093.55	780	0.41
nsNMF	0.74	0.80	0.88	0.88	7257.59	2000	0.74
snmf/r	0.55	0.71	0.87	0.83	1100.07	195	0.81
snmf/l	0.56	0.69	0.86	0.83	1101.12	185	0.73

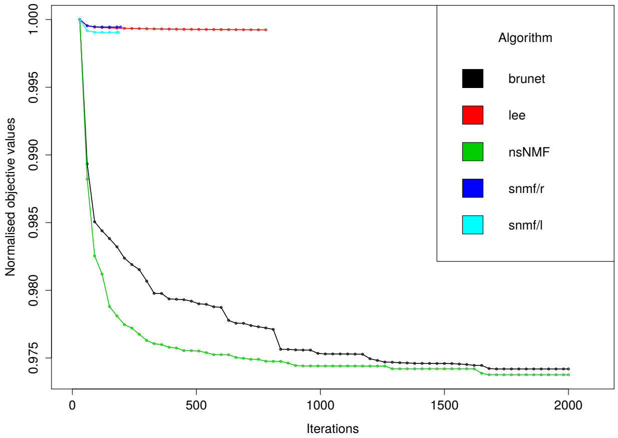


Figure 4.4: Error track for the runs of the three NMF algorithms: brunet, lee, nsNMF, snmf/r, snmf/l

4.3.3 Experiment 3: NMF on the LincRNA expression dataset

In this set of experiment, we applied the same five NMF algorithms on the LincRNA expression profiles encoded the the $L \in \mathbb{R}_{+}^{n_t \times n_l}$ matrix, where n_t is the number of tissue samples where the n_l number of LincRNA abundance scores were calculated. Since the expression scores of the transcripts from an RNA-seq experiment is merely read-counts of the fragmented reads, the matrix L is always non-negative and we did not require any preprocessing before applying NMF on the matrix.

The main goal of this particular experiment is to achieve a low-rank representation of the expression profile. There were $n_t = 22$ samples of the 562 lincRNAs. Figure 4.5 illustrates the metrics we seek to estimate rank for the NMF algorithm. As we see a decreasing trend of the cophenetic correlation coefficient at rank 4. The dispersion and the explained variance (evar) are at increasing trend at or after rank 4. We, therefore, picked 4 to be the rank. Using the 4-rank representation of the expression profile, we identified clusters of lincRNAs. Essentially, since the lincRNAs were missing class labels, it was not possible for us to evaluate. However, we put the

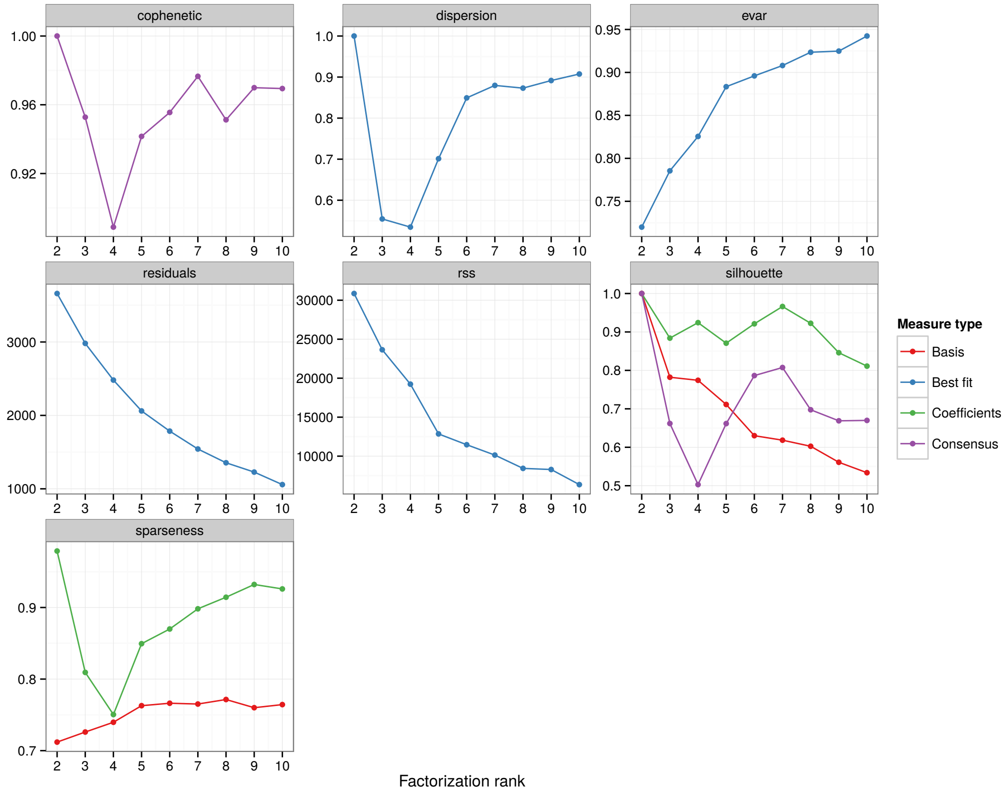


Figure 4.5: Estimation of the rank parameter. Each of the quality measures were computed from 10 runs for each value of the rank. Here, we estimate the rank to be 4 (according to [91]).

clustering results in a publicly available website, so that interested researchers can infer significant relationships among lincRNAs (Please see section: Availability).

Once again, we drew performance comparison of the five NMF algorithms on the dataset. Since we lack the class labels for this particular cases, we omit the external evaluation scores, rather showing the intrinsic evaluation measures in Table 4.4.

Table 4.4: Comparison results of running several NMF algorithms on the LincRNA-expression dataset.

Method	**Sparseness**	**Sparseness**	**Silhouette**	**Silhouette**	**Residuals**	**Iteration**	**Cophenetic**	**Dispersion**
	(basis)	**(coef.)**	**(coef.)**	**(basis)**			**corr.**	
							coef.	
brunet	0.740	0.750	0.924	0.774	2481.242	440	0.889	0.535
lee	0.763	0.734	0.769	0.735	6477.119	740	1	1
nsNMF	0.772	0.841	0.914	0.819	3286.715	480	0.917	0.561
snmf/r	0.761	0.795	0.781	0.720	9279.551	75	1	1
snmf/l	0.800	0.606	0.724	0.757	9575.074	115	1	1

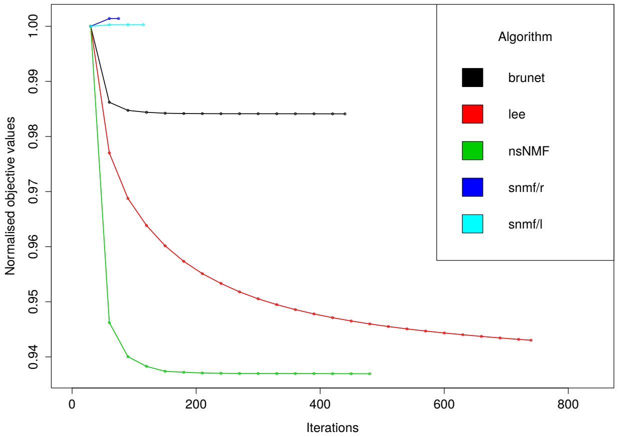


Figure 4.6: Error track for the runs of NMF algorithms: brunet, lee, nsNMF, snmf/r, snmf/l

4.3.4 Experiment 4: Integrative NMF approach for LincRNA-gene co-module identification

The integrative NMF algorithm (Algorithm 5) was applied to identify lincRNA-gene co-modules. The algorithm requires setting five parameters: rank r, interaction constraint parameter λ_1, relationship constraint parameter λ_2, growth parameter for the basis matrix γ_1 and sparsity parameter γ_2. Since we have extracted 645 diseases with OMIM phenotype IDs, we can further categorize these diseases into 20 classes according to the work of [99]. For this regard, we manually assigned the 645 diseases into the 20 classes. It should be noted that the mapping turned out to be a surjective function. Figure 4.7 illustrates the distribution of the 20 disease classes that represents the 645 mapped diseases from the dataset.

Since we ended up with the 20 disease classes, we set the number of lincRNA-gene co-module to be 20 $(= \text{rank}, r)$. However, the remaining four parameters were determined empirically using a brute-force search over $\gamma_1, \gamma_2 \in \{10, 5\}, \lambda_1, \lambda_2 \in \{0.1, 0.01, 0.001\}$ on the matrices. Each integrative NMF function was allowed to

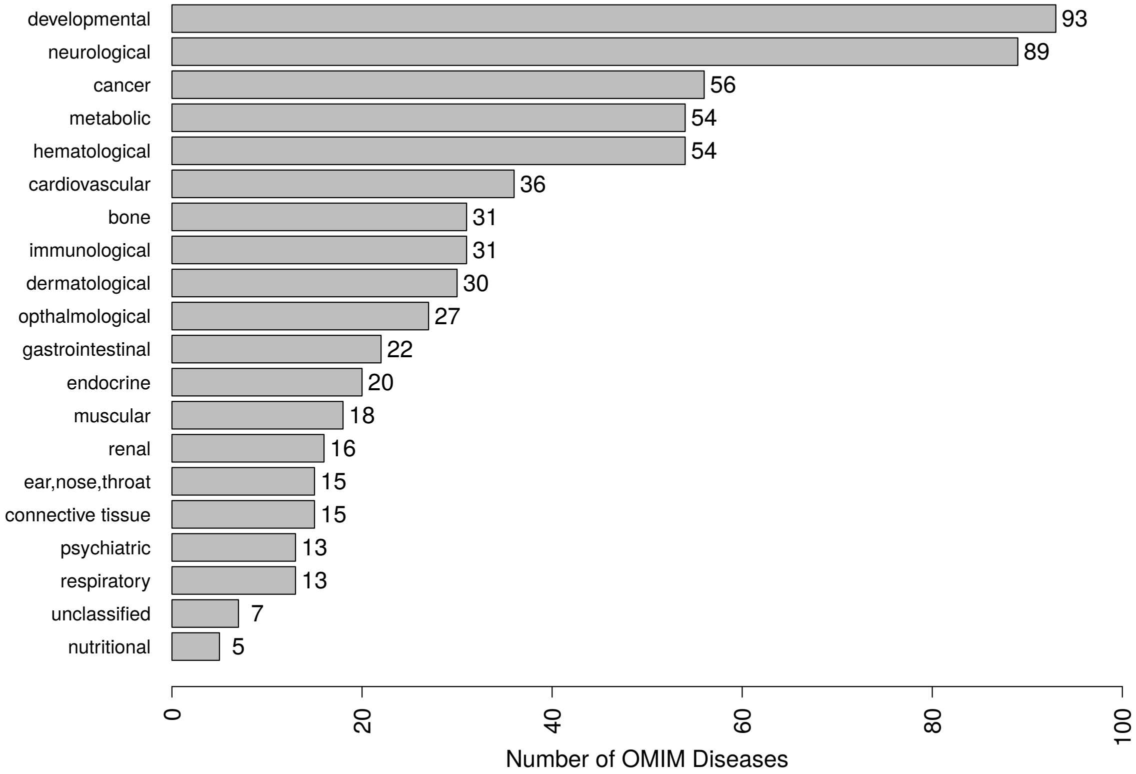


Figure 4.7: . The mapping-frequency distribution of the 20 disease classes. It was interesting to find out that the "developmental" and "neurological" disease classes were mapped to about one-third of all the diseases.

iterate at most 20 times, and each of the NMF call is repeated three times. Thus, altogether 108 NMF function calls were made with the same dataset with rank $r = 20$. We carefully measured relative error in each iteration inside an NMF function call for a specific parameter settings. Here, the relative error is computed as:

$$\text{relative error} = \frac{\frac{1}{n_t n_l}\sum_{ij}|l_{ij} - (WH_l)_{ij}|}{\frac{1}{n_t n_l}\sum_{ij} l_{ij}} + \frac{\frac{1}{n_t n_c}\sum_{ij}|c_{ij} - (WH_c)_{ij}|}{\frac{1}{n_t n_c}\sum_{ij} c_{ij}}, \tag{4.19}$$

based on the matrices we defined earlier. We noticed that the NMF algorithm was gradually decreasing the relative error with the parameter setting: $\lambda_1 = 0.001, \lambda_2 =$

$0.001, \gamma_1 = 10, \gamma_2 = 10$. Thus, we fixed our parameters to have these settings before running the integrative NMF algorithm.

The output coefficient matrices H_l and H_c of the algorithm were used to identify lincRNA-gene co-modules. We used the maximum coefficient in each column of H_l and H_c to discover patterns and determine memberships among the lincRNAs and genes [91]. Then, for each predicted co-module, we computed the percentage of lincRNAs co-associating with the coding genes for causing one of the 20 category of diseases. Table 4.5 presents the percentage scores of our integrative NMF algorithm for each of the 20 co-modules. We noticed that the co-modules 1, 5, 7 and 9 do not contain any lincRNAs, therefore, the percentage coverage score is left blank in the table. We compared our co-module association results with that of k-means and agglomerative hierarchical clustering with average linkage, with $k = 20$ and cut at 20 respectively. We also included the percentage scores of the lincRNAs co-associating with genes in the table. But interestingly, we see that both k-means and hierarchical clustering failed to identify such lincRNA-gene co-modules, leaving most of the entries in the table blank. This proves the robustness of our integrative NMF algorithm over traditional clustering approaches to deal with such problem. The list of co-modules and the members were kept in a publicly accessible url (Please see the section: Availability).

4.4 Conclusion and Future Research Scopes

Many lncRNAs play critical roles in human diseases and disorder pathways. An lncRNA may implicate multiple diseases, while a disease could be a result by association of several canonical lncRNAs. A comprehensive understanding of the associations is necessary in diagnosis, and novel drug discovery, and future research in this domain. However, a very little is known about the association of lncRNAs

Table 4.5: Performance comparisons of lincRNA-gene co-module discoveries of our proposed integrative NMF approach along with two base-line clustering algorithms, k-means and hierarchical (agglomerative) clustering

Module no.	**Percentage of lncRNAs co-associating with genes**		
	Integrative-NMF	**K-means**	**Hierarchical Clustering**
1			0.99
2	0.83		
3	0.67		
4	0.96	1.00	
5			
6	1.00		
7			
8	0.90		
9		0.98	
10	1.00		
11	1.00		
12	0.94		
13	1.00		
14	0.89		
15	0.75		
16	0.80		
17	0.95		
18	0.78		
19	0.75		
20	0.78		

with diseases, co-associations of such molecules with other genes as compared to the exponential rate of discovery of the lncRNAs per year.

In this article, we proposed the three NMF-based formulations for solving three different problems: (i) lncRNA-disease association problem, (ii) clustering of lncRNAs based on expressions, (iii) lncRNA-gene co-module discovery through integration of existing knowledge about them. We implemented the NMF algorithms to solve the problems. The models have two-fold properties – they are able to explain each of the associated lncRNA as well as the disease in a latent feature space that can be considered a dimensionality reduction step before further processing. Secondly, the NMF factors can be used to retrieve bi-clusters, that is, groups of similar lncRNAs, and groups of similar diseases in the latent feature dimension. Initially, we thought that any NMF-based formulation that only considers the existing knowledge of lncRNA-disease association would be fair enough to be used in practical association prediction problems. But, through the data integration based NMF experiments we realized the computational framework provided association results can also confidently provide meaningful biological insight of the associations through co-modules which is currently grasped attention to many non-coding research groups.

There are some limitations exist in our NMF-based lncRNA-disease association approach. Firstly, although we adapted a grid search technique to choose parameter settings in our experiments, we still need a better and faster way to accomplish this. Secondly, we only incorporated the existing lncRNA-disease association information, relationship networks among lncRNAs and genes, but did not include disease-related information, disease-disease similarity network, textual profiles of diseases which presumably would produce even meaningful results. Thirdly, in this current setup, lncRNA association inference for a query disease can only be possible for a fixed set of diseases and lncRNAs, because of the inherent transductive property of the problem formulation. However, it might be a prominent research possibility to overcome this limitation that would enable the users to apply the same model to identify novel potential lncRNA-associated diseases as well.

4.5 Availability

All the pre-processed data, source codes (written in R language) and experiment results were available at a publicly available website located at `http://biomecis.uta.edu/~ashis/res/netmahib2015/`.

4.6 Comments

Fixing an issue with the integrative NMF optimization problem: Consider the objective function $J(W, H_l, H_c)$ for the integrative NMF as designed in Equation 4.14 from Chapter 4 (re-written here for convenience to the readers):

$$\min_{W,H_l,H_c} J = ||L - WH_l||_F^2 + ||C - WH_c||_F^2 - \lambda_1 \text{tr}(H_c X H_c^T) - \lambda_2 \text{tr}(H_l Y H_c^T)$$

If any optimization algorithm stops at a solution tuple (W^*, H_l^*, H_c^*) as the optimum point, one can immediately claim a second tuple optimum than that point, which is, $(W^{**}, H_l^{**}, H_c^{**})$, where $W^{**} = \frac{1}{10}W^*$, $H_l^{**} = 10H_l^*$ and $H_c^{**} = 10H_c^*$. To prove this claim, we can compare the two quantities: $J(W^*, H_l^*, H_c^*)$ and $J(W^{**}, H_l^{**}, H_c^{**})$. Here, $||L - W^{**}H_l^{**}||_F^2 = ||L - W^*H_l^*||_F^2$, and $||C - W^{**}H_c^{**}||_F^2 = ||C - W^*H_c^*||_F^2$. But, $\lambda_1 \text{tr}(H_c^{**} X H_c^{T**}) = 100\lambda_1 \text{tr}(H_c^* X H_c^{T*})$, and $\lambda_2 \text{tr}(H_l^{**} Y H_c^{T**}) = 100\lambda_2 \text{tr}(H_l^* Y H_c^{T*})$. This implies $J(W^{**}, H_l^{**}, H_c^{**}) < J(W^*, H_l^*, H_c^*)$. Now, given the tuple $(W^{**}, H_l^{**}, H_c^{**})$ that is just claimed as the optimum solution, then another optimum solution can be found using similar fashion: $(W^{***}, H_l^{***}, H_c^{***})$, and so on. Exact optimum solution, thus, can never be obtained for this phenomenon.

As a remedy we propose an alternate objective as shown in Equation 4.20.

$$\begin{aligned} \min_{W,H_l,H_c} J(W, H_l, H_c) = ||L - WH_l||_F^2 + ||C - WH_c||_F^2 \\ -\lambda_1 \text{tr}(WH_c X H_c^T W^T) - \lambda_2 \text{tr}(WH_l Y H_c^T W^T) \\ +\gamma_1 ||W||_F^2 + \gamma_2 \left(\sum_j ||h_j||_1^2 + \sum_k ||h_k||_1^2 \right), \end{aligned} \tag{4.20}$$

The objective function can be written as:

$$\begin{aligned} J \;=\; & tr(L^T L - 2H_l^T W^T L + H_l^T W^T W H_l) + tr(C^T C - 2H_c^T W^T C + H_c^T W^T W H_c) \\ & -\lambda_1 tr(WH_c X H_c^T W^T) - \lambda_2 tr(WH_l Y H_c^T W^T) + \gamma_1 tr(W^T W) \\ & +\gamma_2 e_{1\times r} H_l H_l^T e_{1\times r}^T + \gamma_2 e_{1\times r} H_c H_c^T e_{1\times r}^T \end{aligned} \tag{4.21}$$

Let ψ_{ij}, ξ_{ij} and ϕ_{ij} be the multipliers for the constraints $W_{ij} \geq 0, (H_l)_{ij} \geq 0$ and $(H_c)_{ij} \geq 0$ respectively. Thus, the Lagrangian is:

$$\mathcal{L}(W, H_l, H_c) = J + tr(\Psi W^T) + tr(\Xi H_l^T) + tr(\Phi H_c^T) \tag{4.22}$$

The partial derivatives of $\mathcal{L}$ with respect to W, H_l, H_c are:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial W} &= -2LH_l^T + 2WH_lH_l^T - 2CH_c^T + 2WH_cH_c^T \\
&\quad -2\lambda_1 WH_cXH_c^T - 2\lambda_2 WH_lYH_c^T + 2\gamma_1 W + \Psi \\
\frac{\partial \mathcal{L}}{\partial H_l} &= -2W^TL + 2W^TWH_l - \lambda_2 W^TWH_cY^T + 2\gamma_2 e_{r\times r}H_l + \Xi \\
\frac{\partial \mathcal{L}}{\partial H_c} &= -2W^TC + 2W^TWH_c - 2\lambda_1 W^TWH_cX + \lambda_2 W^TWH_lY + 2\gamma_2 e_{r\times r}H_c + \Phi
\end{aligned}$$

Based on the KKT conditions $\psi_{ij}W_{ij} = 0, \xi_{ij}\left(H_l\right)_{ij} = 0, \phi_{ij}\left(H_c\right)_{ij} = 0$. So, we get the following equations for W, H_l, H_c:

$$\begin{aligned}
&\left[-2LH_l^T - 2CH_c^T - 2\lambda_1 WH_cXH_c^T - 2\lambda_2 WH_lYH_c^T\right]_{ij} W_{ij} \\
&\quad + \left[2WH_lH_l^T + 2WH_cH_c^T + 2\gamma_1 W\right]_{ij} W_{ij} = 0, \\
&\left[-2W^TL - \lambda_2 W^TWH_cY^T\right]_{ij} (H_l)_{ij} + \left[2W^TWH_l + 2\gamma_2 e_{r\times r}H_l\right]_{ij} (H_l)_{ij} = 0 \\
&\left[-2W^TC - 2\lambda_1 W^TWH_cX\right]_{ij} (H_c)_{ij} + \left[2W^TWH_c + \lambda_2 W^TWH_lY + 2\gamma_2 e_{r\times r}H_c\right]_{ij} (H_c)_{ij} = 0
\end{aligned}$$

Thus, we get the following update rules to obtain W, H_l, H_c:

$$W_{ij} \leftarrow W_{ij} \frac{\left(LH_l^T + CH_c^T + \lambda_1 WH_cXH_c^T + \lambda_2 WH_lYH_c^T\right)_{ij}}{\left(WH_lH_l^T + WH_cH_c^T + \gamma_1 W\right)_{ij}} \tag{4.23}$$

$$(H_l)_{ij} \leftarrow (H_l)_{ij} \frac{\left(W^TL + \frac{\lambda_2}{2} W^TWH_cY^T\right)_{ij}}{(W^TWH_l + \gamma_2 e_{r\times r}H_l)_{ij}} \tag{4.24}$$

$$(H_c)_{ij} \leftarrow (H_c)_{ij} \frac{\left(W^TC + \lambda_1 W^TWH_cX\right)_{ij}}{\left(W^TWH_c + \frac{\lambda_2}{2} W^TWH_lY + \gamma_2 e_{r\times r}H_c\right)_{ij}} \tag{4.25}$$

CHAPTER 5

LiDiAimc: LincRNA-Disease Associations through Inductive Matrix Completion

5.1 Introduction

LincRNA-Disease association inference problem: The protein-coding genes are the most well studied regions in the entire human genome. However, such genes account for only 2% of the genome [100]. In recent years, it has become evident that the non-protein coding portion of the genome, especially the long intergenic non-coding RNAs (lincRNAs) having length more than 200 bases each with no overlaps with any annotated protein-coding regions, are of critical functional importance for their diverse molecular mechanisms and implications of various human diseases [101]. With the advent of the high-throughput genomic technologies, such as RNA-seq and ChIP-seq, a huge number of lincRNAs have been cataloged. However, determining their functions, specifically the associations of the lincRNAs to human diseases, remain a challenge [72]. *In silico* association inference tools would present, in this regard, an important facet towards discovering causal lincRNA-disease relationships and better understanding of the human diseases. Such tools would be able to rank disease implications by a given lincRNA based on prior knowledge.

Limitations of existing methods: There are several long non-coding RNA (lncRNA)-disease association inference tools developed in the previous three years. Unfortunately, only a few dealt with the lincRNA-disease inference problem. Due to the intricacies inherent to this association inference problem, only a small number of experimentally validated associations have been reported in the publicly available database, such as lncRNAdisease [67]. For this reason, leveraging multiple complementary data sources is essential for predicting lincRNAs related to disease phenotypes, and thus different inference methods have been developed considering different knowledge sources. For instance, K-RWRH [102], LRLSLDA [68] and TslncRNA-disease [103] are popular family of network based methods. The methods utilize biological networks, such as lincRNA similarity network and disease similarity network and infer lincRNA-disease connections by either using random walk procedure on a derived biological network or by computing a similarity measure between nodes with known disease implications. The association inference problem can also be solved using a matrix completion approach. These approaches suffer from the cold start problem, due to the inability to address the inference predictions of the diseases for novel lincRNAs and vice versa. Furthermore, these methods were presented on a very small set of associations and developed without considering the scalability (e.g., around 200 lncRNAs compared to more than 8000 lincRNAs available to date from [72] research remain overlooked). However, the methods exploiting lincRNA-expression profiles to build similarity networks may only deal with specific disease classes that are only available through the seed or true associations and therefore the methods fall short in generalizing to novel diseases.

Outline of our proposed approach: In this article, we propose a method, LiDiAimc, that can easily integrate complementary features of both the lincRNAs and the diseases. It provides better coverage and generalization than any other methods focusing only on a specific data source. Our method involves two steps. Firstly, we extract features of the lincRNAs and the diseases from multiple data sources. Then, we integrate the features in the Inductive Matrix Completion (IMC) approach [104] to learn the lincRNA-disease association network. Such a scheme enables LiDiAimc method overcoming the cold-start problem, as well as providing an interface to a more generalized integration of multiple data sources of both the lincRNAs and the diseases. We evaluate our proposed LiDiAimc method through extensive experiments and the results show superior performance compared to the state-of-the-art methods.

Summary of contributions:

- We demonstrate that the integration of diverse features of the lincRNAs and the diseases available through publicly available data-servers can overcome worse predictive performance issue faced by the inference tools which occurs due to the extreme sparsity inherent to the lincRNA-disease association dataset.
- We provide an application of the Inductive Matrix Completion (IMC) method and show induction on novel diseases and novel lincRNAs that are not seen at the training time, unlike the traditional matrix factorization methods and network-based inference methods discussed earlier which are transductive in nature.
- We present a comparison of our proposed LiDiAimc method with the state-of-the-art methods on a set of OMIM disease phenotypes. The results show superiority of our method.

5.2 Methods

5.2.1 Overview of the System Architecture

The goal of a lincRNA-disease association inference method would be to compute rank of potential lincRNAs for a specific disease of interest utilizing available multiple data sources that describes both the lincRNA and disease entities and vice versa. It can be represented as predicting likelihood of an edge in a bipartite graph between a pair of putative lincRNA and disease as illustrated in Figure 5.1. Firstly, we form the lincRNA-disease association matrix $A \in \mathbb{R}^{N_l \times N_d}$, where each row cor-

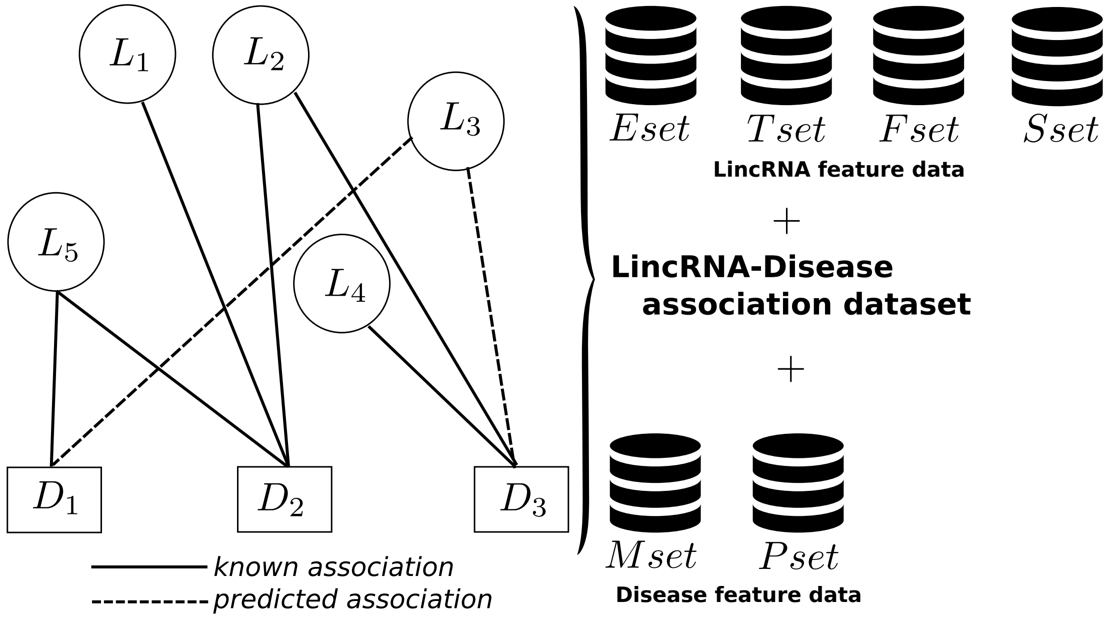


Figure 5.1: The Disease-LincRNA association problem, represented as a bipartite graph. Besides the known lincRNA-disease associations, we have considered four data sources for lincRNAs: $Eset, Tset, Fset$ and $Sset$ representing tissue expression, transcriptional regulations, functional annoations and SNP associations respectively, and two data sources for diseases: $Mset$ and $Pset$ representing OMIM term frequency inverse document frequency information and phenotypic similarity respectively. Further discussions on each of the data sources can be found in Section 5.3.

responds to a lincRNA of total N_l lincRNAs and each column to a disease of total N_d diseases, such that $A_{ij} = 1$ if lincRNA i is associated with disease j and 0 if the relationship is not known. Secondly, we prepare the feature matrix for the lincRNAs $X \in \mathbb{R}^{N_l \times f_l}$, where each row represents the conglomeration of f_l features about a lincRNA obtained through multiple data sources. Similarly, we prepare the feature matrix for the diseases $Y \in \mathbb{R}^{N_d \times f_d}$, where each row represents the f_l features of the corresponding to a disease from several data sources. Now, we look for developing a lincRNA-disease association inference method utilizing the three data matrices A, X and Y, that would be able to provide better generalization of the integration approaches offered by the state-of-the-art methods, as well as solve the cold-start problem by offering predictions of novel lincRNAs and novel diseases that might not be seen by the model during the learning step.

5.2.2 LiDiAimc Method

The most popular solution approach to this association prediction problem would be the low rank matrix completion algorithm. In that case the objective is to recover the underlying low rank matrix by using the observed entries of A, which is typically formulated as:

$$\min_{W,H} \quad \frac{1}{2}||A - WH||_F^2 + \frac{\lambda_1}{2}||W||_F^2 + \frac{\lambda_2}{2}||H||_F^2, \tag{5.1}$$

where $W \in \mathbb{R}^{N_l \times r}$ and $H \in \mathbb{R}^{N_d \times r}$ with r being the dimension of the latent feature space for both the lincRNAs and the diseases, and λ_1, λ_2 are the two regularization parameters. This approach was adapted in similar disease association problems as in [11] and in collaborative filtering methods for recommender systems by [105].

However, the formulation 5.1 is restricted to the transductive setting, i.e., predictions can only be made to existing lincRNAs and diseases available at training. As outlined in the work by [106] that the formulation can not be used to predict on rows and columns of A with no known entries.

The Inductive Matrix Completion approach [104] enables us to incorporate side information of the lincRNAs and diseases. The formulation overcomes the limitation imposed by the transductive matrix completion approach. Therefore we can predict association between new lincRNAs and diseases that are not included in the A matrix during the training time. The trick is to obtain two factor matrices W and H to define the latent feature space for the corresponding features of lincRNAs and diseases respectively rather than the lincRNA and disease entities themselves.

Let the matrix $X \in \mathbb{R}^{N_l \times f_l}$ be the training feature matrix of the N_l lincRNAs where the i^{th} row $\mathbf{x}_i \in \mathbb{R}^{f_l}$ denotes the feature vector of lincRNA i. Similarly, the matrix $Y \in \mathbb{R}^{N_d \times f_d}$ represents the feature matrix of the N_d diseases where the j^{th} row $\mathbf{y}_j \in \mathbb{R}^{f_d}$ denotes the feature vector of disease j. Here, the IMC approach tries to recover a low-rank matrix $Z \in \mathbb{R}^{f_l \times f_d}$ using the observed entries of A, and the X and Y feature matrices. The entry A_{ij} is modeled as $\mathbf{x}_i^T Z \mathbf{z}_j$. By forming Z as WH^T, where $W \in \mathbb{R}^{f_l \times r}$ and $H \in \mathbb{R}^{f_d \times r}$, the problem can be solved by obtaining the two factors W and H through solving the following optimization problem:

$$\min_{W,H} \quad \varphi = \frac{1}{2}||A - XWH^TY^T||_F^2 + \frac{\lambda_1}{2}||W||_F^2 + \frac{\lambda_2}{2}||H||_F^2, \\ \text{such that,} \quad W \geq 0, H \geq 0 \tag{5.2}$$

where λ_1, λ_2 are the regularization parameters that trade off accrued loss on the observed entries and the trace norm constraint.

Once the W and H matrices are obtained, given a new lincRNA i' that was not part of the training data, the prediction $A_{i'j}$ can be computed for a disease j as long as we have feature vector $\mathbf{x}_{i'}$, using the model as: $A_{i'j} = \mathbf{x}_{i'} WH^T \mathbf{y}_j$. Similarly, prediction can also be made for a new disease j' with an lincRNA in the set using new feature vector $\mathbf{y}_{j'}$ by $A_{ij'} = \mathbf{x}_i WH^T \mathbf{y}_{j'}$. However, the prediction between a new disease (j') and a new lincRNA (i') can also be computed through using their corresponding feature vectors, $\mathbf{x}_{i'}$ and $\mathbf{y}_{j'}$ through: $A_{i'j'} = \mathbf{x}_{i'} WH^T \mathbf{y}_{j'}$.

5.2.3 Dimensionality Reductions for the Features

As most of the data sources forming the feature matrices X and Y of the lincRNAs and diseases respectively are high dimensional, we applied principal component analysis (PCA) on each of these to construct robust and useful lincRNA and disease feature matrices. For instance, consider the lincRNA and single nucleotide polymorphism (SNP) dataset, $Sset \in \mathbb{R}^{N_l \times N_\psi}$ of N_l lincRNAs and N_ψ SNPs. Let, $U \in \mathbb{R}^{N_l \times N_\zeta}$ denotes the matrix of eigen vectors corresponding to the top N_ζ latent features of the i^{th} lincRNA. We performed PCA on functional annotation dataset ($Fset$), SNP association dataset ($Sset$) of the lincRNAs, and the OMIM term frequency inverse document frequency (TF-IDF) dataset ($Mset$) , similarity matrix dataset ($Pset$) of the disease entities in order to obtain low-dimensional feature matrices X and Y for the lincRNAs and diseases respectively.

5.2.4 Optimization through Stochastic Gradient Descent

The objective function in Equation 5.2 is non-convex in both W and H together. Therefore, it is unrealistic to expect an algorithm to find the global minima [107]. In this section and the following section we introduce two iterative algorithms which can obtain local minima to the function: Stochastic Gradient Descent (SGD) and non-linear Conjugate Gradient Descent (CGD). The solvers use alternating minimization to optimize Equation 5.2 by fixing W and solve for H and vice versa (Equation 5.3, 5.4). Since the loss function used in Equation 5.2 is Frobenius norm, the objective function of one variable (W or H) becomes convex and can be solved using the two algorithms.

$$
\begin{aligned}
& & \text{We have fixed } W^{(t)} \text{ after the } t^{\text{th}} \text{ iteration.} \\
H^{(t+1)} & \leftarrow & \arg\min_{H} \quad \varphi(:, H) \qquad (5.3) \\
& & \text{We now have fixed } H^{(t+1)} \\
W^{(t+1)} & \leftarrow & \arg\min_{W} \quad \varphi(W, :) \qquad (5.4)
\end{aligned}
$$

First we discuss how to minimize the function φ using multiplicative update rules obtained through the SGD method. The function φ can be rewritten as:

$$\begin{aligned}\varphi &= \frac{1}{2}\mathrm{Tr}\left(\left(A - XWH^TY^T\right)^T\left(A - XWH^TY^T\right)\right)\\ &\quad +\frac{\lambda_1}{2}\mathrm{Tr}\left(W^TW\right) + \frac{\lambda_2}{2}\mathrm{Tr}\left(H^TH\right)\\ &= \frac{1}{2}\mathrm{Tr}\left(A^TA\right) + \frac{1}{2}\mathrm{Tr}\left(YHW^TX^TXWH^TY^T\right)\\ &\quad -\mathrm{Tr}\left(A^TXWH^TY^T\right) + \frac{\lambda_1}{2}\mathrm{Tr}\left(W^TW\right) + \frac{\lambda_2}{2}\mathrm{Tr}\left(H^TH\right)\end{aligned} \tag{5.5}$$

Now, let ψ_{ik}, ϕ_{jk} be Lagrange multipliers for the constraints $W_{ik} \geq 0$ and $H_{jk} \geq 0$ respectively. And $\Psi = (\psi_{ik})$ and $\Phi = \left(\phi_{jk}\right)$. Then, the Lagrange $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}\mathrm{Tr}\left(A^TA\right) + \frac{1}{2}\mathrm{Tr}\left(YHW^TX^TXWH^TY^T\right) && (5.6)\\ &\quad -\mathrm{Tr}\left(A^TXWH^TY^T\right) + \frac{\lambda_1}{2}\mathrm{Tr}\left(W^TW\right) + \frac{\lambda_2}{2}\mathrm{Tr}\left(H^TH\right)\\ &\quad +\mathrm{Tr}\left(\Psi^TW\right) + \mathrm{Tr}\left(\Phi^TH\right) && (5.7)\end{aligned}$$

The partial derivatives of the Lagrange $\mathcal{L}$ with respect to W and H are:

$$\frac{\partial\mathcal{L}}{\partial W} = -X^TAYH + X^TXWH^TY^TYH + \lambda_1 W + \Psi \tag{5.8}$$

$$\frac{\partial\mathcal{L}}{\partial H} = -Y^TA^TXW + Y^TYHW^TX^TXW + \lambda_2 H + \Phi \tag{5.9}$$

Now by applying the KKT conditions $\psi_{ik}W_{ik} = 0$ and $\phi_{jk}H_{jk} = 0$ we get the following equations of W_{ik} and H_{jk} respectively:

$$\begin{aligned}&-\left(X^TAYH\right)_{ik}W_{ik} + \left(X^TXWH^TY^TYH\right)_{ik}W_{ik}\\ &\qquad +\lambda_1\left(W\right)_{ik}W_{ik} = 0\end{aligned} \tag{5.10}$$

$$\begin{aligned}&-\left(Y^TA^TXW\right)_{jk}H_{jk} + \left(Y^TYHW^TX^TXW\right)_{jk}H_{jk}\\ &\qquad +\lambda_2\left(H\right)_{jk}H_{jk} = 0\end{aligned} \tag{5.11}$$

These equations lead to the following update rules:

$$W_{ik} \leftarrow W_{ik} \frac{\left(X^T A Y H\right)_{ik}}{\left(X^T X W H^T Y^T Y H + \lambda_1 W\right)_{ik}} \tag{5.12}$$

$$H_{jk} \leftarrow H_{jk} \frac{\left(Y^T A^T X W\right)_{jk}}{\left(Y^T Y H W^T X^T X W + \lambda_2 H\right)_{jk}} \tag{5.13}$$

We have the following theorem corresponding to these two update rules.

Theorem 5.2.1. *The objective function of the LiDiAimc problem (Equation 5.2) is nonincreasing under the update rules (Equation 5.12 and 5.13).*

In the following section we provide a detailed proof of Theorem 5.2.1. Our proof follows the sketches from the proof of the original Non-negative Matrix Factorization function by [79].

Finally, we arrive at a point when it is appropriate to define Algorithm 6 to solve the objective function taking care of all the issues discussed so far in this section.

Algorithm 6 SGD to solve Equation 5.2.

```
Input: $A \in \mathbb{R}_+^{N_l \times N_d}$, rank $r$, $X \in \mathbb{R}_+^{N_l \times f_l}$, $Y \in \mathbb{R}_+^{N_d \times f_d}$ and the two initial seed
    matrices $W \in \mathbb{R}_+^{f_l \times r}$ and $H \in \mathbb{R}_+^{f_d \times r}$.
Output: Calculate $W, H$ such that $A \approx X W H^T Y^T$
repeat
    Update $H$ matrix using Equation 5.13.
    Update $W$ matrix using Equation 5.12. Here we will be using the $H$ calculated
    at the previous step.
until convergence criterion is met
return $W, H$
```

5.2.5 Proof of Theorem 5.2.1

Proof. Here, we show that the objective function is non-increasing under the update rules. Specifically, here we prove that the objective function is non-increasing under the update step for H, and same feature under the update rule for W can be similarly proved. Here we adopted the same strategy applied in [79] that introduced the concept of an auxiliary function in the Expectation-Maximization algorithm. Now, we define the auxiliary function G for our proof.

Definition 5.2.1. $G(H, H')$ is an auxiliary function for the function $F(H)$ if $G(H, H') \geq F(H)$ and $G(H, H) = F(H)$.

Lemma 5.2.2. *If G is an auxiliary function of F, then F is non-increasing under the update*

$$H^{(t+1)} = \underset{H}{argmin} \quad G(H, H^{(t)}) \tag{5.14}$$

Proof.

$$\begin{aligned} F(H^{(t+1)}) &\leq G(H^{(t+1)}, H^{(t)}) \\ &\leq G(H^{(t)}, H^{(t)}) = F(H^{(t)}) \end{aligned}$$

■ □

With a proper auxiliary function, the update rule for H is exactly the required update solution to the Lemma 5.2.2. Now we use F_{ab} to denote the part of the objective function which is only relevant to H_{ab}. We get–

$$\begin{aligned} F'_{ab} &= \left(\frac{\partial \mathcal{F}}{\partial H}\right)_{ab} \\ &= \left(-Y^T R^T XW + Y^T Y H W^T X^T XW + \lambda_2 H\right)_{ab} \\ F''_{ab} &= \left(W^T X^T XW \otimes Y^T Y\right)_{aa} + (\lambda_2 I_k)_{aa} \end{aligned}$$

Here, $\otimes$ denotes the kronecker product. As the update is element-wise, it is sufficient to show that each F_{ab} is non-increasing under the update rule for H.

Lemma 5.2.3. *The function*

$$\begin{aligned} G(H, H_{ab}^{(t)}) &= F_{ab}(H_{ab}^{(t)}) + F'_{ab}(H_{ab}^{(t)}) \left(H - H_{ab}^{(t)}\right) \\ &+ \frac{\left(\left(W^T X^T X W \otimes Y^T Y + \lambda_2 I_k\right) H\right)_{ab}}{H_{ab}^{(t)}} \left(H - H_{ab}^{(t)}\right)^2 \end{aligned} \tag{5.15}$$

is an auxiliary function for F_{ab}.

Proof. Obviously the first criterion of an auxiliary function is met, because $G(H, H) = F_{ab}(H)$. We only need to show the second criterion, that is $G(H, H_{ab}^{(t)}) \geq F_{ab}(H)$.

Let us compare the Taylor series expansion of the function $F_{ab}(H)$:

$$\begin{aligned} F_{ab}(H) &= F_{ab}(H_{ab}^{(t)}) + F'_{ab}(H_{ab}^{(t)}) \left(H - H_{ab}^{(t)}\right) \\ &+ \frac{F''_{ab}(H_{ab}^{(t)})}{2!} \left(H - H_{ab}^{(t)}\right)^2 + \cdots + \text{ignoring higher orders} \end{aligned}$$

And so,

$$\begin{aligned} F_{ab}(H) &= F_{ab}(H_{ab}^{(t)}) + F'_{ab}(H_{ab}^{(t)}) \left(H - H_{ab}^{(t)}\right) \\ &+ \frac{1}{2} \left(W^T X^T X W \otimes Y^T Y + \lambda_2 I_k\right)_{aa} \left(H - H_{ab}^{(t)}\right)^2 \end{aligned} \tag{5.16}$$

Now, since

$$\begin{aligned} &\left(\left(W^T X^T X W \otimes Y^T Y + \lambda_2 I_k\right) H\right)_{ab} \\ &= \sum_k \left(W^T X^T X W \otimes Y^T Y + \lambda_2 I_k\right)_{ak} (H_{kb}^{(t)}) \\ &\geq H_{ab}^{(t)} \left(W^T X^T X W \otimes Y^T Y + \lambda_2 I_k\right)_{aa} \end{aligned}$$

That is,

$$\frac{\left(\left(W^T X^T XW \otimes Y^T Y + \lambda_2 I_k\right) H\right)_{ab}}{H_{ab}^{(t)}} \geq \left(W^T X^T XW \otimes Y^T Y + \lambda_2 I_k\right)_{aa}$$

Hence, comparing Equation 5.15 and 5.16 we have:

$$G(H, H_{ab}^{(t)}) \geq F_{ab}(H)$$

■ □

Now, we can have the following update rule based on the auxiliary function $G(H, H_{ab}^t)$, by replacing $G(H, H_{ab}^t)$ in Equation 5.14 by the results of Equation 5.15:

$$\begin{aligned} H_{ab}^{(t+1)} &= H_{ab}^{(t)} - H_{ab}^{(t)} \frac{F'_{ab}(H_{ab}^{(t)}}{(Y^T Y H W^T X^T X W + \lambda_2 H)_{ab}} \\ &= H_{ab}^{(t)} \left(1 - \frac{F'_{ab}(H_{ab}^{(t)}}{(Y^T Y H W^T X^T X W + \lambda_2 H)_{ab}}\right) \\ &= H_{ab}^{(t)} \frac{\left(Y^T R^T X W\right)_{ab}}{(Y^T Y H W^T X^T X W + \lambda_2 H)_{ab}} \end{aligned}$$

Due to the property of the auxiliary function $G(H, H_{ab}^{(t)})$ for F_{ab}, F_{ab} is non-increasing under this update rule. □ □

5.2.6 Optimization through Nonlinear Conjugate Gradient Descent

Nonlinear Conjugate Gradient Descent (CGD) [108], another prominent iterative method for solving sparse systems, like ours, is investigated here due to its popularity in recent years and is compared against the Stochastic Gradient Descent (SGD) based method developed in the previous section in terms of convergence quality. One of the best properties of the Conjugate Gradient method is its ability to generate a set of conjugate vectors efficiently; the vectors successively are used in minimizing each of the functions (Equation 5.3 and 5.4).

To solve a minimization problem for $x \in \mathbb{R}^n$ of a function $f(x)$, CGD tends to solve x successively by using: $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a stepsize obtained through a line search and d_k is the search direction defined by Equation 5.17,

$$d_k = \begin{cases} g_k & \text{for } k = 1 \\ g_k + \beta_k d_{k-1} & \text{for } k \geq 2, \end{cases} \tag{5.17}$$

where g_k is the gradient of the objective function. Here, only the stepsize α_k and the parameter β_k remain to be determined in the definition of the CGD. Since our objective function uses squared loss function, the line search for solving α_k (Equation 5.18) can be made computationally fast through using the operations to compute the Hessian (∇^2) proposed by [109].

$$\alpha_k = \arg\min_{\alpha} \quad f(x_k + \alpha d_k); \alpha > 0 \tag{5.18}$$

Different formulae for the parameter β_k result in different conjugate gradient methods. We investigated methods by [108] (CGD-FR) and [110] (CGD-PR). However, sometimes the methods may cycle infinitely without approaching a solution even if the stepsize α_k is chosen to the least positive minimizer of the line search function. The remedy to the issue is to restart CGD whenever the β parameter found to be negative by setting $d = g$, a solution suggested by [111].

Thus, to solve the equation 5.4 for optimum W we present Algorithm 7. However, a symmetric algorithm was prepared to solve equation 5.3. The computation of the Hessian $\nabla^2_W \varphi$ required at line 7 and 9 in Algorithm 7 was achieved through the fast operations developed in the study by [109].

Algorithm 7 CGD to solve Equation 5.4. Calculate W.

Input: Gradient: $\nabla_W \varphi$, Hessian: $\nabla^2_W \varphi$ of the objective function f and option: Any of the four variations of CGD to update β parameter { FR, PR }

Output: W

```
Initialize $W_0$ to any seed; $g_0 = -\nabla_W \varphi(W_0)$; $d_0 = g_0$
$k = 0$
repeat
    if $||g_k||$ is small then
        return $W_k$
    else
        $\alpha_k = \frac{g_k^T g_k}{d_k^T \nabla^2_W \varphi(W_k) d_k}$
        $W_{k+1} = W_k + \alpha_k d_k$
        $g_{k+1} = g_k - \alpha_k \nabla^2_W \varphi(W_{k+1}) d_k$
        if option is Fletcher Reeves (FR) then
            $\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$
        else if option is Polak-Ribière (PR) then
            $\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}$
        end if
        $\beta_k = \max\{0, \beta_k\}$ //[111]
        $d_{k+1} = g_{k+1} + \beta_k d_k$
        $k = k + 1$
    end if
until Convergence criterion is met
return $W_k$
```

5.3 Experiments

5.3.1 Disease-LincRNA Association dataset

We obtained human lincRNA-disease associations by combining the LncRNADisease database [67] and the supporting dataset from the co-expression based association study conducted by [103]. The combined dataset contains 46,934 associations among 8194 lincRNA genes and 1213 diseases. Since none of the two datasets adapted standard naming of the diseases, we retrieved top-5 closely matched OMIM phenotypes for each of the disease names from the pool using OMIM API [76], and prepared the association matrix between 8194 lincRNA genes and 2661 OMIM phenotypes. The matrix is very sparse having only 0.22% non-zero entries. To compare different approaches on the novel association prediction, we use use 10-fold cross validation over the association dataset.

5.3.2 LincRNA Feature datasets

RNA-seq provided Expression profiles of lincRNAs on different tissues underline the impact of the lincRNAs for diseases occurring corresponding tissues. Although not all diseases are tissue-specific, neither are the lincRNAs, the profiles still can be used to distinguish between co-expressed lincRNAs to implicate diseases. RNA-seq measurement of 8194 lincRNA expression levels on 22 human tissues are obtained from the Human BodyMap Project 2.0 [72]. Expression scores are represented in terms of FPKM values (Fragments Per Kilobase of exons per Million Fragments mapped).

ChIP-seq provided Transcription Factor Binding Sites (TFBS) of the lincRNAs unravel the transcriptional regulatory relationships of lincRNAs with transcription factors. We obtained 160,588 relationships among the 8194 lincRNAs and 120 transcription factors from ChIP-Base dataset [112]. There are only 217 lincRNAs have relationship with one transcription factor, and out of 120 transcription factors, the minimally related transcript factor, “BACH1” has 11 lincRNA connections, and there are 6130 lincRNAs connecting with a transcript factor, “HNF4A”.

Functional annotations of the lincRNAs dictate their characterizations and involvement on various biological activities inside human cells that implicitly correlate with various disease phenotypes. Linc2GO [113] presents a database of such annotations of lincRNAs based on the ceRNA hypothesis [114]. We retrieved 8111 GO BP (Biological Process) terms, 3218 GO MF (Molecular Function) terms and 193 KEGG Pathway terms associated with the 8194 lincRNAs from the database, resulting a total of 11522 functional terms for each lincRNA in our study. However, this annotation matrix is also sparse, having 0.11% non-zero entries. We use the leading 100 singular vectors of the matrix as the representative features of the lincRNAs contributed from the Linc2GO dataset.

Single Nucleotide Polymorphisms (SNPs) in lincRNAs were found to be linked to their abnormal expressions and dysregulations, thereby playing key roles in various phenotypes and diseases [115]. The lncRNASNP dataset [5] provides a comprehensive resource of SNPs in human lncRNAs, and we extracted 368,494 SNPs in the 8194 lincRNAs from the database. The SNP-lincRNA association is sparse with 0.0077% non-zero entries. We use the leading 100 singular vectors of the matrix as the representative features of the lincRNAs contributed from the lncRNASNP dataset.

Finally, we considered only those lincRNAs having all these four types of features. Therefore, we ended up having a catalog of 6540 lincRNAs with corresponding features.

5.3.3 Disease Feature datasets

Term Frequency Inverse Document Frequency (TF-IDF) of the 2661 OMIM phenotypes obtained from the OMIM text corpus provides a standard statistic that reflect how important a term is to a OMIM phenotype text collection. The TF-IDF score increases proportionally to the frequency of occurrences of a term in a particular page, but is offset by the frequency of the term in the whole corpus. This phenomenon helps to identify important keywords associated appearing only in the corresponding OMIM page, as well as less important terms appearing most of the pages. The number of terms considered in the scheme is 20491, thus resulting in a TF-IDF matrix of size 2661 by 20491. We use the leading 100 singular vectors of the matrix as the representative features of the diseases contributed from the OMIM TF-IDF dataset.

Phenotypic similarity profiles of the lincRNAs were retrieved from a recent study by [116], where the authors developed a method to accumulate the MeSH terms associated with the publications referenced in the OMIM phenotype pages and able to compute scores that reflect the molecular relatedness between two OMIM entries. The similarity matrix thereby is symmetric of dimension 2661 by 2661. We reduce the dimensionality of the feature space using PCA, retaining the top 100 principal components.

We considered only those diseases having all these two types of features. Thus, we ended up having a catalog of 2148 diseases with corresponding features.

5.3.4 Baselines

We compare the results of our proposed LiDiAimc method with four approaches. Firstly, the standard non-negative matrix factorization on the lincRNA-disease association matrix A from Equation 5.2. This becomes a special case for the Inductive

Matrix Completion objective where the lincRNA feature matrix (X) and the disease feature matrix (Y) are set to identity. We compare three other methods that provide interfaces to scale their corresponding framework to a much larger dataset like ours (i.e., considering associations among the 6540 lincRNAs and 2418 disease phenotypes). The approaches differ much in solving the association problems, from significant test for associating identities, solving the Graph Laplacian Regularized Least Squares, to Kernalized Random Walk with restart approach. We describe each of the four methods in more detail below.

5.3.4.1 NMF [79]

Here, we consider matrix completion on the bipartite network A and solve the optimization Equation 5.1 enforcing non-negativity constraints over W and H using Alternating Least Squares (ALS) method. The standard matrix completion formulation does not accommodate any of the side information available about the lincRNAs and diseases. After the convergence of NMF Algorithm we retrieved rank of the predictions using the estimated values of the matrix, i.e., higher the estimated A_{ij}, more relevant is the lincRNA i for disease j.

5.3.4.2 LRLSLDA [68]

The Laplacian Regularized Least Squares for LncRNA-Disease Association (LRLSLDA) method is based on preparing two similarity matrices. Firstly, the lincRNA-lincRNA similarity matrix was built through the integration of the pairwise expression correlation obtained from the same dataset [72] we used in this study and disease implication matrix. Secondly, the disease-disease similarity matrix, the pairwise source implicator lincRNA matrix. LRLSLDA builds two separate classifiers to compute probability of disease-lincRNA association in the lincRNA space and the disease space respectively. The two probabilities obtained through the two classifiers were combined by a mean operation. The final probability score, F^*_{ij} reflects the probability that lincRNA j is related to the disease i. The computationally expensive operation in LRLSLDA is during the pairwise similarity matrix constructions which prohibits its usability in scalable framework development. Moreover, there are eight parameters used in LRLSLDA, which comparatively is a large number to tune in order to make the method computationally efficient.

5.3.4.3 TsLincRNA-Disease [103]

Tissue - Specificity based LincRNA - Disease association prediction framework (TsLincRNA - Disease) draws a demarcation line between tissue-specific and non-tissue-specific classes utilizing the tissue-specificity index for each of the lincRNAs in the study. The tissue expression dataset [72] for each of the lincRNAs are obtained through the same source as in our study. The tissue-specific lincRNAs undergo a statistical significance test to be marked as disease causing in the particular tissue in its profile. However, for identifying relationships among the diseases and the non-tissue-specific lincRNAs a human lincRNA-gene co-expression network is first constructed utilizing publicly available gene expression profile dataset, gene-disease association datasets, then a mean enrichment analysis for the set of genes co-expressed with each lincRNA follows predicting association with the diseases in the study.

5.3.4.4 K-RWRH [102]

Kernel-based Random Walk with Restart method in a heterogeneous network is an extension to the RWRH algorithm proposed in [117]. Here the heterogeneous network is constructed by a disease-disease similarity matrix, lincRNA-lincRNA similarity matrix and known lincRNA-disease relationship matrix. It predicts potential lincRNA-disease association through simulating the random walk with restart from a given set of known disease and lincRNA seed nodes. After some steps, the steady state probability distribution is obtained. The lincRNAs and the diseases (representing the nodes in the network) are ranked based on the steady probabilities.

Note that the first method does not use any of lincRNA or disease specific features such as TF-IDF, disease phenotype similarity, RNA-seq provided expression profiles, ChIP-seq provided Transcription factor binding sites, functional GO annotations and SNP linkages. However, remaining three methods also do not use none of these features except the expression profiles. For all the methods, including our LiDiAimc we rank the predictions using the estimated values corresponding to a lincRNA for each of the diseases considered in our study. For the LiDiAimc method, we construct the lincRNA and disease feature matrices $X \in \mathbb{R}^{N_l \times f_l}$ with $f_l = 342$ and $Y \in \mathbb{R}^{N_d \times f_d}$ with $f_d = 200$ for the set of $N_l = 6540$ lincRNAs and $N_d = 2418$ diseases (in terms of OMIM phenotypes). We set the best parameters values for each of the methods through cross-validation except LRLSLDA, in which case we set the eight parameter values as suggested by the authors. All the experiments were run on an Intel(R) Core (TM) i5-2400 CPU running at 3.10GHz, 4-cores, 6MB L2-cache, 12GB of RAM (DDR3 1333) hosting Ubuntu 14.04 operating system

5.3.5 Evaluation Metrics

The lincRNA-disease association prediction algorithm under evaluation computes a ranking score for each candidate disease (i.e., disease that is not reported to be connected with a lincRNA before) and returns the top-k highest ranked diseases as recommendations to a target lincRNA. Thus, for the evaluation of the predictive accuracy, the goal is to find out how many disease-lincRNA associations previously marked off in the preprocessing step recovered in the returned disease recommendations. More specifically, we used two evaluation metrics: (1) the ratio of recovered diseases to the k recommended diseases for the target lincRNA, and (2) the ratio of recovered diseases to the set of diseases deleted in preprocessing [118]. The first metric is called *precision*@k and the latter is known as *recall*@k. The metrics are defined in Equation 5.19 and 5.20. In our experiment, we tested the performance when $k = \{5, 10, 20, 40, 50, 60, 70, 80, 90, 100\}$.

$$precision@k = \frac{1}{N_l} \sum_{l=1}^{N_l} \frac{|P_l(k) \cap D_l|}{k} \tag{5.19}$$

$$recall@k = \frac{1}{N_l} \sum_{l=1}^{N_l} \frac{|P_l(k) \cap D_l|}{|D_l|}, \tag{5.20}$$

where $P_l(k)$ being the top-k ranked diseases for lincRNA l, D_l is the set of diseases related to the lincRNA l marked off during the training step, N_l is the total number lincRNAs in the evaluation dataset. We performed 10-fold cross-validation to measure the performance of our proposed LiDiAimc method as well as the competitive four methods. It is worth noting that the *precision*@k and *recall*@k in our experiments are not high. This is because of the sparsity in the lincRNA-disease association dataset having density only 0.003. Similar performance can also be observed in other association recommendation works by [119] and [120] just to name a few. Therefore, the low precision obtained in our experiment is reasonable. In this article, we emphasize on comparing relative performance of the methods rather than their absolute performance.

5.4 Results and Discussions

5.4.1 Effects of the Parameter Settings

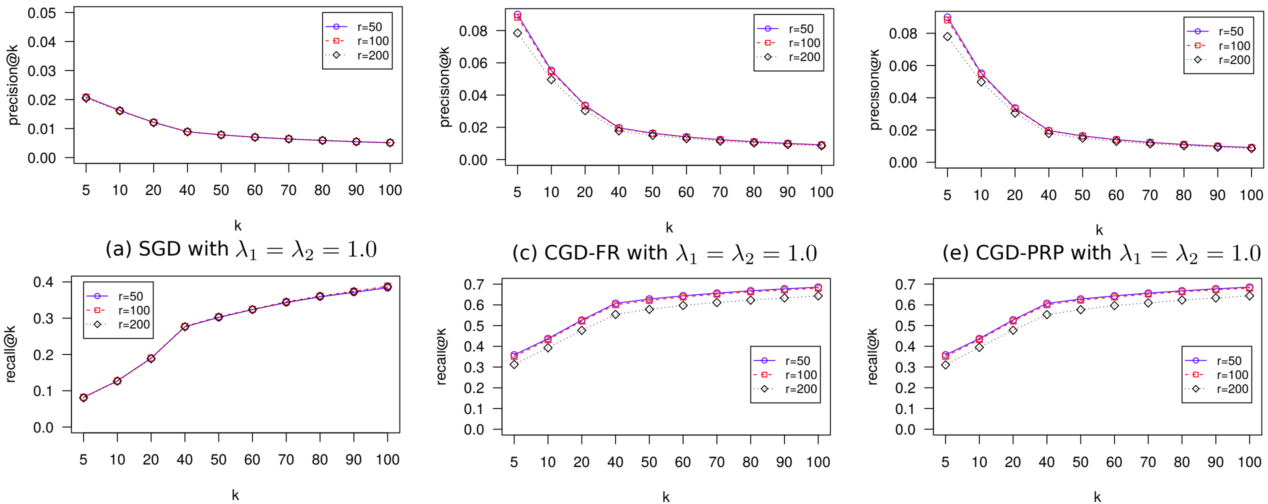
The parameters to the LiDiAimc method are the rank (r) of the basis, W and the coefficient, H matrices and the regularization parameters λ_1, λ_2 for W and H

matrices respectively. The *precision*@k and *recall*@k performance of the LiDiAimc with the three solution schemes (SGD, CGD-PR and CGD-FR) are presented in Figure 5.2. It is evident from the figure that the top-k association retrieval performance varies with the changes in the rank parameter. We varied the rank parameter from 50 to 200, which is equal to $\min(\text{rank}(X), \text{rank}(Y))$, where X, Y are the lincRNA and disease feature matrices respectively. We see that there is little to no boost up of the performance from the changes of rank value from 50 to 100. However, continuing to increase the rank to the maximum possible value, 200, the performance degrades in terms of both the *precision*@k and *recall*@k. This issue can be justified as an overfitting problem.

We set the regularization parameters $\lambda_1 = \lambda_2 = 1.0$ in the optimization function (Equation 5.2) as we found that the predictive performance degrades if the parameters are set to values deviating far away from 1.0 (data not shown). We found the cut-off value for the parameters through cross-validation for all possible values in the range (0.1, 10.0).

5.4.2 True LincRNA-Disease Association Retrieval

The 10-fold cross-validation results on 2418 OMIM diseases are presented in Figure 5.3. The Y-axis in the plots (a,b) gives the *precision*@k and *recall*@k scores for various k values in the horizontal X-axis. We observe that most of proposed LiDiAimc variants significantly dominate the competitive methods over all k values. The best *precision*@k and *recall*@k recorded are close to 10% and 38% respectively at the top-5 association prediction cases and are obtained through utilizing the Polak-Ribière (PR) approach for solving through CGD, and also the Fletcher-Reeves (FR) approach.



(a) SGD with $\lambda_1 = \lambda_2 = 1.0$ (c) CGD-FR with $\lambda_1 = \lambda_2 = 1.0$ (e) CGD-PRP with $\lambda_1 = \lambda_2 = 1.0$

(b) SGD with $\lambda_1 = \lambda_2 = 1.0$ (d) CGD-FR with $\lambda_1 = \lambda_2 = 1.0$ (f) CGD-PRP with$\lambda_1 = \lambda_2 = 1.0$

Figure 5.2: Performance of LiDiAimc for different values of rank parameter, r. (a-b): SGD based solution shows a slight improvement of precision@k and recall@k values for increasing the rank parameter, r, (c-d): For Fletcher-Reeves (FR) solution, $r = 50$ seems to out-performs the others. (e-f): In the Polak-Ribière (PR) solution, again the $r = 50$ shows better performance than with any rank lower or higher than that.

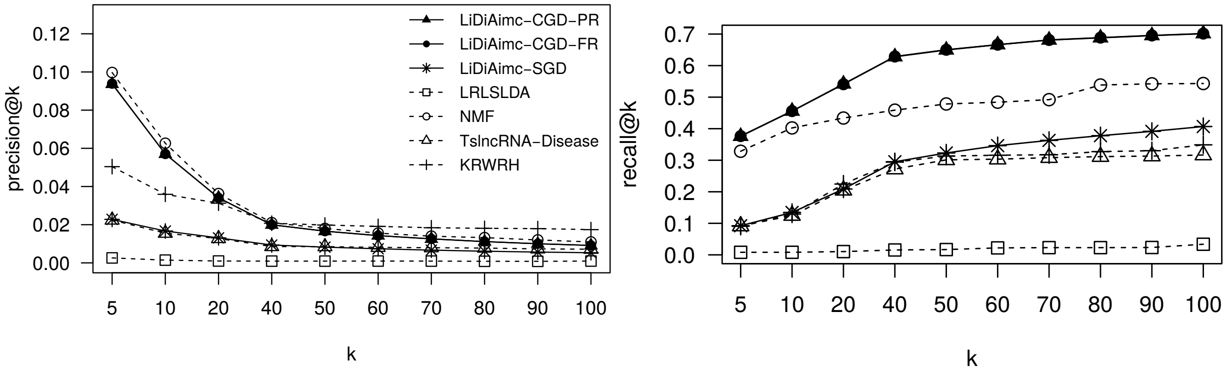


Figure 5.3: Comparision of lincRNA-disease association methods. (a) k-vs-*precision@k* plot for all the seven methods. The three solid lines represent three of our proposed solutions from the LiDiAimc model, and the dotted four lines denote the performance lines of the four competitive methods. (b) k-vs-*recall@k* plot for the nine methods. The proposed LiDiAimc method (the three variants) are trained with 342 lincRNA features and 200 disease features, with a rank, $r = 100$. NMF was trained with the same binary association matrix we used in LiDiAimc with a rank $r = 100$.

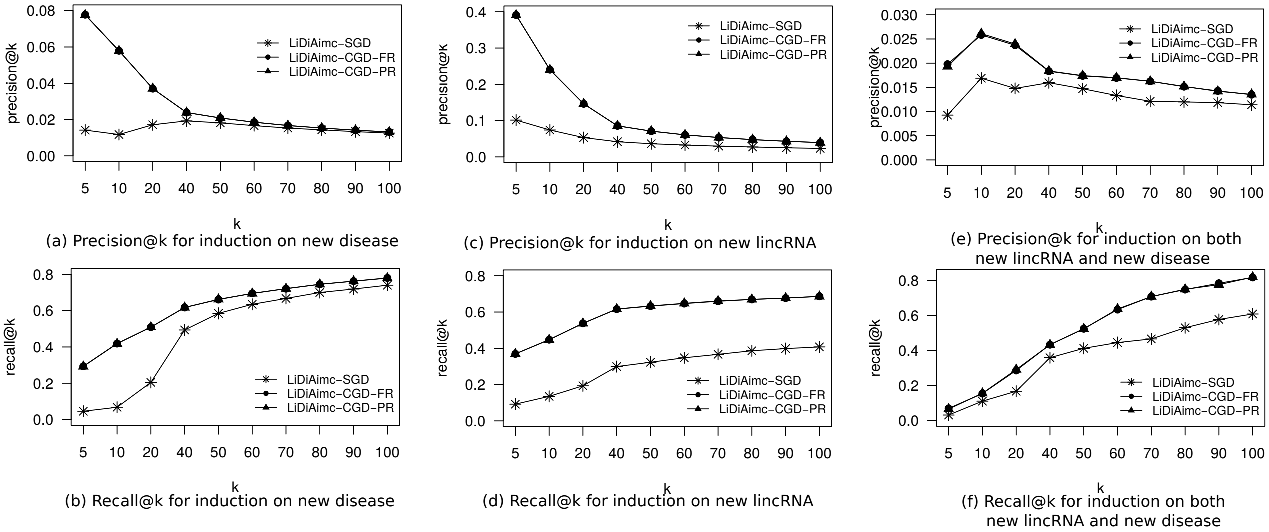
The matrix completion on A performs significantly better than the three other baseline algorithms. LRLSLDA performs worse in terms of $precision@k$ and the $recall@k$ scores. This is because, the method only relies on known association matrix and the expression profiles of the lincRNAs. Moreover, it comes with a lot of parameters to learn, and is not easily scalable in larger context, like ours, because of the complex `pinv` operations to compute the Laplacians. Among the two approaches to solve the LiDiAimc, SGD thrives to top the two competitive methods, namely LRLSLDA and TslncRNA-disease in terms of $precision@k$, but successfully dominates over the three methods (LRLSLDA, KRWRH and the TslncRNA-disease) in terms of $recall@k$.

5.4.3 Induction on new Associations and Case Studies

Next, we investigate the power of the inductive learning which we introduced in this study via LiDiAimc method to predicting the associations between (i) a new disease to a well studied lincRNA, (ii) a new lincRNA to a well studied disease and (iii) a new disease to a new lincRNA. In order to investigate these three features offered by a recommendation system like ours, we randomly picked 10% of the subject lincRNA entries and the corresponding associations from our datasets (X, the lincRNA featureset and A, the lincRNA-disease association data matrix for case i and iii above) and the subject disease entries and the respective associations from Y, the disease featureset and A matrix for case ii and iii above as new test samples. LiDiAimc was then trained with the remaining entries and associations. We evaluate each of the models with the respective set-aside test cases. We repeat the above steps 10 times and recorded the average predictive scores for each of the solution strategies we proposed for LiDiAimc. The only assumption in the LiDiAimc for induction is that all the features for the novel disease (or the lincRNA or both) may be available during prediction. Here we underline the power of inductive learning of the trained models which is readily usable for prediction for a new test lincRNA or disease (or both) entries even though the entries were absent during the training. Note that all the baseline methods are missing from each of the plots in Figure 5.4 as none could make such prediction of the novel disease and lincRNA associations using the respective learned models because of their inherent transductive formulations.

Figure 5.4(a,b) illustrates the performance of our three solution approaches for LiDiAimc on new diseases. The *precision@k* and *recall@k* curves for both the Fletcher-Reeves (FR) and Polak-Ribière (PR) methods for the CGD based LiDiAimc almost superimpose each other and both present superior performance than that of the SGD based approach for predicting upto the top-50 lincRNA associations with the novel diseases. CGD based strategies are also seen performing better than the SGD for induction on the novel lincRNA (Figure 5.4 c,d) and both (Figure 5.4 e,f).

Finally, we applied LiDiAimc to prioritize all the 6,540 candidate lincRNAs for each of the 2,418 OMIM disease phenotypes under investigated in the study. All the known lincRNA-disease associations were treated as ground truth dataset and were used as training associations. The top-20 predicted diseases for each of the lincRNAs are publicly released to benefit experimental validation from biologists (please check the availability URL). According to the predictive result for lincRNAs, the transcript TCONS_00000721 (gene XLOC_001186) are associated to ovarian diseases (OMIM IDs: 184700, 311360, 615723) which is confirmed by [121] through relating the protein coding EXD3 gene which is in the vicinity of the lincRNAs. The lincRNAs TCONS_00000895 (gene: XLOC_000148) and TCONS_00001488 (gene: XLOC_000824) are predicted to be linked to testes disorders. It is verified through Gene Expression Atlas database by the two related protein coding genes: ZNF502 and DCAF16. The lincRNA TCONS_00013953 (gene id: XLOC_006604) is predicted



(a) Precision@k for induction on new disease

(c) Precision@k for induction on new lincRNA

(e) Precision@k for induction on both new lincRNA and new disease

(b) Recall@k for induction on new disease

(d) Recall@k for induction on new lincRNA

(f) Recall@k for induction on both new lincRNA and new disease

Figure 5.4: Performance of LiDiAimc for induction on novel diseases, novel lincRNAs and both. (a-b): Inductive performance of the three solutions to the LiDiAimc in predicting associations of new diseases with a well studied set of lincRNAs, (c-d): Inductive performance of the three solutions to the LiDiAimc in predicting associations of new lincRNAs with a well studied set of diseases. (e-f): Inductive performance of the three solutions to the LiDiAimc in predicting associations of new diseases with a set of new lincRNAs.

to be associated with breast cancer (OMIM ID: 604370) because of the NRF1 coding gene which has true associations with the disease [122].

5.5 Conclusions and Future Research Scopes

In this manuscript, we have proposed a novel method, LiDiAimc, for predicting associations between the long intergenic non-coding RNAs (lincRNAs) and diseases. The method presents an integration interface for various categories of features of both the lincRNAs and diseases obtained through different independent data sources for explaining the relationships between the two entities, as no single data source can potentially capture all the relevant relations. We investigated three solution approaches to develop our method and presented results of a comprehensive analysis of our LiDiAimc approach underlining the fact that the choice of the features of the lincRNAs and diseases is the best as well as the integration framework performs superior than the competitive methods LRLSLDA, KRWRH and TslncRNA-disease (which rely on the lincRNA expressions and true association matrix) and the NMF (that relies only on the true associations). In our experiments we find that LiDiAimc method performs the best in predicting associations between already studied set of lincRNAs and diseases as well as between novel set of lincRNAs and diseases which makes the method a suitable association prediction tool for the biologists.

Several possible extensions to the LiDiAimc method presented here can be made: the inductive framework (as opposed to its transductive versions) is not limited to the types of features used in the experiments we presented, as new sources of information can be integrated easily via rank-1 updates. The framework itself can be extended to address the sparsity issue inherent to the true association matrix.

5.6 Availability

The dataset and the association prediction results are all available at `http://biomecis.uta.edu/~ashis/res/LiDiAimc`

CHAPTER 6

LincRNA-Disease Associations through Robust Inductive Matrix Completion

6.1 Introduction

LincRNA-Disease association inference problem: It is a surprising fact that, only 2% of the entire human genome codes for protein [100]. In recent years, it has become evident that the non-protein coding portion of the genome, especially the long intergenic non-coding RNAs (lincRNAs) having length more than 200 bases each with no overlaps with any annotated protein-coding regions, are of critical functional importance for their diverse molecular mechanisms and implications of various human diseases [101]. With the advent of the high-throughput genomic technologies, such as RNA-seq and ChIP-seq, a huge number of lincRNAs have been cataloged. But, characterizing their functions, predicting the associations of the lincRNAs to human diseases, remain a challenge [72]. *In silico* association inference tools would present, in this regard, an important facet towards discovering causal lincRNA-disease relationships and better understanding of the human diseases. Such tools would be able to rank disease implications by a given lincRNA based on prior knowledge.

Limitations of existing methods: There are several long non-coding RNA (lncRNA)-disease association inference tools developed in the past few years. Unfortunately, only a few have dealt with the lincRNA-disease inference problem. Due to the intricacies inherent to this problem, only a small number of experimentally validated associations have been reported in the publicly available databases, such as lncRNAdisease [67]. For this reason, leveraging multiple complementary data sources is essential for predicting lincRNAs related to diseases as well as respective phenotypes, and thus different inference methods have been developed considering different knowledge sources. For instance, K-RWRH [102], LRLSLDA [68] and TslncRNA-disease [103] are popular family of network based methods. The methods utilize biological networks, such as lincRNA similarity network and disease similarity network and infer lincRNA-disease connections by either using random walk procedure on a derived biological network or by computing a similarity measure between nodes with known disease implications. The association inference problem can also be solved using a matrix completion approach. These approaches suffer from the cold start problem, due to the inability to address the inference predictions of the diseases for novel lincRNAs and vice versa. Furthermore, these methods were presented on a very small set of associations and developed without considering the scalability (e.g., around 200 lncRNAs compared to more than 8000 lincRNAs available to date from [72] research remain overlooked). However, the methods exploiting lincRNA-expression profiles to build similarity networks may only deal with specific disease classes that are only available through the seed or true associations and therefore the methods fall short in generalizing to novel diseases. Owing to the fact that, a plethora of side information about the lincRNAs and the disease phenotypes are available, and the data is growing extensively every single day. The standard Inductive Matrix Completion (IMC) can take into account these side information along with the known association evidences to predict missing associations [104]. But, the standard IMC uses the least square error function that is well known to be unstable with respect to noises and outliers present in the dataset [123]. However, the side information about the lincRNAs and the diseases possibly contain noise and outliers. To deal with such situation, a robust IMC is needed.

Outline of our proposed approach: We propose a novel robust formulation of IMC using $\ell_{2,1}$ norm penalty function, as well as $\ell_{2,1}$ based regularization. The proposed method is called "robust" as it can handle outliers and noises better than the standard IMC. Also, it can handle joint sparsity, i.e., handle appropriately the feature set, where each feature either has small values for all data points or has large values over all data points.

Summary of contributions:

- We propose Robust IMC that can handle outliers and noises in the dataset, along with the sparsity consideration. We derive the computational algorithm and provide a correctness proof of the algorithm.
- We provide an application of our Robust IMC method to solve the lincRNA-disease association inference problem. We show that RIMC can perform induction to decipher associations between a novel disease and a novel lincRNA, based on the side information about them we have, that are not provided during learning phase. This is unlike the traditional matrix factorization methods and network-based inference methods discussed earlier which are transductive in nature.
- We demonstrate that the integration of diverse features of the lincRNAs and the diseases available through publicly available data-servers can overcome worse predictive performance issue faced by the inference tools which occurs due to the extreme sparsity inherent to the lincRNA-disease association dataset.
- We present a comparison of our proposed RIMC method with standard IMC as well as the state-of-the-art lincRNA-disease association methods.

The rest of the paper is organized as follows. In section II we propose the robust IMC formulation using $\ell_{2,1}$ norm, underline the advantages of the proposed algorithm compared with the standard IMC as well as standard NMF approaches. Here we also show the correctness of the proposed algorithm. In section III we present the configurations for the experiments we conducted in this study. That includes description of the association dataset, side information dataset, feature extraction, summary of baseline algorithms as well as the performance metrics used to evaluate the models. In Section IV, we present the results of the association inference experiments on the dataset, and underline the superior performance of the proposed algorithm than the existing methods. Finally, in section V we conclude the paper by pointing out several future research scopes.

6.2 Robust Inductive Matrix Completion (RIMC)

In this section we review standard Inductive Matrix Completion method; then we present our robust IMC (RIMC) formulation. Later, we provide a computational algorithm for our proposed method along with the correctness of the algorithm. The whole idea of the Inductive Matrix Completion strategy can be summarized into a flow diagram as presented in Figure.

6.2.1 Review on Standard IMC

The Inductive Matrix Completion approach [104] enables us to incorporate side information of both the row and column entities. The formulation overcomes the limitation imposed by the transductive matrix completion approaches (e.g., standard NMF, etc.). Therefore we can predict association between new entities that are not included in the data matrix available at training time. Given input matrix $A \in \mathbb{R}^{M \times N}$ encapsulating the association between M row entities and N column entities. Besides A, side information of both the entities are given in two matrices $X \in \mathbb{R}^{M \times m}$

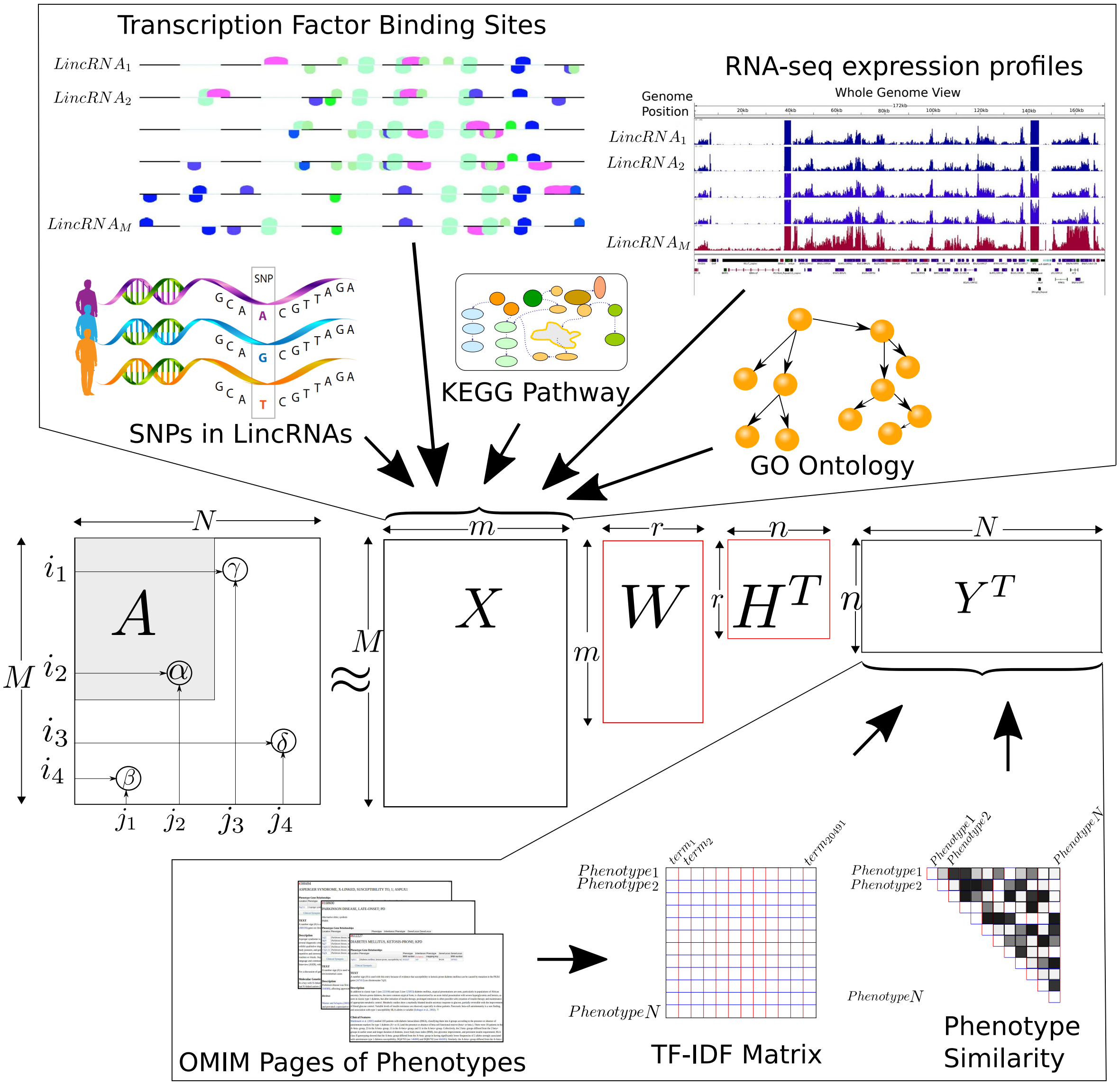


Figure 6.1: Inductive Matrix Completion Flow diagram: (From top-left) Given, A being the association matrix between two different sets (e.g., lincRNAs and disease phenotypes), each having M and N number of items, X being the feature matrix for the M items of the first set (e.g., lincRNAs), and Y being the feature set of the second set (e.g., disease phenotypes). Portion of the association matrix A is shaded denoting the there is atleast one association known between the lincRNA and the corresponding disease phenotype. Feature dimensions of the two sets are m and n respectively, and the feature sets were constructed from a collection of heterogeneous data repositories. The goal is to obtain a low-rank realization of the two feature set in terms of W and H matrices. Once W and H matrices are retrieved, through induction one can retrieve association scores between two items of the two sets, regardless the two items were considered in A, X, Y (as in the calculation of α score) or not (as in computing β, γ, δ scores). Calculation of α utilizes the existing A, X, Y matrices to retrieve corresponding feature vectors. However, for the calculations of β, γ, δ new feature vectors are to be constructed having similar feature dimensionality and then put into the calculations.

containing m features of the M row entities and $Y \in \mathbb{R}^{N\times n}$ containing n features of the N column entities respectively. The standard IMC is defined as,

$$\min_{W,H} \quad J = \frac{1}{2}||A - XWH^TY^T||_F^2 + \frac{\lambda_1}{2}||W||_F^2 + \frac{\lambda_2}{2}||H||_F^2, \\ \text{such that,} \quad W \geq 0, H \geq 0 \tag{6.1}$$

where $||B||_F^2 = \sum_{ij} B_{ij}^2$ is the Frobenius norm (i.e., ℓ_2 norm) of a matrix, and λ_1, λ_2 are the regularization parameters that trade off between the accrued loss on the observed entries and the trace norm regularization constraints. Here, goal is to recover a low-rank matrix $Z \in \mathbb{R}^{m\times n}$ using the observed entries of A, and the X and Y feature matrices. The entry A_{ij} is modeled as $\mathbf{x}_i^T Z \mathbf{y}_j$. By forming Z as WH^T, where $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{n\times r}$. The problem can be solved using Algorithm 8.

Once the W and H matrices are obtained, besides computing the associative scores among the row and column entities from the training set, it can also perform induction on a new row entity i' that was not part of the training data, the prediction $A_{i'j}$ can be computed for a column j as long as we have feature vector $\mathbf{x}_{i'}$, using the model as: $A_{i'j} = \mathbf{x}_{i'}WH^T\mathbf{y}_j$. Similarly, prediction can also be made for a new column entity j' with a row entity in the set using new feature vector $\mathbf{y}_{j'}$ by $A_{ij'} = \mathbf{x}_iWH^T\mathbf{y}_{j'}$. However, the prediction between a new column entity (j') and a new row entity (i') can also be computed through using their corresponding feature vectors, $\mathbf{x}_{i'}$ and $\mathbf{y}_{j'}$ through: $A_{i'j'} = \mathbf{x}_{i'}WH^T\mathbf{y}_{j'}$.

6.2.2 Robust IMC (RIMC) Formulation

One limitation of the standard IMC is that it is prone to outliers in the given dataset. Given $A \in \mathbb{R}^{M\times N}, X \in \mathbb{R}^{M\times m}, Y \in \mathbb{R}^{N\times n}$, the loss function of the standard IMC is:

$$||A - XWH^TY^T||_F^2 = \sum_{i=1}^{M} ||A_{i,:} - (XWH^TY^T)_{i,:}||_2^2 \tag{6.4}$$

Algorithm 8 COMPUTE_STANDARD_IMC(A,X,Y,r)

```
Input: association matrix $A \in \mathbb{R}^{M\times N}$; feature matrix for the $M$ row entities $X$;
    feature matrix for the $N$ column entities $Y$; desired rank $r$
Output: Calculate the two factor matrices $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{n\times r}$
Initialize $W$ and $H$ as random dense matrix maintaining the non-negativity constraints $W_{ik} \geq 0, H_{jk} \geq 0$.
repeat
    Update $H$ matrix using the following equation:
        $H_{jk} \leftarrow H_{jk} \frac{\left(Y^T A^T X W\right)_{jk}}{\left(Y^T Y H W^T X^T X W + \lambda_2 H\right)_{jk}} \quad (6.2)$
    Update $W$ matrix using the following equation. Here we will be using the $H$ calculated at the previous step.
        $W_{ik} \leftarrow W_{ik} \frac{\left(X^T A Y H\right)_{ik}}{\left(X^T X W H^T Y^T Y H + \lambda_1 W\right)_{ik}} \quad (6.3)$
until convergence criterion is met
return $W, H$
```

Here, the error for each row entity of the objective function, the squared residue error is accumulated in the form of $||A_{i,:} - (XWH^TY^T)_{i,:}||_2^2$. Hence, a few outliers with large error could dominate the overall computation. The second limitation of the standard IMC is that the ℓ_2 norm based regularization (i.e., ridge regularization) does not handle joint sparsity across the feature data matrices. By joint sparsity we refer to the set of features having either small scores across all data points, or large scores across all data points. Thus it is very important to present a robust IMC formulation.

The robust IMC formulation involves $\ell_{2,1}$ norm instead of ℓ_2 norm to define the loss function which is:

$$||A - XWH^TY^T||_{2,1} = \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{N}(A - XWH^TY^T)_{ij}^2} \tag{6.5}$$

Here, the error for each data point is not squared, and thus the large errors due to outliers do not dominate the objective function as they would in the standard IMC formulation.

We now propose robust IMC formulated as:

$$\begin{aligned} \min_{W,H} \quad J = ||A - XWH^TY^T||_{2,1} + \lambda_1 R(W) + \lambda_2 R(H), \\ \text{such that,} \quad W \geq 0, H \geq 0 \end{aligned} \tag{6.6}$$

Here, we have several options as the regularization function $R(\cdot)$; such as: $R_1(B) = ||B||_F^2$, $R_2(B) = \sum_{i=1}^{M} ||B_{i,:}||_1$, $R_3(B) = \sum_{i=1}^{M} ||B_{i,:}||_2^0$ and $R_4(B) = \sum_{i=1}^{M} ||B_{i,:}||_2$. Here, $R_1(\cdot)$ is the ridge regularization and is adapted in the standard IMC formulation, $R_2(\cdot)$ is the LASSO regularization which is a non-convex function and difficult to optimize. $R_3(\cdot)$ involves the ℓ_0 norm and is the most desirable [124], and $R_4(\cdot)$ employs the $\ell_{2,1}$ norm. We selected the $R_4(\cdot)$ because it is convex and can be easily optimized according to [125].

Thus given the data matrices A, X, Y, in this paper we optimize the following robust IMC formulation:

$$\begin{aligned} \min_{W,H} \quad J \quad &= \quad ||A - XWH^TY^T||_{2,1} + \lambda_1 ||W||_{2,1} + \lambda_2 ||H||_{2,1}, \\ &\text{such that,} \quad W \geq 0, H \geq 0 \end{aligned} \tag{6.7}$$

6.2.3 Algorithm for RIMC

The main contribution of this manuscript is to derive Algorithm 9 that solves the robust IMC optimization problem (Equation 6.7).

Algorithm 9 COMPUTE_ROBUST_IMC(A,X,Y,r)

```
Input: association matrix $A \in \mathbb{R}^{M \times N}$; feature matrix for the $M$ row entities $X$;
    feature matrix for the $N$ column entities $Y$; desired rank $r$
Output: Calculate the two factor matrices $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{n \times r}$
Initialize $W$ and $H$ as random dense matrix maintaining the non-negativity constraints $W_{ik} \geq 0, H_{jk} \geq 0$.
Initialize $D \in \mathbb{R}^{M \times M}, P \in \mathbb{R}^{m \times m}$, $Q \in \mathbb{R}^{n \times n}$ as identity matrices.
repeat
    Update $H$ matrix using the following equation:
    $H_{\gamma\psi} \leftarrow H_{\gamma\psi} \frac{\left(Y^T A^T D X W\right)_{\gamma\psi}}{\left(Y^T Y H W^T X^T D X W + \lambda_2 Q H\right)_{\gamma\psi}},$
    Update the diagonal matrix $D$ using the following equation:
    $D_{ii} = 1 \Big/ \sqrt{\sum_{j=1}^{N} (A - X W H^T Y^T)_{ij}^2}$    (6.8)
    Update the diagonal matrix $Q$ using the following equation:
    $Q_{ii} = 1 \Big/ \sqrt{\sum_{j=1}^{r} H_{ij}^2}$    (6.9)
    Update $W$ matrix using the following equation. Here we will be using the $H$ calculated at the previous step.
    $W_{\alpha\beta} \leftarrow W_{\alpha\beta} \frac{\left(X^T D A Y H\right)_{\alpha\beta}}{\left(X^T D X W H^T Y^T Y H + \lambda_1 P W\right)_{\alpha\beta}}$
    Update the diagonal matrix $P$ using the following equation –
    $P_{ii} = 1 \Big/ \sqrt{\sum_{j=1}^{r} W_{ij}^2}$    (6.10)
until convergence criterion is met
```

6.2.4 Convergence of the RIMC Algorithm

Here, we are going to prove the convergence of Algorithm 9 described in Theorem.

Theorem 6.2.1. *Algorithm 9 will monotonically decrease the objective function of the problem (Equation 6.7) in each iteration and converge to the global optimum of the problem.*

However, it can be rephrased using the following two statements:

(A) Updating H using equation 6.8 while fixing W, the objective function of the problem (Equation 6.7) monotonically decreases.

(B) Updating W using equation 6.10 while fixing H, the objective function of the problem (Equation 6.7) monotonically decreases.

Proof. We prove Theorem 6.2.1 (A,B) separately in the following two sections. □

6.2.5 Proof of Theorem 6.2.1(A): Updating of H

Proof. We now focus on proving Theorem 6.2.1(A). The proof requires the following two lemmas: (Lemma 6.2.2 and 6.2.3).

Lemma 6.2.2. *Let, $H^{(t)}$ be the H at the t^{th} iteration, and $H^{(t+1)}$ is obtained from the next iteration. Then, under the update rule of Equation 6.8, the following inequality holds.*

$$\begin{aligned}
tr\left((A - XWH^{(t+1)^T}Y^T)^T D(A - XWH^{(t+1)^T}Y^T)\right)& \\
+\lambda_1 tr\left(W^T PW\right) + \lambda_2 tr\left(H^{(t+1)^T} QH^{(t+1)}\right)& \\
\leq tr\left((A - XWH^{(t)^T}Y^T)^T D(A - XWH^{(t)^T}Y^T)\right)& \\
+\lambda_1 tr\left(W^T PW\right) + \lambda_2 tr\left(H^{(t)^T} QH^{(t)}\right)&, \qquad (6.11)
\end{aligned}$$

where, $D_{ii} = 1\Big/\sqrt{\sum_{j=1}^{N}(A - XWH^{(t)^T}Y^T)^2_{ij}}$, *and* $Q_{ii} = 1\Big/\sqrt{\sum_{j=1}^{r} H^{(t)^T}_{ij}}$

The proof of Lemma 6.2.2 is given in section 6.2.7.

Lemma 6.2.3. *Under the update rule of Equation 6.8, the following inequality holds:*

$$\begin{aligned}
&||A - XWH^{(t+1)^T}Y^T||_{2,1} + \lambda_1||W||_{2,1} + \lambda_2||H^{(t+1)}||_{2,1}\\
&-||A - XWH^{(t)^T}Y^T||_{2,1} - \lambda_1||W||_{2,1} - \lambda_2||H^{(t)}||_{2,1} \leq\\
&\frac{1}{2}\left\{tr\left((A - XWH^{(t+1)^T}Y^T)^T D(A - XWH^{(t+1)^T}Y^T)\right)\right.\\
&+\lambda_1 tr\left(W^T PW\right) + \lambda_2 tr\left(H^{(t+1)^T}QH^{(t+1)}\right)\\
&-tr\left((A - XWH^{(t)^T}Y^T)^T D(A - XWH^{(t)^T}Y^T)\right)\\
&\left.-\lambda_1 tr\left(W^T PW\right) - \lambda_2 tr\left(H^{(t)^T}QH^{(t)}\right)\right\}, \qquad (6.12)
\end{aligned}$$

where D, P, Q matrices are defined earlier.

The proof of Lemma 6.2.3 is given in section 6.2.8.

Now, if we take a look at the right hand side of the inequality in Equation 6.12, the value is negative or zero according to Lemma 6.2.2. This completes the proof that the objective function of Equation 6.7 decreases monotonically. □

6.2.6 Proof of Theorem 6.2.1(B): Updating of W

Proof. We now focus on proving Theorem 6.2.1(B). The proof requires the following two lemmas: (Lemma 6.2.4 and 6.2.5).

Lemma 6.2.4. *Let, $W^{(t)}$ be the W at the t^{th} iteration, and $W^{(t+1)}$ is obtained from the next iteration. Then, under the update rule of Equation 6.10, the following inequality holds.*

$$\begin{aligned} tr\left((A - XW^{(t+1)}H^TY^T)^TD(A - XW^{(t+1)}H^TY^T)\right) \\ +\lambda_1 tr\left(W^{(t+1)^T}PW^{(t+1)}\right) + \lambda_2 tr\left(H^TQH\right) \\ \leq tr\left((A - XW^{(t)}H^TY^T)^TD(A - XW^{(t)}H^TY^T)\right) \\ +\lambda_1 tr\left(W^{(t)^T}PW^{(t)}\right) + \lambda_2 tr\left(H^TQH\right), \end{aligned} \tag{6.13}$$

where, D, P, Q are defined earlier.

Proof of Lemma 6.2.4 is provided in section 6.2.9.

Lemma 6.2.5. *Under the update rule of Equation 6.10, the following inequality holds:*

$$\begin{aligned} &||A - XW^{(t+1)}H^TY^T||_{2,1} + \lambda_1||W^{(t+1)}||_{2,1} + \lambda_2||H||_{2,1} \\ &-||A - XW^{(t)}H^TY^T||_{2,1} - \lambda_1||W^{(t)}||_{2,1} - \lambda_2||H||_{2,1} \leq \\ &\frac{1}{2}\left\{tr\left((A - XW^{(t+1)}H^TY^T)^TD(A - XW^{(t+1)}H^TY^T)\right)\right. \\ &+\lambda_1 tr\left(W^{(t+1)^T}PW^{(t+1)}\right) + \lambda_2 tr\left(H^TQH\right) \\ &-tr\left((A - XW^{(t)}H^TY^T)^TD(A - XW^{(t)}H^TY^T)\right) \\ &\left.-\lambda_1 tr\left(W^{(t)^T}PW^{(t)}\right) - \lambda_2 tr\left(H^TQH\right)\right\}, \end{aligned} \tag{6.14}$$

where D, P, Q matrices are defined earlier.

Proof of Lemma 6.2.5 is provided in section 6.2.10.

Now, if we take a look at the right hand side of the inequality in Equation 6.14, the value is negative or zero according to Lemma 6.2.4. This completes the proof that the objective function of Equation 6.7 decreases monotonically. □

6.2.7 Proof of Lemma 6.2.2

Proof. We can re-write equation 6.11 as follows:

$$J(H^{(t+1)}) \leq J(H^{(t)}), \tag{6.15}$$

where

$$\begin{aligned} J(H) &= tr(A - XWH^TY^T)^T D(A - XWH^TY^T) \\ &\quad + \lambda_1 tr(W^TPW) + \lambda_2 tr(H^TQH) \end{aligned} \tag{6.16}$$

And, according to the statement of Lemma 6.2.2, under the update rule of equation 6.8, $J(H)$ monotonically decreases. In order to prove the statement, we follow the approaches utilizing auxiliary functions [79, 126].

Definition 6.2.1. $G(H, H')$ is an auxiliary function for the function $J(H)$ if $G(H, H') \geq J(H)$ for all H' and $G(H, H) = J(H)$.

Now, we define:

$$H^{(t+1)} = \underset{H}{\operatorname{argmin}} \quad G(H, H^{(t)})$$

So, we have

$$\begin{aligned} J(H^{(t+1)}) &= G(H^{(t+1)}, H^{(t+1)}) \leq G(H^{(t+1)}, H^{(t)}) \\ &\leq G(H^{(t)}, H^{(t)}) = J(H^{(t)}) \end{aligned}$$

This proves that $J(H^{(t)})$ is monotonically decreasing.

Now the important steps in the remainder of the proof are: (a) determine a proper auxiliary function, and (b) find the global minima of the auxiliary function.

Lemma 6.2.6. *The function*

$$G(H, H') = tr(A^TDA) - 2tr(YHW^TX^TDA)$$

$$+\lambda_1 tr(W^T PW) + \lambda_2 tr(H^T QH)$$

$$+\sum_{i=1}^{n}\sum_{j=1}^{r}\frac{(Y^T YH'W^T X^T DXW)_{ij}H_{ij}^2}{H'_{ij}} \tag{6.17}$$

is an auxiliary function for J.

Proof. Now $J(H)$ of equation 6.16 can be re-written as:

$$J(H) = tr(A^T DA) - 2tr(YHW^T X^T DA)$$

$$+\lambda_1 tr(W^T PW) + \lambda_2 tr(H^T QH)$$

$$+tr(H^T Y^T YHW^T X^T DXW) \tag{6.18}$$

Now we will be applying the following inequality of matrices according to the investigations by [127, 126]:

$$tr(H^T \Lambda HB) \leq \sum_i \sum_j (\Lambda H'B)_{ji}\frac{H_{ij}^2}{H'_{ij}}, \tag{6.19}$$

where, Λ, B, H are non-negative matrices, and Λ, B are symmetric matrices. And obviously the equality holds in Equation 6.19 when $H = H'$.

In equation 6.19, if we do the substitutions: $\Lambda = Y^T Y, B = W^T X^T DXW, H = H, H' = H'$, we see that the fifth term of equation 6.18 is smaller than the fifth term of equation 6.17. However, the equality holds when $H = H'$. Thus $G(H, H')$ in equation 6.17 is an auxiliary function of $J(H)$. □

Now, we need to find the global minimum of Equation 6.17. Let $f(H) = G(H, H')$. The gradient of $f(H)$ is

$$\frac{\partial f(H)}{\partial H_{ij}} = -2(Y^T A^T DXW)_{ij} + 2\lambda_2 (QH)_{ij}$$

$$+2\frac{(Y^T YH'W^T X^T DXW)_{ij}H_{ij}}{H'_{ij}} \tag{6.20}$$

However, the second order derivative (i.e., the Hessian matrix) would be

$$\frac{\partial^2 f(H)}{\partial H_{ij}\partial H_{kl}} = 2\,(Q)_{jl}\,\delta_{ik}\delta_{k\beta} + \left(2\frac{(Y^T DYH'W^T X^T XW)_{ij}}{H'_{ij}}\right)\delta_{jl}\delta_{ik} \tag{6.21}$$

The Hessian matrix (Equation 6.21 is semi-positive definite implying that $f(H) = G(H, H')$ is a convex function. Thus, there exists a unique global minimum for $f(H)$. The global minimum can be obtained by setting the gradient of $f(H)$ to zero and solve for H. Thus from equation 6.20 we get

$$H_{ij} = H'_{ij}\frac{(Y^T A^T DXW)_{ij}}{(Y^T YH'W^T X^T DXW + \lambda_2 QH)_{ij}} \tag{6.22}$$

By replacing $H^{(t+1)} = H$ and $H^{(t)} = H'$, we would obtain the update rule of Equation 6.8. Therefore, under this rule, the objective function $J(H)$ of Equation 6.16 decreases monotonically, and hence completes the proof. □

6.2.8 Proof of Lemma 6.2.3

Proof. We know that,

$$tr(A - XWH^{(t)^T}Y^T)^T D(A - XWH^{(t)^T}Y^T)$$
$$+ \lambda_1 tr(W^T PW) + \lambda_2 tr(H^{(t)^T} QH^{(t)})$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{N}(A - XWH^{(t)^T}Y^T)_{ij} D_{ii}$$
$$+ \lambda_1 tr(W^T PW) + \lambda_2 \sum_{k=1}^{n}\sum_{l=1}^{r} H_{kl}^{(t)^2} Q_{kk}$$
$$= \sum_{i=1}^{M} ||A_i - (XWH^{(t)^T}Y^T)_i||^2 D_{ii}$$
$$+ \lambda_1 tr(W^T PW) + \lambda_2 \sum_{k=1}^{n} ||H_k^{(t)}||^2 Q_{kk}$$

Similarly, we can see that

$$tr(A - XWH^{(t+1)^T}Y^T)^T D(A - XWH^{(t+1)^T}Y^T)$$

$$+ \lambda_1 tr(W^T PW) + \lambda_2 tr(H^{(t+1)^T} QH^{(t+1)})$$

$$= \sum_{i=1}^{M} ||A_i - (XWH^{(t+1)^T} Y^T)_i||^2 D_{ii}$$

$$+ \lambda_1 tr(W^T PW) + \lambda_2 \sum_{k=1}^{n} ||H_k^{(t+1)}||^2 Q_{kk}$$

Then, the right-hand side ($r.h.s$) of Equation 6.12 becomes

$$r.h.s = \frac{1}{2} \sum_{i=1}^{M} \Big(||A_i - (XWH^{(t+1)^T} Y^T)_i||^2$$

$$-||A_i - (XWH^{(t)^T} Y^T)_i|| \Big) D_{ii} + \lambda_2 \sum_{k=1}^{n} \Big(||H_k^{(t+1)}||^2$$

$$-||H_k^{(t)}||^2 \Big) Q_{kk}$$

$$= \frac{1}{2} \sum_{i=1}^{M} \left(||A_i - (XWH^{(t+1)^T} Y^T)_i||^2 D_{ii} - \frac{1}{D_{ii}} \right)$$

$$+ \lambda_2 \sum_{k=1}^{n} \left(||H_k^{(t+1)}||^2 Q_{kk} - \frac{1}{Q_{kk}} \right)$$

And, the left-hand side ($l.h.s$) of Equation 6.12 becomes

$$l.h.s = \sum_{i=1}^{M} \Big(\sqrt{||A_i - (XWH^{(t+1)^T} Y^T)_i||^2}$$

$$-\sqrt{||A_i - (XWH^{(t)^T} Y^T)_i||^2} \Big)$$

$$+ \lambda_2 \sum_{k=1}^{n} \left(\sqrt{||H_k^{(t+1)}||^2} - \sqrt{||H_k^{(t)}||^2} \right)$$

$$= \sum_{i=1}^{M} \Big(||A_i - (XWH^{(t+1)^T} Y^T)_i||$$

$$-||A_i - (XWH^{(t)^T} Y^T)_i|| \Big)$$

$$+ \lambda_2 \sum_{k=1}^{n} \left(\sqrt{||H_k^{(t+1)}||^2} - \sqrt{||H_k^{(t)}||^2} \right)$$

$$= \sum_{i=1}^{M} \left(||A_i - (XWH^{(t+1)^T} Y^T)_i|| - \frac{1}{D_{ii}} \right)$$

$$+ \lambda_2 \left(\sum_{k=1}^{n} ||H_k^{(t+1)}|| - \frac{1}{Q_{kk}} \right)$$

Now, we compute the difference between the $l.h.s$ and $r.h.s$,

$$l.h.s - r.h.s = \sum_{i=1}^{M} \Big(||A_i - (XWH^{(t+1)^T}Y^T)_i||$$

$$-||A_i - (XWH^{(t+1)^T}Y^T)_i||^2 \frac{D_{ii}}{2} - \frac{1}{2D_{ii}} \Big)$$

$$+ \lambda_2 \sum_{k=1}^{n} \left(||H_k^{(t+1)}|| - ||H_k^{(t+1)}||^2 \frac{Q_{kk}}{2} - \frac{1}{2Q_{kk}} \right)$$

$$= \sum_{i=1}^{M} \frac{D_{ii}}{2} \Big(\frac{||A_i - (XWH^{(t+1)^T}Y^T)_i||}{D_{ii}}$$

$$-||A_i - (XWH^{(t+1)^T}Y^T)_i||^2 - \frac{1}{D_{ii}^2} \Big)$$

$$+ \lambda_2 \sum_{k=1}^{n} \frac{Q_{kk}}{2} \left(\frac{||H_k^{(t+1)}||}{Q_{kk}} - ||H_k^{(t+1)}||^2 - \frac{1}{Q_{kk}^2} \right)$$

$$= \sum_{i=1}^{M} \frac{(-D_{ii})}{2} \left(||A_i - (XWH^{(t+1)^T}Y^T)_i|| - \frac{1}{D_{ii}} \right)^2$$

$$+ \lambda_2 \sum_{k=1}^{n} \frac{(-Q_{kk})}{2} \left(||H_k^{(t+1)}|| - \frac{1}{Q_{kk}} \right)^2$$

$$\leq 0$$

The above inequality holds because, D, Q are non-negative matrices, and the sum of non-positive numbers is always non-positive. This completes the proof. □

6.2.9 Proof of Lemma 6.2.4

Proof. We can re-write equation 6.13 as follows:

$$J(W^{(t+1)}) \leq J(W^{(t)}), \tag{6.23}$$

where

$$\begin{aligned} J(W) &= tr(A - XWH^TY^T)^T D(A - XWH^TY^T) \\ &\quad +\lambda_1 tr(W^TPW) + \lambda_2 tr(H^TQH) \end{aligned} \tag{6.24}$$

And, according to the statement of Lemma 6.2.4, under the update rule of equation 6.10, $J(W)$ monotonically decreases. In order to prove the statement, we follow the approaches utilizing auxiliary functions [79, 126].

Definition 6.2.2. $G(W, W')$ is an auxiliary function for the function $J(W)$ if $G(W, W') \geq J(W)$ for all W' and $G(W, W) = J(W)$.

Now, we define:

$$W^{(t+1)} = \underset{W}{\operatorname{argmin}} \quad G(W, W^{(t)})$$

So, we have

$$\begin{aligned} J(W^{(t+1)}) &= G(W^{(t+1)}, W^{(t+1)}) \leq G(W^{(t+1)}, W^{(t)}) \\ &\leq G(W^{(t)}, W^{(t)}) = J(W^{(t)}) \end{aligned}$$

This proves that $J(W^{(t)})$ is monotonically decreasing.

Now the important steps in the remainder of the proof are: (a) determine a proper auxiliary function, and (b) find the global minima of the auxiliary function.

Lemma 6.2.7. *The function*

$$G(W, W') = tr(A^T DA) - 2tr(YHW^T X^T DA)$$

$$\begin{aligned} &+\lambda_1 tr(W^T PW) + \lambda_2 tr(H^T QH) \\ &+\sum_{i=1}^{m}\sum_{j=1}^{r} \frac{(X^T DXW'H^T Y^T YH)_{ij} W_{ij}^2}{W'_{ij}} \end{aligned} \tag{6.25}$$

is an auxiliary function for J.

Proof. Now $J(W)$ of equation 6.32 can be re-written as:

$$J(W) = tr(A^T DA) - 2tr(YHW^T X^T DA)$$

$$+\lambda_1 tr(W^T PW) + \lambda_2 tr(H^T QH)$$
$$+tr(W^T X^T DXWH^T Y^T YH) \tag{6.26}$$

Now we will be applying the following inequality of matrices according to the investigations by [127, 126]:

$$tr(W^T \Lambda WB) \leq \sum_i \sum_j (\Lambda W' B)_{ji} \frac{W_{ij}^2}{W'_{ij}}, \tag{6.27}$$

where, Λ, B, W are non-negative matrices, and Λ, B are symmetric matrices. And obviously the equality holds in Equation 6.27 when $W = W'$.

In equation 6.27, if we do the substitutions: $\Lambda = X^T DX, B = H^T Y^T YH, W = W, W' = W'$, we see that the fifth term of equation 6.26 is smaller than the fifth term of equation 6.25. However, the equality holds when $W = W'$. Thus $G(W, W')$ in equation 6.25 is an auxiliary function of $J(W)$. □

Now, we need to find the global minimum of Equation 6.25. Let $f(W) = G(W, W')$. The gradient of $f(W)$ is

$$\frac{\partial f(W)}{\partial W_{ij}} = -2(X^T DAYH)_{ij} + 2\lambda_1 (PW)_{ij}$$

$$+2\frac{(X^T DXW'H^T Y^T YH)_{ij} W_{ij}}{W'_{ij}} \tag{6.28}$$

However, the second order derivative (i.e., the Hessian matrix) would be

$$\frac{\partial^2 f(W)}{\partial W_{ij} \partial W_{kl}} = 2\,(P)_{ij}\, \delta_{jl}\delta_{ik}$$
$$+\left(2\frac{(X^T DXW'H^T Y^T YH)_{ij}}{W'_{ij}}\right)\delta_{jl}\delta_{ik} \tag{6.29}$$

The Hessian matrix (Equation 6.29) is semi-positive definite implying that $f(W) = G(W, W')$ is a convex function. Thus, there exists a unique global minimum for $f(W)$.

The global minimum can be obtained by setting the gradient of $f(W)$ to zero and solve for W. Thus from equation 6.28 we get

$$W_{ij} = W'_{ij} \frac{(X^T DAYH)_{ij}}{(X^T DXW'H^T Y^T YH + \lambda_1 PW)_{ij}} \tag{6.30}$$

By replacing $W^{(t+1)} = W$ and $W^{(t)} = W'$, we would obtain the update rule of Equation 6.10. Therefore, under this rule, the objective function $J(W)$ of Equation 6.32 decreases monotonically, and hence completes the proof. □

6.2.10 Proof of Lemma 6.2.5

Proof. We know that,

$$
\begin{aligned}
& tr(A - XW^{(t)}H^TY^T)^T D(A - XW^{(t)}H^TY^T) \\
& + \lambda_1 tr({W^{(t)}}^T PW^{(t)}) + \lambda_2 tr(H^T QH) \\
& = \sum_{i=1}^{M} \sum_{j=1}^{N} (A - XW^{(t)}H^TY^T)_{ij} D_{ii} \\
& + \lambda_1 \sum_{k=1}^{m} \sum_{l=1}^{r} {W_{kl}^{(t)}}^2 P_{kk} + \lambda_2 tr(H^T QH) \\
& = \sum_{i=1}^{M} ||A_i - (XW^{(t)}H^TY^T)_i||^2 D_{ii} \\
& + \lambda_1 \sum_{k=1}^{m} ||W_k^{(t)}||^2 P_{kk} + \lambda_2 tr(H^T QH)
\end{aligned}
$$

Similarly, we can see that

$$
\begin{aligned}
& tr(A - XW^{(t+1)}H^TY^T)^T D(A - XW^{(t+1)}H^TY^T) \\
& + \lambda_1 tr({W^{(t+1)}}^T PW^{(t+1)}) + \lambda_2 tr(H^T QH) \\
& = \sum_{i=1}^{M} ||A_i - (XW^{(t+1)}H^TY^T)_i||^2 D_{ii} \\
& + \lambda_1 \sum_{k=1}^{m} ||W_k^{(t+1)}||^2 P_{kk} + \lambda_2 tr(H^T QH)
\end{aligned}
$$

Then, the right-hand side ($r.h.s$) of Equation 6.14 becomes

$$r.h.s = \frac{1}{2}\sum_{i=1}^{M}\left(||A_i - (XW^{(t+1)}H^TY^T)_i||^2\right.$$
$$\left.-||A_i - (XW^{(t)}H^TY^T)_i||\right) D_{ii} + \lambda_1 \sum_{k=1}^{m}\left(||W_k^{(t+1)}||^2\right.$$
$$\left.-||W_k^{(t)}||^2\right) P_{kk}$$
$$= \frac{1}{2}\sum_{i=1}^{M}\left(||A_i - (XW^{(t+1)}H^TY^T)_i||^2 D_{ii} - \frac{1}{D_{ii}}\right)$$
$$+ \lambda_1 \sum_{k=1}^{m}\left(||W_k^{(t+1)}||^2 P_{kk} - \frac{1}{P_{kk}}\right)$$

And, the left-hand side ($l.h.s$) of Equation 6.14 becomes

$$l.h.s = \sum_{i=1}^{M}\left(\sqrt{||A_i - (XW^{(t+1)}H^TY^T)_i||^2}\right.$$
$$\left.-\sqrt{||A_i - (XW^{(t)}H^TY^T)_i||^2}\right)$$
$$+ \lambda_1 \sum_{k=1}^{m}\left(\sqrt{||W_k^{(t+1)}||^2} - \sqrt{||W_k^{(t)}||^2}\right)$$
$$= \sum_{i=1}^{M}\left(||A_i - (XW^{(t+1)}H^TY^T)_i||\right.$$
$$\left.-||A_i - (XW^{(t)}H^TY^T)_i||\right)$$
$$+ \lambda_1 \sum_{k=1}^{m}\left(\sqrt{||W_k^{(t+1)}||^2} - \sqrt{||W_k^{(t)}||^2}\right)$$
$$= \sum_{i=1}^{M}\left(||A_i - (XW^{(t+1)}H^TY^T)_i|| - \frac{1}{D_{ii}}\right)$$
$$+ \lambda_1 \left(\sum_{k=1}^{m} ||W_k^{(t+1)}|| - \frac{1}{P_{kk}}\right)$$

Now, we compute the difference between the $l.h.s$ and $r.h.s$,

$$l.h.s - r.h.s = \sum_{i=1}^{M}\left(||A_i - (XW^{(t+1)}H^TY^T)_i||\right.$$
$$\left.-||A_i - (XW^{(t+1)}H^TY^T)_i||^2 \frac{D_{ii}}{2} - \frac{1}{2D_{ii}}\right)$$

$$+\lambda_1 \sum_{k=1}^{m} \left(||W_k^{(t+1)}|| - ||W_k^{(t+1)}||^2 \frac{P_{kk}}{2} - \frac{1}{2P_{kk}} \right)$$

$$= \sum_{i=1}^{M} \frac{D_{ii}}{2} \left(\frac{||A_i - (XW^{(t+1)}H^TY^T)_i||}{D_{ii}} \right.$$

$$\left. - ||A_i - (XW^{(t+1)}H^TY^T)_i||^2 - \frac{1}{D_{ii}^2} \right)$$

$$+\lambda_1 \sum_{k=1}^{m} \frac{P_{kk}}{2} \left(\frac{||W_k^{(t+1)}||}{P_{kk}} - ||W_k^{(t+1)}||^2 - \frac{1}{P_{kk}^2} \right)$$

$$= \sum_{i=1}^{M} \frac{(-D_{ii})}{2} \left(||A_i - (XW^{(t+1)}H^TY^T)_i|| - \frac{1}{D_{ii}} \right)^2$$

$$+\lambda_1 \sum_{k=1}^{m} \frac{(-P_{kk})}{2} \left(||W_k^{(t+1)}|| - \frac{1}{P_{kk}} \right)^2$$

$$\leq 0$$

The above inequality holds because, D, P are non-negative matrices, and the sum of non-positive numbers is always non-positive. This completes the proof. □

6.2.11 Correctness of the RIMC Algorithm

In this section we are going to prove that the converged solution presented in Algorithm 9 is the correct optimal solution. In fact, we will show that the converged solution satisfies the Karush-Kuhn-Tucker (KKT) condition of the constrained optimization theory. At first, we have theorem 6.2.8 to prove the correctness of the algorithm with respect to W. Theorem 6.2.9 will prove the correctness of the algorithm with respect to H.

Theorem 6.2.8. *At convergence, the converged solution W^* of the updating rule in Algorithm 9 satisfies the KKT condition.*

Proof. The KKT condition for W with constraints $W_{\alpha\beta} \geq 0$, with $\alpha = 1, \cdots, m; \beta = 1, \cdots, r$ is:

$$\frac{\partial J(W)}{\partial W_{\alpha\beta}} W_{\alpha\beta} = 0, \forall \alpha, \beta \tag{6.31}$$

Similar to Equation 6.16, the $J(W)$ can be written as:

$$\begin{aligned} J(W) &= tr(A - XWH^TY^T)^T D(A - XWH^TY^T) \\ &\quad + \lambda_1 tr(W^TPW) + \lambda_2 tr(H^TQH) \end{aligned} \tag{6.32}$$

Now, the partial derivative of $J(W)$ can be expressed as:

$$\begin{aligned} &\frac{\partial J(W)}{\partial W_{\alpha\beta}} = -2(X^TDAYH)_{\alpha\beta} + 2\lambda_1 \left(PW\right)_{\alpha\beta} \\ &+ 2\left(X^TDXWH^TY^TYH\right)_{\alpha\beta} \end{aligned} \tag{6.33}$$

Thus, the KKT condition for W is:

$$\begin{aligned} &\left[-\left(X^TDAYH\right)_{\alpha\beta} + \lambda_1(PW)_{\alpha\beta} \right. \\ &\left. + \left(X^TDXWH^TY^TYH\right)_{\alpha\beta}\right] W_{\alpha\beta} = 0, \forall \alpha, \beta \end{aligned} \tag{6.34}$$

But, once W converges (according to Algorithm 9), the converged solution W^* satisfies the following:

$$W^*_{\alpha\beta} \leftarrow W^*_{\alpha\beta} \frac{\left(X^TDAYH\right)_{\alpha\beta}}{(X^TDXW^*H^TY^TYH + \lambda_1 PW^*)_{\alpha\beta}}$$

which can be written as

$$\begin{aligned} &\left[-\left(X^TDAYH\right)_{\alpha\beta} + \lambda_1(PW^*)_{\alpha\beta} \right. \\ &\left. + \left(X^TDXW^*H^TY^TYH\right)_{\alpha\beta}\right] W^*_{\alpha\beta} = 0, \forall \alpha, \beta \end{aligned} \tag{6.35}$$

This is identical to equation 6.34. Thus, the converged solution W^* satisfies the KKT condition. □

Theorem 6.2.9. *At convergence, the converged solution H^* of the updating rule in Algorithm 9 satisfies the KKT condition.*

Proof. The KKT condition for H with constraints $H_{\gamma\psi} \geq 0$, with $\gamma = 1, \cdots, n, \psi = 1, \cdots, r$ is:

$$\frac{\partial J(H)}{\partial H_{\gamma\psi}} H_{\gamma\psi} = 0, \forall \gamma, \psi \tag{6.36}$$

Now, the partial derivative of $J(H)$ from equation 6.16 is

$$\frac{\partial J(H)}{\partial H_{\gamma\psi}} = -2\left(Y^T A^T DXW\right)_{\gamma\psi} + 2\lambda_2 (QH)_{\gamma\psi} \\ + 2\left(Y^T Y H W^T X^T DXW\right)_{\gamma\psi} \tag{6.37}$$

Thus, the KKT condition for H is:

$$\Big[-\left(Y^T A^T DXW\right)_{\gamma\psi} + \lambda_2 (QH)_{\gamma\psi} \\ + \left(Y^T Y H W^T X^T DXW\right)_{\gamma\psi}\Big] H_{\gamma\psi} = 0, \forall \gamma, \psi \tag{6.38}$$

But, once H converges (according to Algorithm 9), the converged solution, H^* satisfies the following:

$$H^*_{\gamma\psi} \leftarrow H^*_{\gamma\psi} \frac{\left(Y^T A^T DXW\right)_{\gamma\psi}}{(Y^T Y H^* W^T X^T DXW + \lambda_2 Q H^*)_{\gamma\psi}}$$

which can be written as

$$\Big[-\left(Y^T A^T DXW\right)_{\gamma\psi} + \lambda_2 (QH^*)_{\gamma\psi} \\ + \left(Y^T Y H^* W^T X^T DXW\right)_{\gamma\psi}\Big] H^*_{\gamma\psi} = 0, \forall \gamma, \psi$$

This is identical to equation 6.38. Thus, the converged solution H^* satisfies the KKT condition. □

6.2.12 Computational Complexity of the RIMC Algorithm

Computational complexity is another issue as we hope that RIMC algorithm (Algorithm 9 is not more time consuming. Hence, we analyze the computational cost of the algorithm. Since, $r < min(m, n)$, the numerator of Equation 6.8 can be computed as $(Y^T(A^T(D(XW))))$ with $r(nN + NM + M^2 + Mm)$ scalar multiplications; whereas one can compute the denominator of the equation as $(Y^T(Y(H(W^T(X^T(D(XW)))))))+ \lambda_2(QH)$ with $r(2nN + 2mM + nr + mr + nm)$ scalar multiplications. For computing the numerator of Equation 6.10, one can use the parenthesization $(X^T(D(A(YH))))$ with $r(mM + M^2 + MN + Nn)$ scalar multiplications, and the denominator as $(X^T(D(X(W(H^T(Y^T(YH))))))) + \lambda_1(PW)$ with $r(2nN + 2mM + nr + mr + m^2)$ scalar multiplications. Thus, for the complexity of the Algorithm 9 is $\zeta r(6nN + 2MN + 2M^2 + 6mM + 2nr + 2mr + mn + m^2)$, where ζ is total number of iterations until the convergence (or the stopping) criterion is met.

6.3 Experimental Configurations

6.3.1 Disease-LincRNA Association dataset

We obtained human lincRNA-disease associations by combining the LncRNADisease database [67] and the supporting dataset from the co-expression based association study conducted by [103]. The combined dataset contains 46,934 associations among 8194 lincRNA genes and 1213 diseases. Since none of the two datasets adapted standard naming of the diseases, we retrieved top-5 closely matched OMIM phenotypes for each of the disease names from the pool using OMIM API [76], and prepared the association matrix between 8194 lincRNA genes and 2661 OMIM phenotypes. The matrix is very sparse having only 0.22% non-zero entries. To compare different approaches on the novel association prediction, we use 10-fold cross validation over the association dataset.

6.3.2 LincRNA Feature datasets

RNA-seq provided Expression profiles of lincRNAs on different tissues underline the impact of the lincRNAs for diseases occurring corresponding tissues. Although not all diseases are tissue-specific, neither are the lincRNAs, the profiles still can be used to distinguish between co-expressed lincRNAs to implicate diseases. RNA-seq measurement of 8194 lincRNA expression levels on 22 human tissues are obtained from the Human BodyMap Project 2.0 [72]. Expression scores are represented in terms of FPKM values (Fragments Per Kilobase of exons per Million Fragments mapped).

ChIP-seq provided Transcription Factor Binding Sites (TFBS) of the lincRNAs unravel the transcriptional regulatory relationships of lincRNAs with transcription factors. We obtained 160,588 relationships among the 8194 lincRNAs and 120 transcription factors from ChIP-Base dataset [112]. There are only 217 lincRNAs that have relationship with one transcription factor, and out of 120 transcription factors. The minimally related transcript factor, “BACH1” has 11 lincRNA connections, and there are 6130 lincRNAs connecting with a transcript factor, “HNF4A”.

Functional annotations of the lincRNAs dictate their characterizations and involvement on various biological activities inside human cells that implicitly correlate with various disease phenotypes. Linc2GO [113] presents a database of such annotations of lincRNAs based on the ceRNA hypothesis [114]. We retrieved 8111 GO BP (Biological Process) terms, 3218 GO MF (Molecular Function) terms and 193 KEGG pathway terms associated with the 8194 lincRNAs from the database, resulting a total of 11522 functional terms for each lincRNA in our study. However, this annotation matrix is also sparse, having 0.11% non-zero entries. We use the leading 100 singular vectors of the matrix as the representative features of the lincRNAs contributed from the Linc2GO dataset.

Single Nucleotide Polymorphisms (SNPs) in lincRNAs were found to be linked to their abnormal expressions and dysregulations, thereby playing key roles in various phenotypes and diseases [115]. The lncRNASNP dataset [5] provides a comprehensive resource of SNPs in human lncRNAs, and we extracted 368,494 SNPs in the 8194 lincRNAs from the database. The SNP-lincRNA association is sparse with 0.0077% non-zero entries. We use the leading 100 singular vectors of the matrix as the representative features of the lincRNAs contributed from the lncRNASNP dataset.

Finally, we considered only those lincRNAs having all these four types of features. Therefore, we ended up having a catalog of 6540 lincRNAs with corresponding features.

6.3.3 Disease Feature datasets

Term Frequency Inverse Document Frequency (TF-IDF) of the 2661 OMIM phenotypes obtained from the OMIM text corpus provides a standard statistic that reflects how important a term is to a OMIM phenotype text collection. The TF-IDF score increases proportionally to the frequency of occurrences of a term in a particular page, but is offset by the frequency of the term in the whole corpus. This phenomenon helps to identify important keywords associated appearing only in the corresponding OMIM page, as well as less important terms appearing most of the pages. The number of terms considered in the scheme is 20491, thus resulting in a TF-IDF matrix of size 2661 by 20491. We use the leading 100 singular vectors of the matrix as the representative features of the diseases contributed from the OMIM TF-IDF dataset.

Phenotypic similarity profiles of the lincRNAs were retrieved from a recent study by [116], where the authors developed a method to accumulate the MeSH terms associated with the publications referenced in the OMIM phenotype pages and able to compute scores that reflect the molecular relatedness between two OMIM entries. The similarity matrix thereby is symmetric of dimension 2661 by 2661. We reduce the dimensionality of the feature space using PCA, retaining the top 100 principal components.

We considered only those diseases having all these two types of features. Thus, we ended up having a catalog of 2148 diseases with corresponding features.

6.3.4 Baselines

We compare the results of our proposed method with five approaches. Firstly, the standard non-negative matrix factorization on the lincRNA-disease association matrix A. This can be considered as a special case for the Inductive Matrix Completion objective where the lincRNA feature matrix (X) and the disease feature matrix (Y) are set to identity. We compare three other methods that provide interfaces to scale their corresponding framework to a much larger dataset like ours (i.e., considering associations among the 6540 lincRNAs and 2418 disease phenotypes). The approaches differ much in solving the association problems, from significant test for associating identities, solving the Graph Laplacian Regularized Least Squares, to Kernalized Random Walk with restart approach. We describe each of the four methods in more detail below. Finally, we compared the standard IMC method as discussed in Section 6.2.1.

6.3.4.1 NMF [79]

Here, we consider matrix completion on the bipartite network A and solve the standard optimization equation enforcing the non-negativity constraints over W and H using Alternating Least Squares (ALS) method. The standard matrix completion formulation does not accommodate any of the side information available about the lincRNAs and diseases. After the convergence of NMF Algorithm rank of the predictions can be retrieved using the estimated values of the matrix, i.e., higher the estimated A_{ij}, more relevant is the lincRNA i for disease j.

6.3.4.2 LRLSLDA [68]

The Laplacian Regularized Least Squares for LncRNA-Disease Association (LRLSLDA) computes a weighted rank score for association between an lincRNA with a disease using probabilities retrieved from two independent classifiers modeled using lincRNA-lincRNA and disease-disease similarity matrices. The computationally expensive operation in LRLSLDA is during the pairwise similarity matrix constructions which prohibits its usability in scalable framework development. Moreover, there are eight parameters used in LRLSLDA, which comparatively is a large number to tune in order to make the method computationally efficient.

6.3.4.3 TsLincRNA-Disease [103]

Tissue - Specificity based LincRNA - Disease association prediction framework (TsLincRNA - Disease) draws a demarcation line between tissue-specific and non-tissue-specific classes utilizing the tissue-specificity index for each of the lincRNAs in the study. It uses statistical significance test and a mean enrichment analysis on a co-expression network to predict disease associations with tissue specific and non-specific lincRNAs respectively.

6.3.4.4 K-RWRH [102]

Kernel-based Random Walk with Restart method in a heterogeneous network is an extension to the RWRH algorithm proposed in [117]. Here the heterogeneous network is constructed by a disease-disease similarity matrix, lincRNA-lincRNA similarity matrix and known lincRNA-disease relationship matrix. It predicts potential lincRNA-disease association through simulating the random walk with restart from a given set of known disease and lincRNA seed nodes. After some steps, the steady state probability distribution is obtained. The lincRNAs and the diseases (representing the nodes in the network) are ranked based on the steady state probabilities.

6.3.5 Experimental Setup

Note that the first method does not use any of lincRNA or disease specific features such as TF-IDF, disease phenotype similarity, RNA-seq provided expression profiles, ChIP-seq provided Transcription factor binding sites, functional GO annotations and SNP linkages. However, remaining three methods also do not use any of these features except the expression profiles. For all the methods, including the standard IMC and our proposed RIMC we rank the predictions using the estimated values corresponding to a lincRNA for each of the diseases considered in our study. For the standard IMC method and our proposed RIMC method, we construct the lincRNA and disease feature matrices $X \in \mathbb{R}^{M\times m}$ with $m = 342$ and $Y \in \mathbb{R}^{N\times n}$ with $n = 200$ for the set of $M = 6540$ lincRNAs and $N = 2418$ diseases (in terms of OMIM phenotypes). We set the best parameters values for each of the methods through cross-validation except LRLSLDA, in which case we set the eight parameter values as suggested by the authors. All the experiments were run on an Intel(R) Core (TM) i5-2400 CPU running at 3.10GHz, 4-cores, 6MB L2-cache, 12GB of RAM (DDR3 1333) hosting Ubuntu 14.04 operating system.

6.3.6 Evaluation Metrics

The lincRNA-disease association prediction algorithm under evaluation computes a ranking score for each candidate disease (i.e., disease that is not reported to be connected with a lincRNA before) and returns the top-k highest ranked diseases as recommendations to a target lincRNA. Thus, for the evaluation of the predictive accuracy, the goal is to find out how many disease-lincRNA associations previously marked off in the preprocessing step recovered in the returned disease recommendations. More specifically, we used two evaluation metrics: (1) the ratio of recovered diseases to the k recommended diseases for the target lincRNA, and (2) the ratio of recovered diseases to the set of diseases deleted in preprocessing [118]. The first metric is called *precision*@k and the latter is known as *recall*@k. The metrics are defined in Equation 6.39 and 6.40. In our experiment, we tested the performance when $k = \{5, 10, 20, 40, 50, 60, 70, 80, 90, 100\}$.

$$precision@k = \frac{1}{N_l}\sum_{l=1}^{N_l}\frac{|P_l(k) \cap D_l|}{k} \tag{6.39}$$

$$recall@k = \frac{1}{N_l}\sum_{l=1}^{N_l}\frac{|P_l(k) \cap D_l|}{|D_l|}, \tag{6.40}$$

where $P_l(k)$ being the top-k ranked diseases for lincRNA l, D_l is the set of diseases related to the lincRNA l marked off during the training step, N_l is the total number lincRNAs in the evaluation dataset. We performed 10-fold cross-validation to measure the performance of our proposed RIMC method as well as the competitive methods. It is worth noting that the *precision*@k and *recall*@k in our experiments are not high. This is because of the sparsity in the lincRNA-disease association dataset having density only 0.003. Similar performance can also be observed in other association recommendation works by [119] and [120] just to name a few. Therefore, the low precision obtained in our experiments is reasonable. In this article, we emphasize on comparing relative performance of the methods rather than their absolute performance.

6.4 Results and Discussion

6.4.1 Effect of the parameter settings

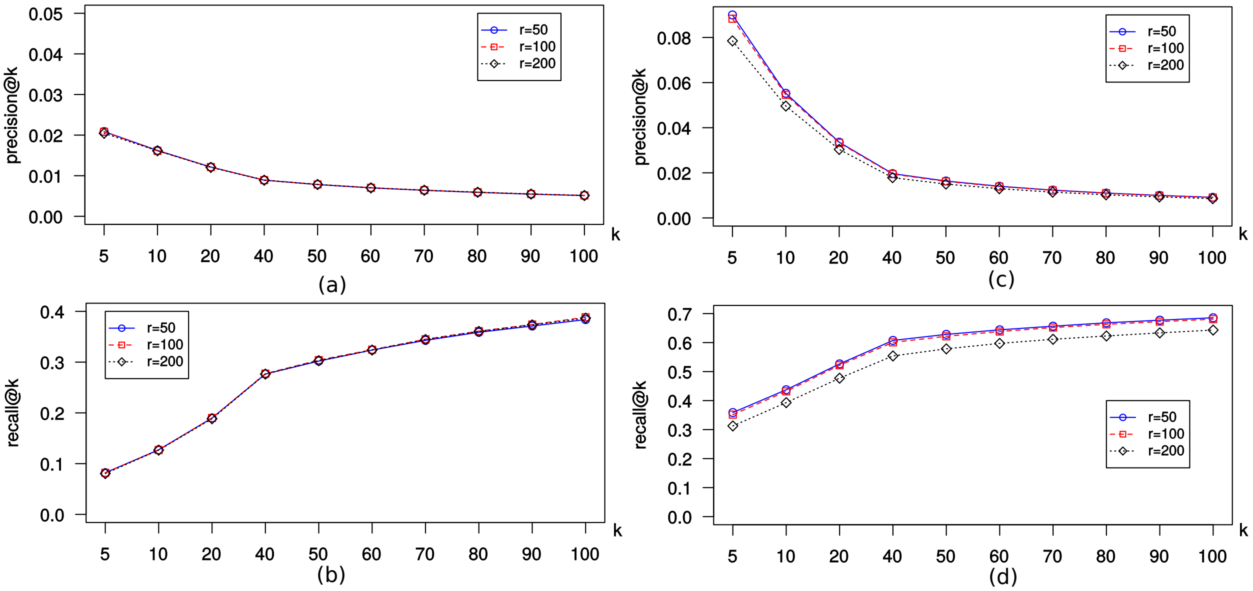


Figure 6.2: Comparing *precision*@k and *recall*@k of the standard IMC and the robust IMC on different values of rank parameter r. Error bars are not shown in this plot as the standard deviations are too small. (a-b): Standard IMC shows a slight improvement of precision@k and recall@k with increasing value for r. (c-d): Robust IMC shows the best performance when $r = 50$.

The parameters to the RIMC method are the rank (r) of the basis W and the coefficient H matrices and the regularization parameters λ_1, λ_2 for W and H matrices respectively. The *precision*@k and *recall*@k performance of the method along with the standard IMC formulation is presented in Figure 6.2. It is evident from the figure that the top-k association retrieval performance varies with the changes in the rank parameter. We varied the rank parameter from 50 to 200, which is equal to $\min(\mathrm{rank}(X), \mathrm{rank}(Y))$, where X, Y are the lincRNA and disease feature matrices respectively. We see that there is little to no boost up of the performance from the changes of rank value from 50 to 100. However, continuing to increase the rank to the maximum possible value, 200, the performance degrades in terms of both the *precision*@k and *recall*@k. This issue can be justified as an over-fitting problem. We set the regularization parameters $\lambda_1 = \lambda_2 = 1.0$ in the optimization functions as we found that the predictive performance degrades if the parameters are set to values deviating far away from 1.0 (data not shown). We found the cut-off value for the parameters through cross-validation for all possible values in the range (0.1, 10.0).

6.4.2 True LincRNA-Disease Association Retrieval

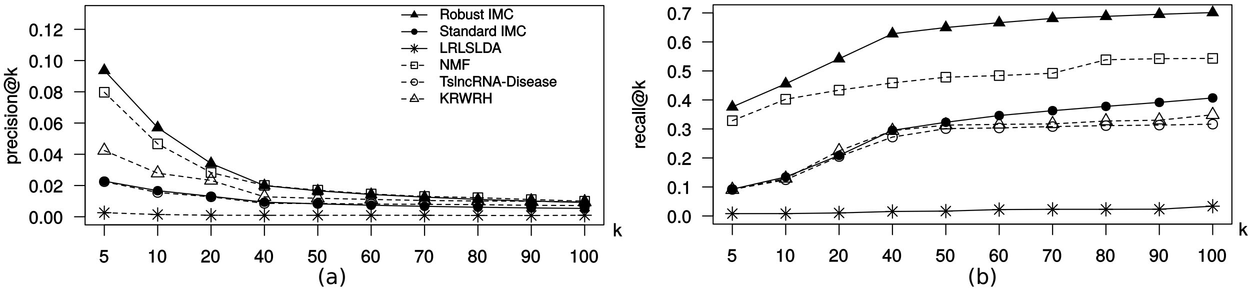


Figure 6.3: Comparision of lincRNA-disease association methods. Error bars are not shown in this plot as the standard deviations are too small. (a) k-vs-*precision*@k plot for all the six methods. (b) k-vs-*recall*@k plot for the six methods. The standard IMC and the proposed RIMC method is trained with 342 lincRNA features and 200 disease features, with a rank, $r = 100$. NMF was trained with the same binary association matrix we used in the IMC experiments with a rank $r = 100$.

The 10-fold cross-validation results on 2418 OMIM diseases are presented in Figure 6.3. The Y-axis in the plots (a,b) gives the *precision*@k and *recall*@k scores for various k values in the horizontal X-axis. We observe that the proposed RIMC significantly dominates the competitive methods over all k values. The best *precision*@k and *recall*@k recorded are close to 10% and 38% respectively at the top-5 association prediction cases. The matrix completion on A performs significantly better than the three other baseline algorithms. LRLSLDA performs worse in terms of *precision*@k and the *recall*@k scores. This is because, the method only relies on known association matrix and the expression profiles of the lincRNAs. Moreover, it comes with a lot of parameters to learn, and is not easily scalable in larger context, like ours, because of the complex `pinv` operations to compute the Laplacians.

6.4.3 Induction on new Associations

We investigate the power of the inductive learning, readily provided by the IMC formulations in both the standard IMC and our proposed RIMC to predict the associations between

i) a new disease to a well studied lincRNA,

ii) a new lincRNA to a well studied disease

iii) a new disease to a new lincRNA.

The only assumption in the IMC framework for induction is that all the features for the novel disease (or the lincRNA or both) may be available during prediction. Here we underline the power of inductive learning of the trained models which is readily usable for prediction for a new test lincRNA or disease (or both) entries even though the entries were absent during the training. Note that all the baseline methods other than the standard IMC are missing from each of the plots in Figure 6.4,6.5, 6.6 as none could make such prediction of the novel disease and lincRNA associations using the respective learned models because of their inherent transductive formulations.

6.4.3.1 Induction experiments on new LincRNAs

We randomly picked 10% of the subject lincRNA entries and the corresponding associations from our datasets (X, the lincRNA featureset and A, the lincRNA-disease association data matrix) and all of the existing disease entries and the respective associations from Y and A. This set-aside entries will be served as test-set for the first batch of induction experiment. Both the standard IMC and the RIMC were then trained with the remaining entries and associations. We evaluate each of the models with the respective set-aside test cases. We repeat the above steps 10 times and recorded the average predictive scores for the comparison. Figure 6.4 illustrates the performance comparison of the standard IMC and our proposed robust IMC for the both new diseases and the new lincRNAs. The *precision*@k curve for the robust IMC show a superior performance than that of the standard IMC based approach for predicting upto the top-50 disease associations with the new lincRNAs. For higher values of k in the top-k predictions, both RIMC and the standard IMC show similar performance. But in terms of numerical precision, RIMC exceeds the standard IMC. However, in the *recall*@k curve, we can see that RIMC performs superior than standard IMC method.

6.4.3.2 Induction experiments on new Diseases

Here, we randomly picked 10% of the subject disease entries and the corresponding associations from our datasets (Y, the disease featureset and A, the lincRNA-disease association data) and the subject lincRNA entries and the respective associations from X and A, and these sets will be considered as test set. Both the standard

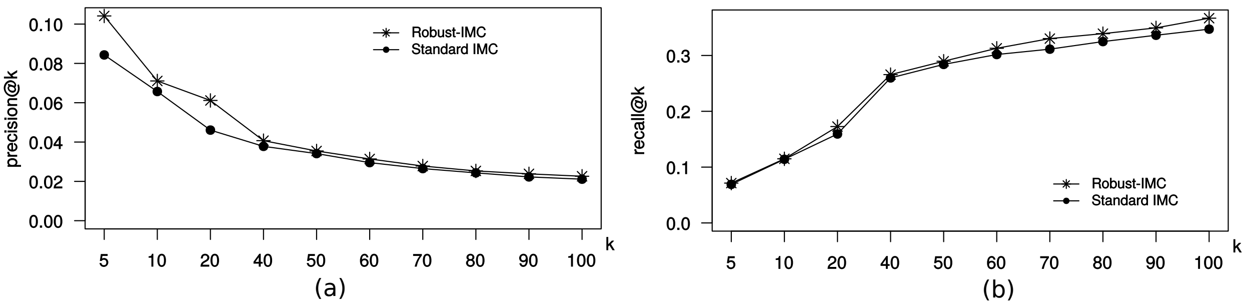


Figure 6.4: Performance comparison of the standard IMC and our proposed robust IMC for induction on existing set of diseases and new lincRNAs. Error bars are not shown in this plot as the standard deviations are too small. (a) k-vs-*precision*@k plot for the two methods, (b) k-vs-*recall*@k plot for the two methods.

IMC and the RIMC were then trained with the remaining entries and associations. We evaluate each of the models with the respective set-aside test cases. We repeat the above steps 10 times and recorded the average predictive scores for the comparison. Figure 6.5 illustrates the performance comparison of the standard IMC and our proposed robust IMC for the both new diseases and the new lincRNAs. The *precision*@k and *recall*@k curves for the robust IMC show a superior performance than that of the standard IMC based approach for predicting upto the top-50 lincRNA associations with the novel diseases.

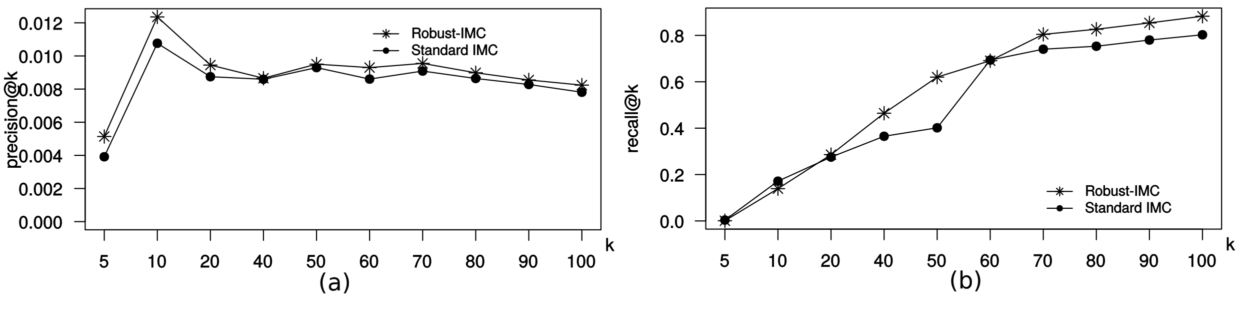


Figure 6.5: Performance comparison of the standard IMC and our proposed robust IMC for induction on new diseases and existing set of lincRNAs. Error bars are not shown in this plot as the standard deviations are too small. (a) k-vs-*precision*@k plot for the two methods, (b) k-vs-*recall*@k plot for the two methods.

6.4.3.3 Induction experiments on both new LincRNAs and new Diseases

Finally, in this batch of induction experiment, we randomly picked 5% of the subject disease entries, and 5% of the subject lincRNA entries and the corresponding associations from our full set of existing datasets (X, Y, the lincRNA and disease feature-sets respectively and A, the lincRNA-disease association data) which will be considered as a test-set for the experiment. Both the standard IMC and the RIMC were then trained with the remaining entries and associations. We evaluate each of the models with the respective set-aside test cases. We repeat the above steps 10 times and recorded the average predictive scores for the comparison. Figure 6.6 illustrates the performance comparison of the standard IMC and our proposed robust IMC for the both new diseases and the new lincRNAs. The *precision@k* plot of for the robust IMC show a superior performance than that of the standard IMC based approach for predicting for both lower and higher values of k in the top-k association ranking with the novel diseases. However, from the *recall@k* cure of the both algorithms, we can see that both RIMC and standard IMC performs similar in the top-k association prediction problem.

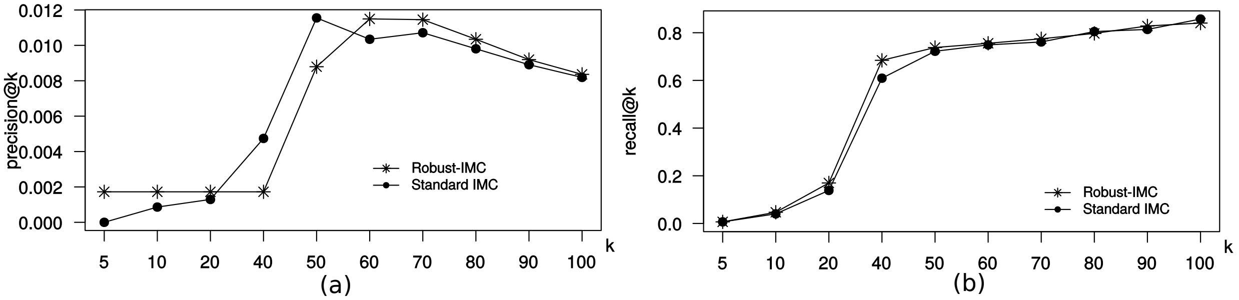


Figure 6.6: Performance comparison of the standard IMC and our proposed robust IMC for induction on both new diseases and new lincRNAs. Error bars are not shown in this plot as the standard deviations are too small. (a) k-vs-*precision@k* plot for the two methods, (b) k-vs-*recall@k* plot for the two methods.

6.5 Conclusions

In this manuscript, we proposed a robust formulation of the inductive matrix completion method using $\ell_{2,1}$ norm. We applied our proposed method for predicting associations between the long intergenic non-coding RNAs (lincRNAs) and diseases. The method presents an integration interface for various categories of features of both the lincRNAs and diseases obtained through different independent data sources for explaining the relationships between the two entities. The proposed method can handle inherent noises and outliers in the dataset, and was shown to outperform the ℓ_2 norm based standard IMC formulation. Besides the standard IMC formulation, our proposed method also outperformed other four lincRNA-disease association solutions. In our experiments we found that our method performs the best in predicting associations between already studied set of lincRNAs and diseases as well as between novel set of lincRNAs and diseases which makes the method a suitable association prediction tool for the biologists.

Two possible extensions to our method presented here can be made: (i) the inductive framework (as opposed to its transductive versions) is not limited to the types of features used in the experiments we presented, as new sources of information can be integrated easily via rank-1 updates. (ii) The framework itself can be extended to address the missing value problem inherent to the side information of the two respective entities.

REFERENCES

[1] E. Pennisi, "ENCODE project writes eulogy for junk DNA," *Science*, vol. 337, no. 6099, pp. 1159–1161, 2012.

[2] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[3] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold, "Genome-wide mapping of in vivo protein-DNA interactions," *Science*, vol. 316, no. 5830, pp. 1497–1502, 2007.

[4] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures," *Monatshefte für Chemie/Chemical Monthly*, vol. 125, no. 2, pp. 167–188, 1994.

[5] J. Gong, W. Liu, J. Zhang, X. Miao, and A.-Y. Guo, "lncRNASNP: a database of SNPs in lncRNAs and their potential functions in human and mouse," *Nucleic acids research*, vol. 43, no. D1, pp. D181–D186, 2015.

[6] A. K. Biswas, B. Zhang, X. Wu, and J. X. Gao, "CNCTDiscriminator: coding and noncoding transcript discriminator – an excursion through hypothesis learning and ensemble learning approaches," *Journal of Bioinformatics and Computational Biology (JBCB)*, vol. 11, no. 05, p. 1342002, 2013.

[7] A. K. Biswas, J. X. Gao, B. Zhang, and X. Wu, "Integrating RNA-seq transcript signals, primary and secondary structure information in differentiating coding and non-coding RNA transcripts," in *The 5th International Conference on Bioinformatics and Computational Biology (BICoB)*. ISCA, 2013.

[8] A. K. Biswas, B. Zhang, X. Wu, and J. X. Gao, "An Information Integration Approach for Classifying Coding and Non-Coding Genomic Data," in *The Proceedings of the Second International Conference on Communications, Signal Processing, and Systems (CSPS).* Springer International Publishing, 2014, pp. 1085–1093.

[9] A. K. Biswas and J. X. Gao, "PR2S2Clust: Patched RNA-seq read segments' structure-oriented clustering," *Journal of Bioinformatics and Computational Biology (JBCB)*, p. 1650027, 2016.

[10] A. K. Biswas, B. Zhang, X. Wu, and J. X. Gao, "QLZCClust: Quaternary lempel-Ziv complexity based clustering of the RNA-seq read block segments," in *IEEE International Conference on Bioinformatics and Bioengineering (BIBE).* IEEE, 2013, pp. 1–4.

[11] A. K. Biswas, M. Kang, D.-C. Kim, C. H. Ding, B. Zhang, X. Wu, and J. X. Gao, "Inferring disease associations of the long non-coding RNAs through non-negative matrix factorization," *Network Modeling Analysis in Health Informatics and Bioinformatics (NetMAHIB)*, vol. 4, no. 1, pp. 1–17, 2015.

[12] A. K. Biswas, J. X. Gao, B. Zhang, and X. Wu, "NMF-Based LncRNA-Disease Association Inference and Bi-Clustering," in *IEEE International Conference on Bioinformatics and BioEngineering (BIBE).* IEEE, 2014, pp. 97–104.

[13] A. K. Biswas, D.-C. Kim, M. Kang, and J. X. Gao, "Robust Inductive Matrix Completion Strategy to Explore Associations between LincRNAs and Human Disease Phenotypes," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, 2016.

[14] A. Machado-Lima, H. A. del Portillo, and A. M. Durham, "Computational methods in noncoding RNA research," *Journal of Mathematical Biology*, vol. 56, no. 1-2, pp. 15–49, 2008.

[15] M. E. Dinger, K. C. Pang, T. R. Mercer, and J. S. Mattick, "Differentiating Protein-Coding and Noncoding RNA: Challenges and Ambiguities," *PLoS Computational Biology*, vol. 4, no. 11, p. e1000176, 2008.

[16] M. Clamp, B. Fry, M. Kamal, X. Xie, J. Cuff, M. F. Lin, M. Kellis, K. Lindblad-Toh, and E. S. Lander, "Distinguishing protein-coding and noncoding genes in the human genome," *Proceedings of the National Academy of Sciences*, vol. 104, no. 49, pp. 19 428–19 433, 2007.

[17] A. R. Gruber, S. Findeiß, S. Washietl, I. L. Hofacker, and P. F. Stadler, "RNAz 2.0: Improved Noncoding RNA Detection," in *Pacific Symposium on Biocomputing*, vol. 15, 2010, pp. 69–79.

[18] N. B. Leontis, A. Lescoute, and E. Westhof, "The building blocks and motifs of RNA architecture," *Current Opinion in Structural Biology*, vol. 16, no. 3, pp. 279–287, 2006.

[19] E. Rivas and S. R. Eddy, "Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs," *Bioinformatics*, vol. 16, no. 7, pp. 583–605, 2000.

[20] L. Kong, Y. Zhang, Z. Ye, X. Liu, S. Zhao, L. Wei, and G. Gao, "CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine," *Nucleic Acids Research*, vol. 35, no. suppl 2, pp. W345–W349, 2007.

[21] R. T. Arrial, R. C. Togawa, and M. M. Brigido, "Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus *Paracoccidioides brasiliensis*," *BMC Bioinformatics*, vol. 10, no. 1, p. 239, 2009.

[22] L. Wang, H. J. Park, S. Dasari, S. Wang, J.-P. Kocher, and W. Li, “CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model,” *Nucleic Acids Research*, vol. 41, no. 6, pp. e74–e74, 2013.

[23] D. Langenberger, C. Bermudez-Santana, J. Hertel, S. Hoffmann, P. Khaitovich, and P. Stadler, “Evidence for human microRNA-offset RNAs in small RNA sequencing data,” *Bioinformatics*, vol. 25, no. 18, pp. 2298–2301, 2009.

[24] G. Chen, K. Yin, L. Shi, Y. Fang, Y. Qi, P. Li, J. Luo, B. He, M. Liu, and T. Shi, “Comparative Analysis of Human Protein-Coding and Noncoding RNAs between Brain and 10 Mixed Cell Lines by RNA-Seq,” *PloS ONE*, vol. 6, no. 11, p. e28318, 2011.

[25] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, *et al.*, “GENCODE: The reference human genome annotation for The ENCODE Project,” *Genome Research*, vol. 22, no. 9, pp. 1760–1774, 2012.

[26] K. D. Pruitt, T. Tatusova, G. R. Brown, and D. R. Maglott, “NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy,” *Nucleic Acids Research*, vol. 40, no. D1, pp. D130–D135, 2012.

[27] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[28] P. Rice, I. Longden, and A. Bleasby, “EMBOSS: the European molecular biology open software suite,” *Trends in Genetics*, vol. 16, no. 6, pp. 276–277, 2000.

[29] A. Mortazavi, B. Williams, K. McCue, L. Schaeffer, and B. Wold, “Mapping and quantifying mammalian transcriptomes by RNA-Seq,” *Nature Methods*, vol. 5, no. 7, pp. 621–628, 2008.

[30] D. Ramsköld, E. T. Wang, C. B. Burge, and R. Sandberg, "An Abundance of Ubiquitously Expressed Genes Revealed by Tissue Transcriptome Sequence Data," *PLoS Computational Biology*, vol. 5, no. 12, p. e1000598, 2009.

[31] T. Joachims, "Training linear SVMs in linear time," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2006, pp. 217–226.

[32] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[33] C. Trapnell, L. Pachter, and S. L. Salzberg, "Tophat: discovering splice junctions with RNA-seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[34] D. Langenberger, S. Pundhir, C. T. Ekstrøm, P. F. Stadler, S. Hoffmann, and J. Gorodkin, "deepblockalign: a tool for aligning RNA-seq profiles of read block patterns," *Bioinformatics*, vol. 28, no. 1, pp. 17–24, 2012.

[35] S. Pundhir and J. Gorodkin, "MicroRNA discovery by similarity search to a database of RNA-seq profiles," *Frontiers in genetics*, vol. 4, p. 133, 2013.

[36] D. H. Mathews and D. H. Turner, "Prediction of RNA secondary structure by free energy minimization," *Current opinion in structural biology*, vol. 16, no. 3, pp. 270–278, 2006.

[37] S. Washietl, I. L. Hofacker, M. Lukasser, A. Hüttenhofer, and P. F. Stadler, "Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome," *Nature biotechnology*, vol. 23, no. 11, pp. 1383–1390, 2005.

[38] J. S. Mattick and I. V. Makunin, "Non-coding RNA," *Human molecular genetics*, vol. 15, no. suppl 1, pp. R17–R29, 2006.

[39] D. P. Bartel, "MicroRNAs: genomics, biogenesis, mechanism, and function," *cell*, vol. 116, no. 2, pp. 281–297, 2004.

[40] M. A. Carmell and G. J. Hannon, “RNase III enzymes and the initiation of gene silencing,” *Nature structural & molecular biology*, vol. 11, no. 3, pp. 214–218, 2004.

[41] S. Washietl, I. L. Hofacker, and P. F. Stadler, “Fast and reliable prediction of noncoding RNAs,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 7, pp. 2454–2459, 2005.

[42] S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen, “Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering,” *PLoS computational biology*, vol. 3, no. 4, p. e65, 2007.

[43] J. H. Havgaard, E. Torarinsson, and J. Gorodkin, “Fast pairwise structural RNA alignments by pruning of the dynamical programming matrix,” *PLOS computational biology*, vol. 3, no. 10, p. e193, 2007.

[44] S. Heyne, F. Costa, D. Rose, and R. Backofen, “GraphClust: alignment-free structural clustering of local RNA secondary structures,” *Bioinformatics*, vol. 28, no. 12, pp. i224–i232, 2012.

[45] A. L. Fred and A. K. Jain, “Combining multiple clusterings using evidence accumulation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 835–850, 2005.

[46] A. Strehl and J. Ghosh, “Cluster ensembles — a knowledge reuse framework for combining multiple partitions,” *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[47] L. Hubert and P. Arabie, “Comparing partitions,” *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[48] R. D. Morin, M. D. O’Connor, M. Griffith, F. Kuchenbauer, A. Delaney, A.-L. Prabhu, Y. Zhao, H. McDonald, T. Zeng, M. Hirst, *et al.*, “Application of

massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells," *Genome Research*, vol. 18, no. 4, pp. 610–621, 2008.

[49] M. Somel, S. Guo, N. Fu, Z. Yan, H. Hu, Y. Xu, Y. Yuan, Z. Ning, Y. Hu, C. Menzel, *et al.*, "MicroRNA, mRNA, and protein expression link development and aging in human and macaque brain," *Genome research*, vol. 20, no. 9, pp. 1207–1218, 2010.

[50] K. Sato, M. Hamada, K. Asai, and T. Mituyama, "CENTROIDFOLD: a web server for RNA secondary structure prediction," *Nucleic acids research*, vol. 37, no. suppl 2, pp. W277–W280, 2009.

[51] A. Lempel and J. Ziv, "On the complexity of finite sequences," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 75–81, 1976.

[52] H. H. Otu and K. Sayood, "A new sequence distance measure for phylogenetic tree construction," *Bioinformatics*, vol. 19, no. 16, pp. 2122–2130, 2003.

[53] G. V. Bard, "Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric," in *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*. Australian Computer Society, Inc., 2007, pp. 117–124.

[54] Z. Fang and J. Wang, "Efficient identifications of structural similarities for graphs," *Journal of Combinatorial Optimization*, vol. 27, no. 2, pp. 209–220, 2014.

[55] H. Wang, Y. Yang, H. Wang, and D. Chen, "Soft-voting clustering ensemble," in *Multiple Classifier Systems*. Springer, 2013, pp. 307–318.

[56] E. Dimitriadou, A. Weingessel, and K. Hornik, "Voting-merging: An ensemble method for clustering," in *Artificial Neural NetworksâĂŤICANN 2001*. Springer, 2001, pp. 217–224.

[57] N. Stepenosky, D. Green, J. Kounios, C. M. Clark, and R. Polika, "Majority vote and decision template based ensemble classifiers trained on event related potentials for early diagnosis of Alzheimer's disease," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5. IEEE, 2006, pp. V–V.

[58] S. Will, T. Joshi, I. L. Hofacker, P. F. Stadler, and R. Backofen, "LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs," *RNA*, vol. 18, no. 5, pp. 900–914, 2012.

[59] J. H. Havgaard, R. B. Lyngsø, and J. Gorodkin, "The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search," *Nucleic Acids Research*, vol. 33, no. suppl 2, pp. W650–W653, 2005.

[60] F. Crick, "Central Dogma of Molecular Biology," *Nature*, vol. 227, no. 5258, pp. 561–563, 1970.

[61] R. J. Taft, K. C. Pang, T. R. Mercer, M. Dinger, and J. S. Mattick, "Non-coding RNAs: regulators of disease," *The Journal of pathology*, vol. 220, no. 2, pp. 126–139, 2010.

[62] J. E. Wilusz, H. Sunwoo, and D. L. Spector, "Long noncoding RNAs: functional surprises from the RNA world," *Genes & development*, vol. 23, no. 13, pp. 1494–1504, 2009.

[63] S. Chung, H. Nakagawa, M. Uemura, L. Piao, K. Ashikawa, N. Hosono, R. Takata, S. Akamatsu, T. Kawaguchi, T. Morizono, *et al.*, "Association of a novel long non-coding RNA in 8q24 with prostate cancer susceptibility," *Cancer science*, vol. 102, no. 1, pp. 245–252, 2011.

[64] L. D. Sacco, A. Baldassarre, and A. Masotti, "Bioinformatics tools and novel challenges in long non-coding RNAs (lncRNAs) functional analysis," *International journal of molecular sciences*, vol. 13, no. 1, pp. 97–114, 2011.

[65] P. Kapranov, J. Cheng, S. Dike, D. A. Nix, R. Duttagupta, A. T. Willingham, P. F. Stadler, J. Hertel, J. Hackermüller, I. L. Hofacker, *et al.*, “RNA maps reveal new RNA classes and a possible function for pervasive transcription,” *Science*, vol. 316, no. 5830, pp. 1484–1488, 2007.

[66] T. R. Mercer, M. E. Dinger, and J. S. Mattick, “Long non-coding RNAs: insights into functions,” *Nature Reviews Genetics*, vol. 10, no. 3, pp. 155–159, 2009.

[67] G. Chen, Z. Wang, D. Wang, C. Qiu, M. Liu, X. Chen, Q. Zhang, G. Yan, and Q. Cui, “LncRNADisease: a database for long-non-coding RNA-associated diseases,” *Nucleic acids research*, vol. 41, no. D1, pp. D983–D986, 2013.

[68] X. Chen and G.-Y. Yan, “Novel human lncRNA–disease association inference based on lncRNA expression profiles,” *Bioinformatics*, p. btt426, 2013.

[69] Q. Liao, C. Liu, X. Yuan, S. Kang, R. Miao, H. Xiao, G. Zhao, H. Luo, D. Bu, H. Zhao, *et al.*, “Large-scale prediction of long non-coding RNA functions in a coding–non-coding gene co-expression network,” *Nucleic acids research*, vol. 39, no. 9, pp. 3864–3878, 2011.

[70] X. Guo, L. Gao, Q. Liao, H. Xiao, X. Ma, X. Yang, H. Luo, G. Zhao, D. Bu, F. Jiao, *et al.*, “Long non-coding RNAs function annotation: a global prediction method based on bi-colored networks,” *Nucleic acids research*, vol. 41, no. 2, pp. e35–e35, 2013.

[71] X. Yang, L. Gao, X. Guo, X. Shi, H. Wu, F. Song, and B. Wang, “A Network Based Method for Analysis of lncRNA-Disease Associations and Prediction of lncRNAs Implicated in Diseases,” *PLOS ONE*, vol. 9, no. 1, p. e87797, 2014.

[72] M. N. Cabili, C. Trapnell, L. Goff, M. Koziol, B. Tazon-Vega, A. Regev, and J. L. Rinn, “Integrative annotation of human large intergenic noncoding RNAs

reveals global properties and specific subclasses," *Genes & development*, vol. 25, no. 18, pp. 1915–1927, 2011.

[73] J.-H. Li, S. Liu, H. Zhou, L.-H. Qu, and J.-H. Yang, "starBase v2.0: decoding miRNA-ceRNA, miRNA-ncRNA and protein–RNA interaction networks from large-scale CLIP-Seq data," *Nucleic acids research*, vol. 42, no. D1, pp. D92–D97, 2014.

[74] J. Yuan, W. Wu, C. Xie, G. Zhao, Y. Zhao, and R. Chen, "NPInter v2.0: an updated database of ncRNA interactions," *Nucleic acids research*, vol. 42, no. D1, pp. D104–D108, 2014.

[75] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature biotechnology*, vol. 28, no. 5, pp. 511–515, 2010.

[76] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders," *Nucleic acids research*, vol. 33, no. suppl 1, pp. D514–D517, 2005.

[77] A. Bauer-Mehren, M. Rautschka, F. Sanz, and L. I. Furlong, "DisGeNET: a Cytoscape plugin to visualize, integrate, search and analyze gene–disease networks," *Bioinformatics*, vol. 26, no. 22, pp. 2924–2926, 2010.

[78] A. Lin, R. T. Wang, S. Ahn, C. C. Park, and D. J. Smith, "A genome-wide map of human genetic interactions inferred from radiation hybrid genotypes," *Genome research*, vol. 20, no. 8, pp. 1122–1132, 2010.

[79] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[80] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[81] S. Zhang, Q. Li, J. Liu, and X. J. Zhou, "A novel computational framework for simultaneous integration of multiple types of genomic data to identify microRNA-gene regulatory modules," *Bioinformatics*, vol. 27, no. 13, pp. i401–i409, 2011.

[82] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[83] K. Devarajan, "Nonnegative matrix factorization: an analytical and interpretive tool in computational biology," *PLoS computational biology*, vol. 4, no. 7, p. e1000029, 2008.

[84] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *Neural Networks, IEEE Transactions on*, vol. 17, no. 3, pp. 683–695, 2006.

[85] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.* IEEE, 2008, pp. 63–72.

[86] Q. Gu and J. Zhou, "Local learning regularized nonnegative matrix factorization," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[87] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[88] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[89] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[90] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear algebra and its applications*, vol. 416, no. 1, pp. 29–47, 2006.

[91] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the national academy of sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.

[92] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[93] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2907–2912, 1999.

[94] Z.-Y. Zhang, T. Li, C. Ding, X.-W. Ren, and X.-S. Zhang, "Binary matrix factorization for analyzing gene expression data," *Data Mining and Knowledge Discovery*, vol. 20, no. 1, pp. 28–52, 2010.

[95] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, vol. 22, no. 9, pp. 1122–1129, 2006.

[96] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[97] L. N. Hutchins, S. M. Murphy, P. Singh, and J. H. Graber, “Position-dependent motif characterization using non-negative matrix factorization,” *Bioinformatics*, vol. 24, no. 23, pp. 2684–2690, 2008.

[98] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, “Nonsmooth nonnegative matrix factorization (nsNMF),” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 403–415, 2006.

[99] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási, “The human disease network,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 21, pp. 8685–8690, 2007.

[100] R. P. Alexander, G. Fang, J. Rozowsky, M. Snyder, and M. B. Gerstein, “Annotating non-coding regions of the genome,” *Nature Reviews Genetics*, vol. 11, no. 8, pp. 559–571, 2010.

[101] M. Esteller, “Non-coding RNAs in human disease,” *Nature Reviews Genetics*, vol. 12, no. 12, pp. 861–874, 2011.

[102] G. U. Ganegoda, M. Li, W. Wang, and Q. Feng, “Heterogeneous network model to infer human disease-long intergenic non-coding RNA associations,” *NanoBioscience, IEEE Transactions on*, vol. 14, no. 2, pp. 175–183, 2015.

[103] M.-X. Liu, X. Chen, G. Chen, Q.-H. Cui, and G.-Y. Yan, “A computational framework to infer human disease-associated long noncoding RNAs,” *PloS one*, vol. 9, no. 1, p. e84408, 2014.

[104] P. Jain and I. S. Dhillon, “Provable Inductive Matrix Completion,” *arXiv preprint arXiv:1306.0626*, 2013.

[105] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, no. 8, pp. 30–37, 2009.

[106] N. Natarajan and I. S. Dhillon, “Inductive matrix completion for predicting gene–disease associations,” *Bioinformatics*, vol. 30, no. 12, pp. i60–i68, 2014.

[107] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[108] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.

[109] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, “Large-scale Multi-label Learning with Missing Labels,” in *International Conference on Machine Learning (ICML)*, vol. 32, jun 2014.

[110] E. Polak and G. Ribière, “Note sur la convergence de méthodes de directions conjuguées,” *Revue française d’informatique et de recherche opérationnelle, série rouge*, vol. 3, no. 1, pp. 35–43, 1969.

[111] J. C. Gilbert and J. Nocedal, “Global convergence properties of conjugate gradient methods for optimization,” *SIAM Journal on optimization*, vol. 2, no. 1, pp. 21–42, 1992.

[112] J.-H. Yang, J.-H. Li, S. Jiang, H. Zhou, and L.-H. Qu, “ChIPBase: a database for decoding the transcriptional regulation of long non-coding RNA and microRNA genes from ChIP-Seq data,” *Nucleic acids research*, vol. 41, no. D1, pp. D177–D187, 2013.

[113] K. Liu, Z. Yan, Y. Li, and Z. Sun, “Linc2GO: a human LincRNA function annotation resource based on ceRNA hypothesis,” *Bioinformatics*, vol. 29, no. 17, pp. 2221–2222, 2013.

[114] L. Salmena, L. Poliseno, Y. Tay, L. Kats, and P. P. Pandolfi, "A ceRNA hypothesis: the Rosetta Stone of a hidden RNA language?" *Cell*, vol. 146, no. 3, pp. 353–358, 2011.

[115] X. Zhang, L. Zhou, G. Fu, F. Sun, J. Shi, J. Wei, C. Lu, C. Zhou, Q. Yuan, and M. Yang, "The identification of an ESCC susceptibility SNP rs920778 that regulates the expression of lncRNA HOTAIR via a novel intronic enhancer," *Carcinogenesis*, p. bgu103, 2014.

[116] H. Caniza, A. E. Romero, and A. Paccanaro, "A network medicine approach to quantify distance between hereditary disease modules on the interactome," *Scientific reports*, vol. 5, 2015.

[117] Y. Li and J. C. Patra, "Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network," *Bioinformatics*, vol. 26, no. 9, pp. 1219–1224, 2010.

[118] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2014, pp. 831–840.

[119] D. Shin, S. Cetintas, K.-C. Lee, and I. S. Dhillon, "Tumblr blog recommendation with boosted inductive matrix completion," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* ACM, 2015, pp. 203–212.

[120] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-Aware Point of Interest Recommendation on Location-Based Social Networks," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1721–1727, 2015.

[121] K. Yoshihara, A. Tajima, S. Adachi, J. Quan, M. Sekine, H. Kase, T. Yahata, I. Inoue, and K. Tanaka, "Germline copy number variations in BRCA1-

associated ovarian cancer patients," *Genes, Chromosomes and Cancer*, vol. 50, no. 3, pp. 167–177, 2011.

[122] B. Kunkle, Q. Felty, F. Trevino, and D. Roy, "Oncomine meta-analysis of breast cancer microarray data identifies upregulation of NRF-1 expression in human breast carcinoma," *Distribution*, pp. 715–719, 2009.

[123] W. Liu, N. Zheng, and Q. You, "Nonnegative matrix factorization and its applications in pattern recognition," *Chinese Science Bulletin*, vol. 51, no. 1, pp. 7–18, 2006.

[124] D. Luo, C. Ding, and H. Huang, "Towards Structural Sparsity: An Explicit l2/l0 Approach," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 344–353.

[125] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell - 2, 1$-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[126] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using l-2,1-norm," in *Proceedings of the 20th ACM international conference on Information and knowledge management.* ACM, 2011, pp. 673–682.

[127] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

BIOGRAPHICAL STATEMENT

Ashis Kumer Biswas was born in Tangail, Bangladesh. He received his Bachelor and Masters' degree in Computer Science and Engineering from University of Dhaka, Bangladesh, in 2007 and 2008 respectively. In 2010, he joined as a Lecturer in the same university and is now on a study leave to pursue his Ph.D at the University of Texas at Arlington. His current research interests include high throughput sequencing databases, characterizing non-coding RNAs and machine learning. His paper titled "NMF-based LncRNA-Disease Association Inference and Bi-clustering" has received the best paper award in the 14th IEEE International Conference on BioInformatics and BioEngineering (BIBE-2014).