GENOMIC INTEGRATIVE ANALYSIS TO IMPROVE FUSION TRANSCRIPT DETECTION, LIQUID ASSOCIATION AND BICLUSTERING

by

Shuchang Liu

BS, Shanghai Jiao Tong University, 2012

Submitted to the Graduate Faculty of

the School of Medicine in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Shuchang Liu

It was defended on

April 18, 2017

and approved by

Takis Benos, PhD, Professor, Department of Computational and Systems Biology

Yongseok Park, PhD, Assistant Professor, Department of Biostatistic

Seyoung Kim, PhD, Assistant Professor, Department of Computational Biology

Dissertation Director: George C. Tseng, PhD, Professor, Department of Biostatistics,

Human Genetics and Computational Biology

GENOMIC INTEGRATIVE ANALYSIS TO IMPROVE FUSION TRANSCRIPT DETECTION, LIQUID ASSOCIATION AND BICLUSTERING

Shuchang Liu, PhD

University of Pittsburgh, 2017

More data provide more possibilities. Growing number of genomic data provide new perspectives to understand some complex biological problems. Many algorithms for single-study have been developed, however, their results are not stable for small sample size or overwhelmed by study-specific signals. Taking the advantage of high throughput genomic data from multiple cohorts, in this dissertation, we are able to detect novel fusion transcripts, explore complex gene regulations and discovery disease subtypes within an integrative analysis framework.

In the first project, we evaluated 15 fusion transcript detection tools for paired-end RNA-seq data. Though no single method had distinguished performance over the others, several top tools were selected according to their F-measures. We further developed a fusion meta-caller algorithm by combining top methods to re-prioritize candidate fusion transcripts. The results showed that our meta-caller can successfully balance precision and recall compared to any single fusion detection tool.

In the second project, we extended liquid association to two meta-analytic frameworks (MetaLA and MetaMLA). Liquid association is the dynamic gene-gene correlation depending on the expression level of a third gene. Our MetaLA and MetaMLA provided stronger detection signals and more consistent and stable results compared to single-study analysis. When applied our method to five Yeast datasets related to environmental changes, genes in the top triplets were highly enriched in fundamental biological processes corresponding to environmental changes.

In the third project, we extended the plaid model from single-study analysis to multiple cohorts for bicluster detection. Our meta-biclustering algorithm can successfully discovery biclusters with higher Jaccard accuracy toward large noise and small sample size. We also introduced the concept of gap statistic for pruning parameter estimation. In addition, biclusters detected from five breast cancer mRNA expression cohorts can successfully select genes highly associated with many breast cancer related pathways and split samples with significantly different survival behaviors.

In conclusion, we improved the fusion transcripts detection, liquid association analysis and bicluster discovery through integrative-analysis frameworks. These results provided strong evidence of gene fusion structure variation, three-way gene regulation and disease subtype detection, and thus contribute to better understanding of complex disease mechanism ultimately.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 HIGH-THROUGHPUT GENOMIC DATA

The concept of central dogma was proposed in molecular biology to explain the flow of genetic information [Crick, 1958; Crick et al., 1970]. For most creatures, DNA carrying the genetic information are first transcribed into RNA, and then RNA are translated into proteins to construct the organism. In order to explore the functional process, especially the disease mechanism, data from different molecular levels are collected to learn the mechanisms of regulation, interaction, modification, etc. These multiple types of omics data are listed in but not limited to Fig. 1.1. At DNA level, single nucleotide polymorphisms (SNPs) data, gene mutation (insertion, deletion and mutation) and copy number variation (CNV) data are widely learned. At the epigenetic level, data for DNA methylation, histone modification, transcription factor (TF) regulation are collected. At RNA level, messenger RNA (mRNA) and micro-RNA (miRNA) data are analyzed to study alternative splicing, trans-splicing or other modification mechanisms. At protein level, data for ribosomal RNA (rRNA), post-translation modification or protein-folding are measured for analysis.

Genome refers to a complete set of DNA within a cell of an organism, and genomics is the study of genome, including structure, function, evolution, mapping, regulation or modification of the genes as well as the influence of environmental factors. Thus genomics offers new opportunities to study many complex diseases and their diagnostic methods. In order to explore genes in whole-genome scale, many high throughput technologies are developed and applied in real application. Data generated from different platforms require specific pre-processing and data mining pipelines. In the following, we will introduce two main platforms – microarray and next generation sequencing (NGS) technologies, as well as

their data format and some conventional pre-processing pipelines.

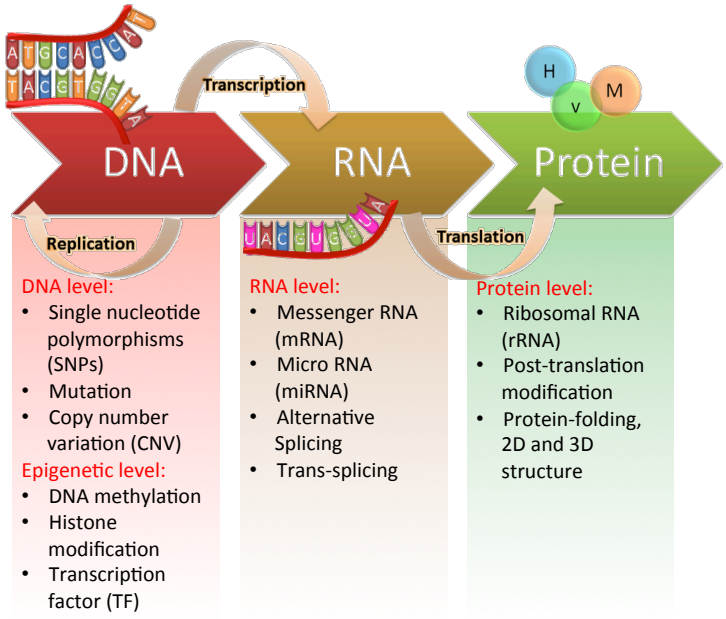


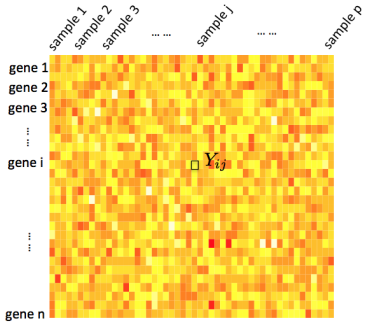
Figure 1.1: Protein expression in central dogma.

1.1.1 Microarray

A microarray is a 2D chip to assay biological signals by hybridizing a large amount of biological material to the probes attached to it in parallel. The technology was proposed in mid 90's and nowadays has been largely applied to high throughput genomics data analysis: expression profiling, SNP detection, ChIP on chip, etc. mRNA expression profiling is one of the microarray applications that can simultaneously monitor the expression levels of tens of thousands of transcripts with high sensitivity and specificity. These advantages provides opportunities to study gene association, genes that are differentially expressed between case and control samples, gene expressions responding to environmental factors or developmental stage.

A genomic cohort typically consist of attributes of thousands of genes among multiple samples by combining several microarray experimental data. Take the mRNA expression

profiling as an example. After pre-processing, it forms a data matrix format where conventionally each row represents a probe (or gene) and each column represents a sample (or condition). Fig. 1.2 shows the detailed symbol annotation. Mathematically, the data matrix is defined as $Y \in \mathcal{R}^{n \times p}$, where n and p are total number of genes and samples. Element Y_{ij} represents the expression intensity (or log2 intensity) of gene i for sample j.



Symbol	Annotation
n	number of genes
i	gene index, $i = 1, \ldots, n$
p	number of samples
j	sample index, $j = 1, \ldots, p$
Y_{ij}	expression intensity for gene i in sample j

Figure 1.2: Data Structure for gene expression profile.

Microarray gene expression data matrix usually requires the following pre-processing steps.

1. *Missing value imputation.* Influenced by the background noise, signals for low expressed transcripts are hard to be detected. In this case, probes for some samples will show out NA values if the signal is lower or noisy to a certain cutoff. Many DNA microarray imputation methods have been developed and compared comprehensively [Liew et al., 2011; Moorthy et al., 2014]. In this dissertation, we implemented the simplest k-NN imputation method, where missing value is imputed by the mean value of its k nearest

neighbors defined by pairwise gene distance [Troyanskaya et al., 2001; Altman, 1992].

2. *Sample normalization.* Since different microarrays (for samples or conditions) are measured given slightly different reagent concentration, their intensity range may differ from each other. In order to compare the genes among different samples, each sample should be normalized to the same scale at the very beginning. Based on different normalization purposes, many methods have been proposed [Quackenbush, 2002], where quantile normalization is a widely used one [Amaratunga and Cabrera, 2001].
3. *Probes to genes mapping.* Probes on gene chips are designed to be oligonucleotides that can hybridize to mRNA molecules specifically. The probe can thus quantify the mRNA from which the oligonucleotides sequenced from. In order to study the expression pattern in gene level (instead of probe level), probes need to be matched to their corresponding gene annotations. In real microarray probe design, more than one probe may match to the same gene. In this dissertation, for a given gene, we chose the representative probe which has the highest interquartile range (IQR) among samples.
4. *Gene filtering.* In general, filtering out potentially non-significant genes will gain statistical power [van Iterson et al., 2010; Bourgon et al., 2010]. In order to detect more significant signals under the same FDR control, we filtered out those non-expressed (low mean) and non-informative (low variance) genes in the pre-processing step.

Note that, based on different research purposes, the order and methods of these pre-processing steps need to be adjusted accordingly.

1.1.2 Next generation sequencing

Next generation sequencing (NGS), also known as high-throughput sequencing (HTP) , were developed in the mid to late 1990s. It is named '*next-generation*', because NGS is able to sequence millions of DNA segments in parallel, instead of targeted sequencing on a few DNA molecules. This technology is also called *deep sequencing* given the fact that it provides possibility to sequence the whole transcriptome and genome for a given organism. Based on different sequencing principles, NGS includes the following technologies or platforms: Massively Parallel Signature Sequencing (MPSS), Polony Sequencing, 454 pyrosequencing,

Illumina (Solexa) sequencing, ABI SOLiD sequencing, Ion semiconductor sequencing, DNA nanoball sequencing, etc.

NGS technology includes but not limited to the following applications:

1. *Whole genome sequencing (WGS) or whole exon sequencing (WES)* is to sequence the complete DNA sequence of an organism's genome in parallel, including chromosomal DNA, mitochondria DNA and chloroplast DNA (for plant). The advance of WGS accelerates the study of genomic structure variation (deletion, insertion, inversion, tandem duplication, dispersed duplication, copy-number variation, etc) and single nucleotide polymorphism (SNP: single nucleotide substitution, insertion and deletion) [Ng and Kirkness, 2010; Choi et al., 2009]. Taking the advantage of WGS technology, genome wide association study (GWAS) is able to find genetic variations that contribute to or associate with some complex diseases, such as cancer, asthma and heart disease.
2. *RNA sequencing (RNA-seq)* is a high-throughput transcriptome sequencing technology to determine all or part of the messenger RNA (mRNA), small RNA and non-coding RNA in a given sample. In this technique, RNA molecules are selected by poly-A oligonucleotide, and then reverse transcribed into DNA molecules for sequencing. RNA-seq technology, on one hand, can be applied to detect structure variations at the transcriptome level, for example: fusion transcript detection, alternative splicing, novel isoform detection, etc [Trapnell et al., 2010; Liu et al., 2016]. On the other hand, RNA-seq data can be processed to gene expression data that has similar format as microarray data (Fig. 1.2). Compared to microarray technology, RNA-seq is not limited by probe design, and thus is able to discover novel transcripts (or isoforms).
3. *Whole genome bisulfite sequencing (WGBS) or MethylC-seq* is a technique to sequence bisulfite-treated DNA fragments to determine their methylation patterns. In the library preparation when given bisulfite treatment, only un-methylated cytosines are converted into uracils, while methylated cytosines keep unchanged. In the PCR step, uracils are amplified as thymines. As a result, only methylated cytosines are sequenced as they are, while un-methylated cytosines turn out to be thymines [Urich et al., 2015]. Cytosine methylation can significantly modify spatial gene expression and chromatin structure, and thus associate with many complex diseases. The application of WGBS technique

provides a strong support to locate and quantify the cytosine methylation patterns.

4. *ChIp sequencing (ChIP-seq)* is a method to analyze protein interactions with DNA. In the chromatin immunoprecipitation (ChIP) step, DNA combined with proteins of interest are sonicated into fragments and then immunoprecipitated by the protein-specific antibody. Then in the sequencing step, selected DNA fragments are sequenced by the NGS technology pipeline. As a result, only the DNA regions that can specifically bind to the protein of interest are collected and sequenced. With the help of ChIP-seq technology, protein binding sites can be mapped precisely and globally, and thus accelerate the study of protein-DNA interaction [Park, 2009].
5. *Hi-C sequencing* is a technology to study the chromatin 3D interaction in a high throughput scale. Chromosome conformation capture (3C) method analyzes the spatial interactions between genomic loci. Improved from 3C method that quantifies the interaction between two specific fragments, on the contrary, Hi-C method quantifies all the possible pairwise interaction in the whole chromatin scale. In the Hi-C technique, DNA in the same crosslinked complex is first digested with a restriction enzyme, marked with a biotinylated nucleotide at the end, and then ligated together to form one chimeric DNA molecule. After removing biotin from the end of the DNA pieses, those DNA fragments with internal biotin are selected and sequenced in a high throughput scale. As a result, spatially interactive DNA pairs are sequenced together. After aligning the reads to reference genome, the Hi-C sequencing technology is able to quantify spatial connection between genomic loci in a whole chromatin scale [Belton et al., 2012].

In this dissertation, we majorly focused on analyzing RNA-seq data. Starting from the raw FASTQ file, it usually requires the following pre-processing steps.

1. *Quality control.* Quality control is always the very first step to check the sequencing data quality. Several key points are analyzed to evaluate the quality: per base sequence quality, per sequence quality scores, per sequence GC content, sequence duplication levels, overrepresented sequences, etc [Wang et al., 2012a].
2. *Data trimming.* Based on the sequencing quality, reads are trimmed out majorly by two criteria. (1) Read quality score. For example, a moving window is applied to check along

the reads. If the average quality inside a window is smaller than a certain threshold, then the sequence inside and after this window will be trimmed out, because it's a general trend that the sequencing quality drops as the reads extending longer. (2) Adaptor trimming. If the length of sequencing read is greater than that of DNA fragment, the reads might include adaptor sequence at the tail part. In order not to influence the following alignment analysis, reads need to be trimmed by the adaptor region to increase the alignment rate [Bolger et al., 2014].

3. *Sequence alignment.* Generally there are three types of alignment. (1) Reference-based alignment. Given the organism genome reference, reads are mapped to the position with the highest matching score. Specifically for RNA-seq data, because of alternative splicing, one transcript can consist of more than one non-consecutive regions (exons). In this case, one read might be split into several short pieces for alignment [Kim et al., 2013]. (2) *De novo* assembly. Without any prior knowledge, reads with overlapped regions are formed into contigs, and then contigs will be assembled into scaffols [Baker, 2012]. Compared to reference-based method, *de novo* assembly requires large computing, but on the other hand, it can potentially discover novel transcripts. (3) Mix of the two methods. In order to take the advantage of both pipelines, reads can be first aligned to the organism reference. And then those un-mapped read can build up their own library by the *de novo* assembly. Hybridize of the two pipelines can both increase the alignment efficiency and accuracy, as well as discover novel transcripts.

4. *FPKM value or read count value calculation.* Structure variations at the transcript level can be detected by raw or aligned RNA-seq data directly. However, for gene expression profiling, further processing are needed to calculate the gene abundance. FPKM, defined as fragments per kilo-base per million reads, is normalized by gene length and total number of reads to describe gene expression intensity. Since its data structure is the same as microarray's (Fig. 1.2), many conventional pipeline for microarray analysis (DE, clustering, classification, etc.) can also be applied into FPKM value. Besides, read count for each gene region can be calculated. Instead of continuous expression intensity, element Y_{ij} in the data matrix represents the number of reads aligned to gene i in sample j. In this scenario, algorithms designed for continuous value or based on

normal assumption are no longer suitable for this new data structure. Many new models (for example, poisson or negative binomial models) are developed to simulate the data and for differential expression analysis [Robinson et al., 2010; Love et al., 2014].

1.1.3 Large public database and data depositories

Take the advantage of public genomic databases and data repositories, researchers are able to search and collect multiple types of omic data contributed by different cohorts. Here we listed several public database used in this dissertation:

- The National Center for Biotechnology Information (NCBI)
  https://www.ncbi.nlm.nih.gov/
- Gene Expression Omnibus (GEO)
  https://www.ncbi.nlm.nih.gov/geo/
- The Cancer Genome Atlas (TCGA)
  https://cancergenome.nih.gov/
  University of Pittsburgh mirror: Pittsburgh Genome Resource Repository (PGRR)
  http://pgrr.pitt.edu/
  Broad institute collection: Firebrowse
  http://firebrowse.org/

1.2 INTEGRATIVE ANALYSIS FOR GENOMIC DATA

Motivated by the availability of large public database, the genomic integrative analysis aims to statistically combine multiple types of omics data from different cohorts. Specifically, as illustrated in Fig. 1.3 horizontal meta-analysis refers to the integration of the same type of omics data (for example, gene expression data) among multiple cohorts. Vertical integrative analysis, on the other hand, refers to the study of multiple types of omics data within the same cohort Tseng et al. [2012].

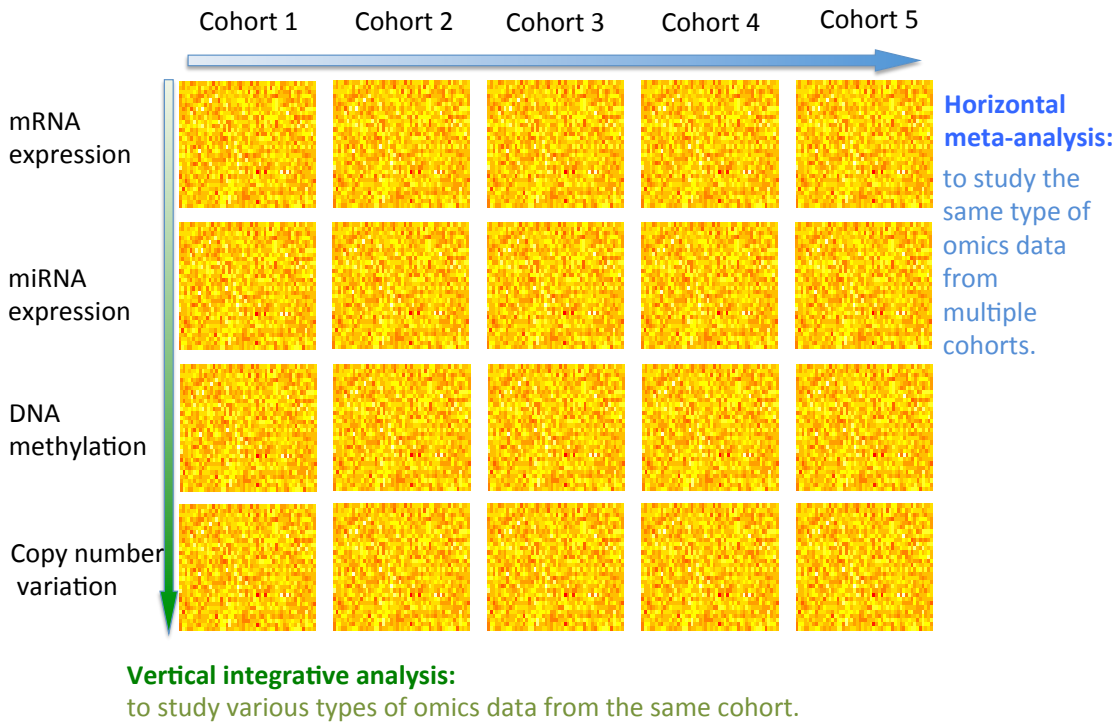


Figure 1.3: Illustration of integrative analysis for different types of omcis data among multiple cohorts.

Compared to single-study analysis, integrative analysis has but not limited to the following advantages:

- Results from single study is lack of power, especially when the sample size is small. However, integrative analysis is able to gain power by enlarging the sample size.
- Combining normalized single-study directly (mega-analysis) will introduce strong batch effect due to the sequencing pipeline, platform and sample bias. Integrative analysis, in contrast, can conquer this problem and take the advantage from each single study.
- Signals detected from single study might be study-specific, and thus can not be generalizable to other cohorts. On the contrary, many integrative analysis methods are able to

detect both consistent and study-specific signals among multiple studies.

There are typically two categories of information integration: combining effect sizes and combining p-values.

When combining effect sizes, fixed effects and random effects models are widely used. On the one hand, fixed effect model is defined as $T_k = \theta + \epsilon_k$ with $\epsilon_k \sim \mathcal{N}(0, \sigma_k^2)$, where T_k is the observed effect size of study k, θ is the mean effect size, and ϵ_k is the sampling error for study k. On the other hand, random effects model is defined as $T_k = \theta + \epsilon_k + \zeta_k$ with $\epsilon_k \sim \mathcal{N}(0, \sigma_k^2)$ and $\zeta_k \sim \mathcal{N}(0, \tau^2)$, where ζ_k is an additional term for underlying population variance.

When combining p-values to calculate a meta p-value, many evidence aggregation and order-statistic methods have been developed. For example, Fisher's method, Stouffer's method, Logit method, minP, maxP, rth ordered p-value, etc [Chang et al., 2013]. In order to differentiate study-specific and consistent signals, methods like adaptively weighted Fisher's method are developed to assign binary weight to each study [Li et al., 2011a].

However, many complex machine learning methods cannot be easily extended into meta-analysis by simply combining effect sizes or p-values, and thus many new meta-analytic models are required to fill in the blank. For example, Kang et al. [2012] developed a meta quality control pipeline to quantify multiple data qualities by six data-data similarity measurements. Shen and Tseng [2010] investigated two approaches of meta-analysis for pathway enrichment (MAPE) by combining statistical significance across studies at the gene level (MAPE_G) or at the pathway level (MAPE_P). Huo et al. [2016] propose a meta sparse k-means algorithm for disease subtype detection and matching among multiple cohorts.

In this dissertation, we include one pipeline for the integration of fusion transcript detection tools, and two horizontal meta-analysis pipelines for liquid association and bicluster discovery. We will give a brief review of each concept in the following introduction sections.

1.3 BRIEF REVIEW OF FUSION TRANSCRIPT

Fusion transcript is a chimeric RNA encoded by a fusion gene at the DNA level or formed by trans-splicing of two different genes at the RNA level. At the DNA level, fusion gene is defined as a chimeric gene that formed from two previously separate genes. It can occur as a result of chromosomal translocation, deletion or inversion. At the transcript level, for trans-splicing, exons from two different primary RNA transcripts are fused together by RNA processing.

1.3.1 Classification of fusion transcript

In general, fusion transcripts in prostate cancer can be classified into three categories on the basis of the protein structure of the head gene and the tail gene in the fusion transcripts [Luo et al., 2015]:

1. *Chimera protein forming fusion transcript.* Translation of fusion protein occurs at the ATG start codon of the head gene, and connect it in frame with the C-terminus of the tail gene.
2. *Independent wild type tail gene fusion transcript.* Fusion point occurs at the 5' untranslated region of the mRNA such that both ribosomal binding site and ATG start codon are preserved. The consequence of such fusion is independent translation of the tail gene in the transcript, while the expression of the tail gene is driven by head gene promoter. Head gene may or may not express a truncated protein.
3. *Non-fusion forming fusion transcript.* The ATG translation start codon and the ribosome binding site of the tail gene are deleted, while the head gene has C-terminus truncation or has lost the open reading frame all together. In rare occasion, a wild type head gene is preserved in non-fusion forming fusion transcript.

1.3.2 Application of fusion transcript

The discovery of fusion transcripts has many clinical applications, especially in tumorigenesis and cancer progression. Fusion transcripts may create a chimeric protein with a new or

altered activity. Alternatively, it can contribute to the over-expression of a seemingly normal-expressed gene, or down-regulation of a tumor suppressor gene.

Here we summarize two applications of fusion transcript.

First of all, fusion transcript can behave as biomarker to predict cancer clinical status such as, relapse versus non-relapse, or fast relapse versus non-fast relapse. For example, Yu et al. [2014] detected 8 novel fusion transcripts from prostate cancer. Among 179 samples from UPMC cohort, samples with and without fusion detected showed out significantly different survival trend by Kaplan-Meier analysis. Two conventional indicators – Gleason score and Nomogram – were traditionally used for the prediction of prostate cancer status. However, the performance will be significantly improved by adding the existence status of fusion transcripts as prediction attributes. As another example, Yan et al. [2015] proposed that the ratio of large size copy number variations was able to predict the prostate cancer relapse or non-relapse status. With the addition of fusion transcripts, the prediction results were improved significantly.

Secondly, fusion transcripts (or fusion gene) play roles as therapeutics target for some diseases. For example, Cools et al. [2003] found that fusion FIP1L1-PDGFRα might be the cause of hypereosinophilic syndrome by forming a novel fusion tyrosine kinase via an interstitial chromosomal deletion process. The study showed that FIP1L1-PDGERα is the target of imatinib and the deletion of genetic material may result in gain-of-function fusion protein. Take the research by Chen et al. [2017] as another example. MAN2A1-FER is a fusion gene between the mannosidase domain of MAN2A1 and tyrosine kinase domain of FER. Expression of MAN2A1-FER protein generated dramatic increase of growth and invasion of cancers in vitro and in vivo, while removal of the fusion through knockout generated significant lower level of growth and metastasis.

1.3.3 Detection or validation methods of fusion transcript

Because of the wide application of fusion transcripts, it is important to test the fusion events for a given sample and to discover novel fusion transcripts from new samples. Generally, there are two experimental validation methods.

- *RT-PCR + Sanger sequencing.* These techniques can quantify the DNA molecules and get the exact sequence for a given primer-designed region. As shown in Fig. 1.4, if the sequenced region consists of the sequences from two separate genes, then it is a strong proof that these two genes are fused together.

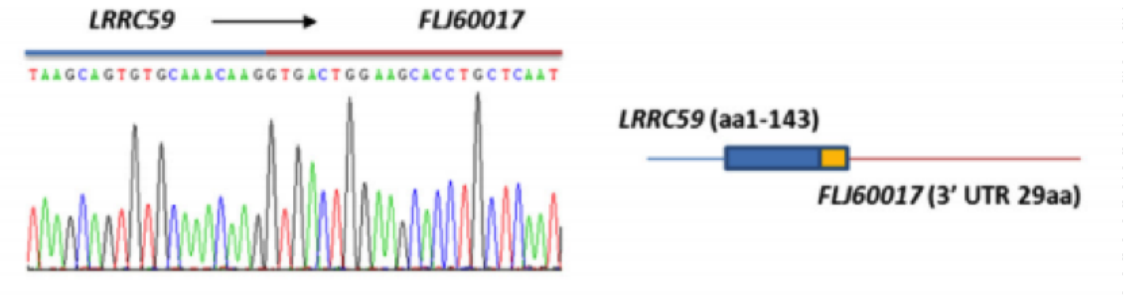


Figure 1.4: Illustration of fusion transcripts validated by RT-PCR + Sanger sequence method.

- *Fluorescence in situ hybridization (FISH).* As illustrate in Fig. 1.5, FISH is able to specifically insert fluorescence markers to two candidate genes. Then under the microscope, if fluorescence spots from two genes pair together, it will strongly support the fusion event.

However, the methods introduced above have their limitations. Both methods can only validate the existence of known (or suspicious) fusions, but are not able to discover novel fusions in a whole-genome scale. For example, RT-PCR + Sanger sequencing can only target on a certain region. It requires the prior knowledge of two candidate genes in order to design their primer sequences. But this technique fails to test all the possible gene combinations in a high throughput scale. Similar for FISH method, scientists are only able to insert markers to candiate gene pairs for testing. Nevertheless, it is impossible to mark all the genes to test all the potential pairwise combinations.

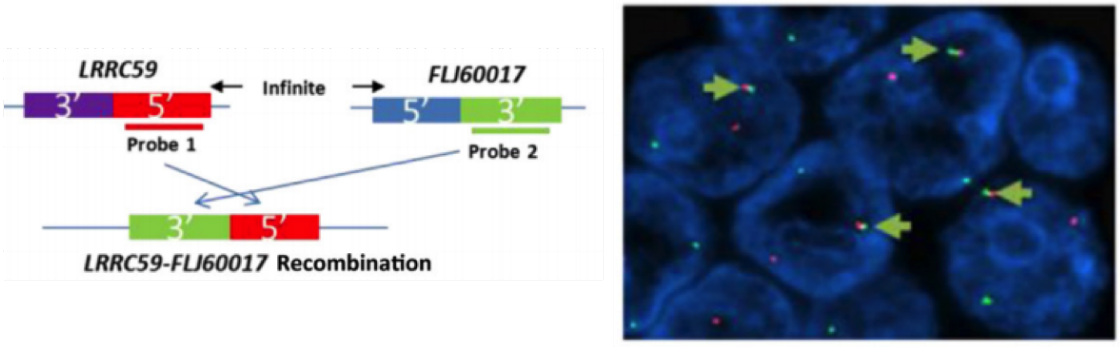


Figure 1.5: Illustration of fusion transcripts validated by fluorescence in situ hybridization (FISH) method.

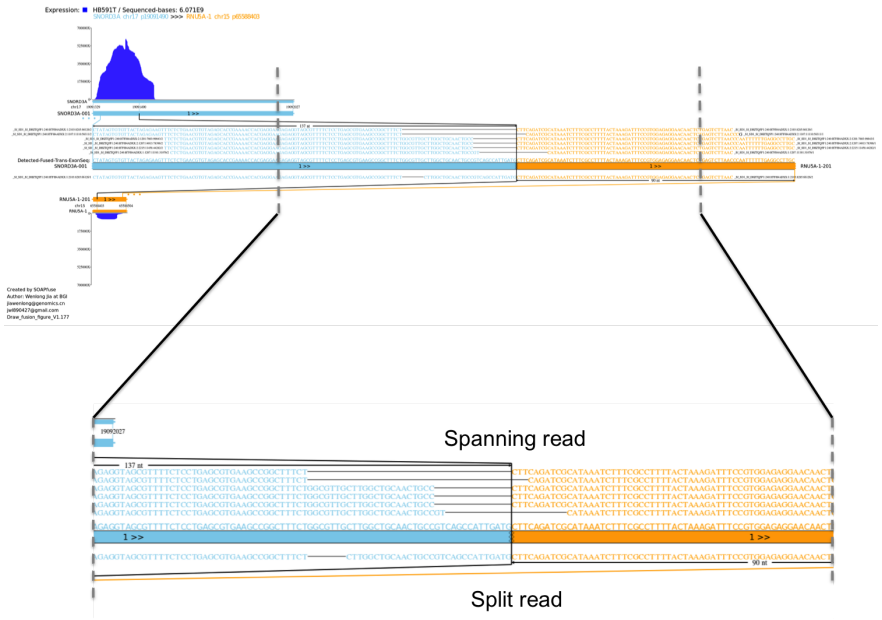


Figure 1.6: Illustration of fusion transcripts detected by RNA-seq.

Those limitations motivate the idea of introducing high throughput sequencing method into fusion transcripts detection. WGS and RNA-seq are two techniques to discover fusion events at DNA and RNA level, respectively. Take the RNA-seq detecting fusion transcripts as an example. Fig. 1.6 shows an illustration drawn by SOAPfuse software [Jia et al., 2013] where reads from RNA-seq support the fusion of two genes marked by blue and orange colors. Generally, there are two kinds of supporting reads. On one hand, *spanning read* is defined as the paired-end read where one end is aligned to blue gene and the other end is aligned to orange gene. *Split read*, on the other hand, exactly covers the break point and splits into two parts that can be aligned to two genes separately. Similar to RNA-seq, WGS detects fusion genes at the DNA level. In addition, it is able to locate the break point inside the intron region by the support of split reads.

Many fusion transcripts detection tools have been developed for paired-end RNA-seq data. In the first project of this dissertation, we comprehensively evaluated 15 pipelines in terms of precision, recall, accuracy, number of supporting reads and computing cost. We then proposed a meta-caller to combine several top tools to improve the balance between precision and recall rate.

1.4 COMPLEX GENE REGULATION PATTERNS

1.4.1 Gene regulation and co-expression

Gene expression is the process of synthesizing multiple functional gene products. For example, coding genes generate proteins, which are the most commonly used functional units to construct the organism. As illustrated in Fig. 1.1, information flows from gene to RNA and then to protein by the following process: transcription, RNA splicing, translation, and post-translational modification of a protein. Alternatively, non-protein coding genes produce functional RNA, such as transfer RNA (tRNA) or small nuclear RNA (snRNA).

Regulation of gene expression refers to the mechanisms that are used to control the amount and timing of functional gene products (protein or RNA). These mechanisms include

the regulation at each step of information flow: transcriptional initiation, RNA processing and post-translational modification of a protein. Gene regulation is the basis for cellular proliferation, differentiation, morphogenesis, apoptosis, and adaptation to various external signals, such as environmental changes, new food sources and other stimuli.

Exploring the gene regulation pattern plays an important role to understand complex biological process, and thus reveals the mechanisms of pathogenesis and tumorigenesis. The study of gene regulation can trace back to 1961 where Jacques Monod identified lac operon, in which the expression of some lactose-involved enzymes in E. coli is triggered by the presence of lactose and absence of glucose. As for now, taking the advantage of high throughput technology, gene regulation can be learnt in a whole-genome scale. That is to say, instead of focusing on a few individual genes, scientists are able to study the genome-wide expression pattern of a sample under certain conditions, such as drug treatment, physical or chemical stimulus, cell cycle, etc.

The relationships of gene expression can be represented in the form of gene network, where each node represents a gene and each edge indicates the interaction [D'haeseleer et al., 2000; De Jong, 2002]. In this scenario, network graph can be divided into two subtypes. *Gene regulatory network* can be translated into a *directed* graph, where an arrow from molecule A to B represents the regulation of A to B. Alternatively, *Gene co-expression network* is an *undirected* graph, where an undirected edge connecting two genes means the co-expression relationship between them.

It takes two steps to construct a gene co-expression network. First, gene distances are calculated between all pairwise genes. The co-expression relationship can be measured by Pearson's correlation coefficient, Mutual Information, Spearman's rank correlation coefficient or Euclidean distance. In the second step, genes are connected only if their co-expression measure is greater than a certain threshold, or the p-value of the measure can reach a defined significance cutoff. For example, an edge will connect two genes as long as their absolute value of Pearson's correlation is greater than 0.7, otherwise no connection.

However, the construction procedure above only considers the pairwise gene association. As a matter of fact, the actual biological regulation patterns are more complex than this simple model. Motivated by this limitation, Li [2002] proposed the concept of liquid association

that will be introduced below.

1.4.2 Liquid association

Liquid association (LA), proposed by Li [2002], is the study of three-way gene regulation. In contrast to pairwise gene association (solid association), LA measures the dynamic gene-gene association given a third scouting gene. For example, as illustrated in Fig. 1.7, gene X and Y are highly positively correlated when the expression level of gene Z is high, while they are highly negatively correlated when the expression of gene Z is low. This kind of dynamic pattern, however, cannot be detected by the traditional pairwise association because the reverse correlations under two conditions may neutralize the overall correlation.

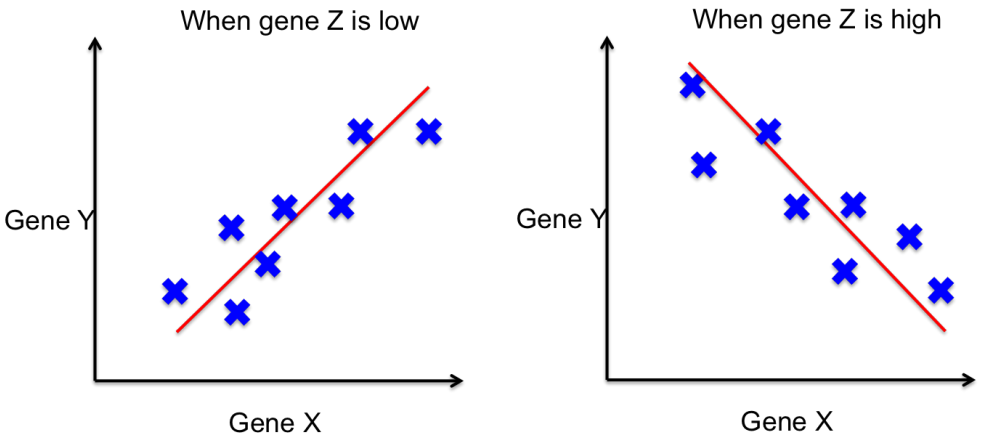


Figure 1.7: Illustration of liquid association between gene X and Y given a third scouting gene Z.

Take the urea-cycle genes as an example to illustrate liquid association [Li, 2002]. Many genes involved in urea cycle are predicted to or physically correlate with each other, however their pairwise Pearson correlations are not significant. Detected by LA method, gene SCH9 behaved as a scouting gene that can regulate many gene pairs correlation. When SCH9 was low, gene pairs (ARG2, ARG1), (ARG2, ARG3), (ARG2, ARG4) and (ARG2, CAR1) were all positively co-expressed. In contrast, when SCH9 was high, all these four gene pairs

were negatively or not correlated. As another example, Li et al. [2004] studied the Stanford cell cycle data and found that the pairwise correlation of gene (GCD11, SUI3), (GCD11, SUI2), and (SUI2, SUI3) were low. However, scouting genes RPL11B, RPL10, DBP10, IFH1 and DBP10 were detected which can dynamically regulate the association of the above gene pairs. In conclusion, liquid association is able to quantify three-way gene regulation, and even predict novel gene regulation or co-expression patterns.

To quantify liquid association, Li [2002]; Li et al. [2004, 2007] estimated the LA score to be $LA(X, Y|Z) = E(XYZ)$ when standard normalized the data. Ho et al. [2011] improved the estimation framework by proposing the modified liquid association (MLA) method. In order to search the gene triplets in whole genome scale, Gunderson and Ho [2014] developed a fast algorithm to pre-filtered those non-significant triplets to reduce the computing cost.

In the second project of this dissertation, we further extended the algorithm into meta-analytic framework to take the advantage of multiple cohorts.

1.5 BICLUSTERING OF GENE EXPRESSION DATA

Clustering is an unsupervised machine learning method to cluster similar objects into the same group. As we introduced in the above section, the gene expression matrix has the format as it is shown in Fig. 1.2, where row represents gene and column represents sample. In general, there are two clustering directions for gene expression data. For one direction in Fig. 1.8A, genes with similar expression patterns are clustered together. In this case, genes within one cluster are expected to be enriched in some similar biological pathways. For the other direction in Fig. 1.8B, similar samples with the same disease subtypes are grouped together. In order to cluster genes or samples, many machine learning clustering methods have been developed, such as k-means [Hartigan and Wong, 1979], hierarchical clustering [Eisen et al., 1998] and model based approaches [McLachlan et al., 2002].

However, there are several limitations for one-way clustering in terms of genes and samples selection. On one hand, a disease subtype is only correlated with the expressions of a

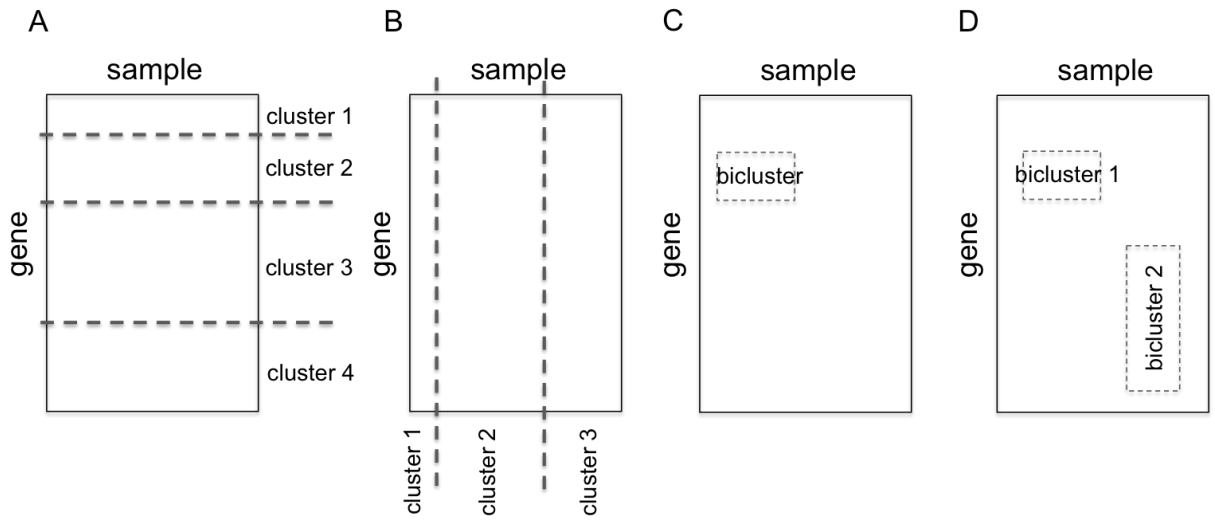


Figure 1.8: Illustration of Clustering and Biclustering on gene expression data.

subset of genes, instead of whole list of genes. So it will weaken the signal if samples are clustered using all the gene attributes. On the other hand, some genes can regulate multiple disease subtypes while the others may only correspond to one or no disease subtype.

These limitations motivate the idea of *biclustering*. That is, the algorithm aims to detect a subset of genes and a subset of samples simultaneously. As illustrated in Fig. 1.8C, a submatrix from the original gene expression matrix is selected. Not limited to one bicluster, usually multiple biclusters with different expression patterns can be detected (Fig. 1.8D).

1.5.1 Types of biclusters

Mathematically, we defined Y_{ij} to be the expression intensity of gene i in sample j. Based on different expression pattern models, Freitas et al. [2012] categorized the biclusters into 5 types.

1. *Bicluster with constant expression values.* Mathematically, $Y_{ij} = \mu$, where μ is the constant mean expression value.
2. *Bicluster with constant values on rows or columns.* For bicluster with constant values on

rows, it can be written as $Y_{ij} = \mu + \alpha_i$ or $Y_{ij} = c \times \alpha_i$, where α_i is the row effect for gene i, and c is a constant parameter. Similarly, bicluster with constant values on columns can be written as $Y_{ij} = \mu + \beta_j$ or $Y_{ij} = c \times \beta_j$, where β_j is the columns effect for sample j.

3. *Bicluster with coherent values.* To combine the previous equations, bicluster element can be defined as $Y_{ij} = \mu + \alpha_i + \beta_j$, or $Y_{ij} = c \times \alpha_i \times \beta_j$.
4. *Bicluster with linear coherent values.* Bicluster can be obtained by the linear format as $Y_{ij} = c \times \alpha_i + \beta_j$.
5. *Bicluster with coherent evolution.* That is , bicluster rows (or columns) include a linear order across a subset of columns (or rows).

1.5.2 Biclustering algorithms

The concept of biclustering was first introduced by Hartigan [1972] and then applied into gene expression data by Cheng and Church [2000]. Many biclustering algorithms have been developed to detect different bicluster patterns [Ruffalo et al., 2011; Eren et al., 2013; Madeira and Oliveira, 2004]. Here we summarize several algorithms that have been widely used.

- *Cheng and Church.* Cheng and Church [2000] first applied biclustering into gene expression matrix. Defined a_{ij} as the data element for row i and column j, I and J are row and column sets of the bicluster. This algorithm aims to detect biclusters that can minimize the mean squared residual (MSR). Mathematically, $MSR = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$, where $a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$, $a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$ and $a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij}$.
- *Conserved gene expression motifs (xMOTIFs).* Murali and Kasif [2003] aims to detect biclusters with consistent row expression. Each row of the data matrix is first discretized into several status. Then a seed is defined as a randomly selected column, and a discriminating set is defined as randomly selected set of columns. xMOTIFs tries to detect rows that have same states over the columns of the seed and the discriminating set.
- *Correlated pattern biclusters (CPB).* Bozdağ et al. [2009] proposed this algorithm to detect biclsuters with high row-wise correlation measured by Pearson correlation coefficient. As an initiation, CPB randomly selects a row and some random columns. Then

it iteratively adds rows that have high correlation with the seed row, and columns that have smaller root mean squared error (RMSE).

- *Plaid model.* Lazzeroni and Owen [2002] assumes the data matrix expression value is the superposition of multiple layers (biclsuters). Mathematically, the expression element can be expressed as $Y_{ij} = \theta_0 + \sum_{k=1}^{K} \theta_k \rho_{ik} \kappa_{jk} + \epsilon_{ij} = (\mu_0 + \alpha_{i0} + \beta_{j0}) + \sum_{k=1}^{K} (\mu_k + \alpha_{ik} + \beta_{jk}) \rho_{ik} \kappa_{jk} + \epsilon_{ij}$, where θ is the overall expression value that can be defined as the main (μ), gene (α) and sample (β) effect, ρ_{ik} and κ_{jk} are gene i and sample j indicator for layer k with 1 meaning selected and zero otherwise. More details will be introduced in the third project.
- *Bayesian biclustering (BBC).* This algorithm extends the plaid model into a hierarchical Bayesian analysis by Gibbs sampling [Gu and Liu, 2008]. However, it only allows the biclusters overlapping in either gene or sample directions. Many other Bayesian methods have been proposed for different models [Caldas and Kaski, 2008; Zhang, 2010].
- *Spectral biclustering.* In a cancer context, checkerboard patterns are defined as genes that are markedly up- or down-regulated in patients with particular types of tumors. Kluger et al. [2003] proposed the spectral algorithm to detect this kind of checkerboard pattern by eigenvectors corresponding to characteristic expression patterns across genes or conditions. As a result, only biclusters with low variance are detected when applying this method.
- *Factor analysis for bicluster acquisition (FABIA).* This model assumes the expression data matrix X to be the sum of p biclsuters and noise γ [Hochreiter et al., 2010]. Mathematically, $X = \sum_{i=1}^{p} \lambda_i z_i^T + \gamma = \Lambda Z + \gamma$, where each bicluster is the outer product of two sparse vectors: row vector λ and column vector z. Some other methods using factor analysis were also developed [Martella et al., 2008].

1.5.3 Evaluation of biclusters

Detected biclusters can be evaluated in two scenarios. On one hand, for synthetic data with the underlying truth, many measurements have been developed to quantify the similarity between two clusters [Horta and Campello, 2014]. Except for those complex evaluation

methods, here we introduced several direct and widely used ones.

Fig. 1.9 shows the overlapping between predicted and true biclusters. True positives (TPs) are defined as the overlapped elements between the two clusters, false positives (FPs) are the elements only detected by predicted bicluster but not by true biclsuters, (FNs) are the elements only in true but not in predicted bicluster, and true negatives (TNs) are those elements outside both clusters. Based on these concepts, three evaluation methods are defined as follows,

$$\begin{aligned} \text{Jaccard} &= \frac{TP}{FP+TP+FN}; \\ \text{Sensitivity} &= \frac{TP}{TP+FN}; \\ \text{Specificity} &= \frac{TN}{FP+TN}. \end{aligned}$$

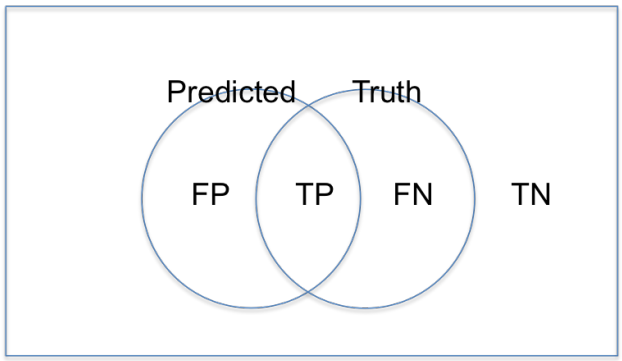


Figure 1.9: Overlapping between predicted bicluster and true bicluster elements.

On the other hand, for real data biclustering without known truth, detected genes and samples are checked respectively to evaluate the performance.

Selected bicluster genes are usually checked by their enrichment of some collected gene sets. A biological pathway describes a series of molecules that are related to a certain biological functions. Genes are conventionally grouped by their correlated biological pathways or physical interactions. Thus, bicluster genes are expected to significantly enriched with some

of these gene sets so that they can be evaluated as functionally related gene sets. Fisher's exact test is used to test the association between biclsuter gene set and some conventionally defined gene pathways [Upton, 1992].

Biclustring is expected to split the samples into different disease subtypes or treatment conditions. With the patients' prior knowledge, association between sample splitting (samples selected or not selected by the biclsuter) and their clinical information. Alternatively, Kaplan-Meier analysis can be used to evaluate the sample splitting in terms of survival behaviors.

Motivated by the high throughput genomic data and the integrative analysis, in the third project, we extended the plaid model for single-study analysis into a meta-analytic framework. In the third project, we will introduce more details of the plaid model and how can its objective function be improved for multiple-cohorts biclustering.

1.6 MAIN CONTRIBUTIONS AND SPECIFIC AIMS

In this dissertation, we are taking the advantage of multiple high-throughput genomics data to extend the fusion transcript detection, liquid association and biclustering into integrative analysis framework. Here are the main contributions of this dissertation,

- We evaluated 15 fusion transcript detection tools in paired-end RNA-seq data, and developed a meta-caller to re-prioritize the fusion transcript candidates by combining the results from top performing tools.
- We explored more robust gene-triplet liquid association among multiple transcriptomic cohorts.
- We extended the plaid model into a meta-analysis framework to detect biclusters from multiple cohorts with consistent gene selection.

Below we outline the three projects in this dissertation.

1.6.1 Aim 1. Comprehensive evaluation of fusion transcript detection algorithms and a meta-caller to combine top performing methods in paired-end RNA-seq data

Background: Fusion transcripts are formed by either fusion genes (DNA level) or trans-splicing events (RNA level). They have been recognized as a promising tool for diagnosing, subtyping and treating cancers. RNA-seq has become a precise and efficient standard for genome-wide screening of such aberration events. Many fusion transcript detection algorithms have been developed for paired-end RNA-seq data but their performance has not been comprehensively evaluated to guide practitioners. In this paper, we evaluated fifteen popular algorithms by their precision and recall trade-off, accuracy of supporting reads, and computational cost. We further combine top-performing methods for improved ensemble detection.

Results: Fifteen fusion transcript detection tools were compared using three synthetic data sets under different coverage, read length, insert size and background noise, and three real datasets with selected experimental validations. No single method dominantly performed the best but SOAPfuse generally performed well, followed by FusionCatcher and JAFFA. We further demonstrated the potential of a meta-caller algorithm by combining top performing methods to re-prioritize candidate fusion transcripts with high confidence that can be followed by experimental validation.

Conclusions: Our result provides insightful recommendations when applying individual tool or combining top performers to identify fusion transcript candidates.

1.6.2 Aim 2. Meta-analytic framework for liquid association

Motivation: Although coexpression analysis via pair-wise expression correlation is popularly used to elucidate gene-gene interactions at the whole-genome scale, many complicated multi-gene regulations require more advanced detection methods. Liquid association is a powerful tool to detect the dynamic correlation of two gene variables depending on the expression level of a third variable (LA scouting gene). Liquid association detection from single transcriptomic study, however, is often unstable and not generalizable due to cohort

bias, biological variation, and limited sample size. With the rapid development of microarray and NGS technology, liquid association analysis combining multiple gene expression studies can provide more accurate and stable results.

Results: In this paper, we proposed two meta-analytic approaches for liquid association analysis (MetaLA and MetaMLA) to combine multiple transcriptomic studies. To compensate demanding computing, we also proposed a two-step fast screening algorithm for more efficient genome-wide screening: bootstrap filtering and sign filtering. We applied the methods to five *Saccharomyces cerevisiae* data sets related to environmental changes. The fast screening algorithm reduced 98% of running time. Compared with single study analysis, MetaLA and MetaMLA provided stronger detection signal and more consistent and stable results. The top triplets are highly enriched in fundamental biological processes related to environmental changes. Our method can help biologists understand underlying regulatory mechanisms under different environmental exposure or disease states.

Availability: : A *MetaLA* R package, data and code for this paper are available at http://tsenglab.biostat.pitt.edu/software.htm.

1.6.3 Aim 3: Meta-analytic plaid model for detecting biclusters when combining multiple transcriptomic studies

Motivation: When analyzing transcriptomic data, clustering genes can identify gene modules with highly correlated patterns across all samples, where the co-expressed genes are likely co-regulated or share common biological functions. To account for patient heterogeneity, biclustering methods can detect gene modules correlated in a subset of samples. With increasing number of gene expression profiles accumulated in public databases, combining multiple transcriptomic studies by meta-analytic approaches not only improves statistical power but also provides more consistent results. This motivates the meta-biclustering method proposed in this paper.

Results: We developed a biclustering plaid model towards meta-analytic framework for integrating multiple transcriptomic studies. Gap statistic was introduced to determine tuning parameters in the algorithm. Using extensive simulations, we showed that the new

meta-biclustering method generated more accurate and robust clustering results. Bicluster genes selected by training cohorts are generalizable to testing cohorts. The method was further applied to five breast cancer expression profiles. Identified bicluster genes were highly enriched in previously characterized breast cancer related pathways. The corresponding bicluster samples were significantly associated with ER status and survival behavior. These expression signatures form basis to characterize disease subtypes for possible personalized medicine.

Availability: : A *MetaBiclust* R package, data and code for this paper are available at http://tsenglab.biostat.pitt.edu/software.htm and GitHub.

2.0 AIM 1. COMPREHENSIVE EVALUATION OF FUSION TRANSCRIPT DETECTION ALGORITHMS AND A META-CALLER TO COMBINE TOP PERFORMING METHODS IN PAIRED-END RNA-SEQ DATA

This is a pre-copyedited, author-produced version of an article accepted for publication in *Nucleic Acid Research* following peer review. The version of record [Liu et al., 2016] is available online at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4797269/.

2.1 INTRODUCTION

Fusion gene is a result of chromosomal insertion, deletion, translocation or inversion that joins two otherwise separated genes. Fusion genes are often oncogenes that play an important role in the development of many cancers. Trans-splicing is an event that two different primary RNA transcripts are ligated together. Both fusion genes (DNA level) and trans-splicing events (RNA level) can form fusion transcripts. These events usually come from different types of aberrations in post-transcription and chromosomal rearrangements: large segment deletion (e.g. the well-known fusion *TMPRSS2-ERG* in prostate cancer [Tomlins et al., 2005]), chromosome translocation (e.g. the well-known fusion *BCR-ABL1* in chronic myeloid leukemia [Barnes and Melo, 2002] and *EML4-ALK* in non-small-cell lung cancer [Soda et al., 2007]), trans-splicing [Gingeras, 2009] or readthrough (two adjacent genes) [Kaye, 2009]. To date, many fusion transcripts have been found and collected in public databases. For example, there are 10,890 fusions in COSMIC (release 72) [Forbes et al., 2015], 1,374 fusion sequences found in human tumors (involving 431 different genes) in TICdb (release 3.3) [Novo et al., 2007], 2,327 gene fusions in the Mitelman database (updated on Feb 2015) [Mitelman

et al., 2015], and 29,159 chimeric transcripts in ChiTaRS (version 2.1) [Frenkel-Morgenstern et al., 2013, 2015]. Some databases (such as COSMIC, TICdb and ChiTaRS) collected fusion gene sequences and some (e.g. COSMIC and ChiTaRS) offered further summaries of the original tissue types.

The advances in Massively Parallel Sequencing (MPS) have enabled sequencing of hundreds of millions of short reads and have been routinely applied to genomic and transcriptomic studies. The per-base sequencing resolution has provided a precise and efficient standard for fusion transcript detection, especially using paired-end RNA-Seq platforms [Maher et al., 2009]. For example, Berger et al. detected and verified 11 fusion transcripts in melanoma samples, and also identified 12 novel chimeric readthrough transcripts [Berger et al., 2010]. McPherson et al. verified 45 out of 268 detected fusion transcripts in ovarian and sarcoma samples [McPherson et al., 2011a]. Kangaspeska et al. detected and verified 13 fusion transcripts in breast cancer cell lines [Kangaspeska et al., 2012]. Sakarya et al. detected and verified another 25 fusion transcripts in breast cancer cell lines [Sakarya et al., 2012]. Furthermore, Chen et al. proposed a method, BreakTrans, which combined RNA-Seq and whole genome sequencing data of breast cancer samples to detect fusion transcripts [Chen et al., 2013]. Since 2010, many computational tools have been developed for detecting fusion transcripts using RNA-Seq data (see a comprehensive list of 23 methods in Table A.1). Wang et al. [Wang et al., 2013], Carrara et al. [Carrara et al., 2013] and Beccuti et al. [Beccuti et al., 2013] provided insightful reviews of these pipelines. Beccuti et al. developed an R package Chimera that can organize and analyze fusion transcripts detected by multiple tools [Beccuti et al., 2014].

Fig. 2.1A shows two common types of fusion transcripts: intact exon (IE) type and broken exon (BE) type. For IE-type, the rearrangements generally occur in intronic regions and the transcript break point locates exactly at the boundary of the exon, while for BE-type the break point can be in the middle of an exon. To detect these fusion transcripts, paired-end reads are powerful to generate *spanning reads*, with one read aligned to gene A and the other paired read aligned to gene B (see left plot of Fig. 2.1B). Alternatively, a read can be partially aligned to gene A and partially to gene B (see right plot of Fig. 2.1B). This kind of supporting reads are called *split reads* and are useful to define the exact transcript

break point of the fusion transcript. The length of the partial alignment to each fused gene is called *anchor length*. We usually require a minimal threshold of anchor length (e.g. 10 bp) otherwise false positives will increase due to ambiguous multiple alignments of the short partial reads.

Despite rapid development of many computational tools, their respective performance has rarely been evaluated systematically. Carrara et al. compared eight fusion transcript detection tools mostly published in or before 2011 [Carrara et al., 2013]. The evaluation used a small scale of simulated datasets and two real datasets, and the comparison considered sensitivity without proper false positive control, causing inconsistent conclusions and failing to provide a useful application guideline. Developers of recently proposed tools, such as SOAPfuse [Jia et al., 2013], FusionQ [Liu et al., 2013] and JAFFA [Davidson et al., 2015], provided similar small-scale comparative study but the evaluations are all minimal and not conclusive. Many obstacles have hindered the generation of a comprehensive and insightful evaluation, including numerous intermediate steps and parameters that may impact the result in each pipeline, difficulties of proper installation of many tools, frequent updates of software versions and lack of convincing benchmarks for evaluation.

In this paper, we aim to perform a comprehensive evaluation of up to 15 fusion transcript detection tools (Table A.2), to provide a conclusive application guideline and to explore an improved ensemble detection algorithm by combining multiple top-performing methods. We applied three synthetic data sets under different coverages, read lengths and background noises (Table A.3, A.4, A.5 and A.6) with 150 designed underlying true fusions (80 IE-type and 70 BE-type) and also evaluated the tools in three real datasets with experimental validations (Table A.7). We evaluated using three criteria: precision-recall plot (for both synthetic and real data), accuracy of supporting reads (for synthetic data only) and computation cost (for one synthetic and one real dataset). The results will provide researchers and practitioners with insightful recommendations when using these pipelines. Among the 15 evaluated tools, no single method dominantly performed the best for all data. We further explored an ensemble (or meta-caller) algorithm by combining three top-performing algorithms (SOAPfuse, FusionCatcher and JAFFA) to improve recall rate while maintain high

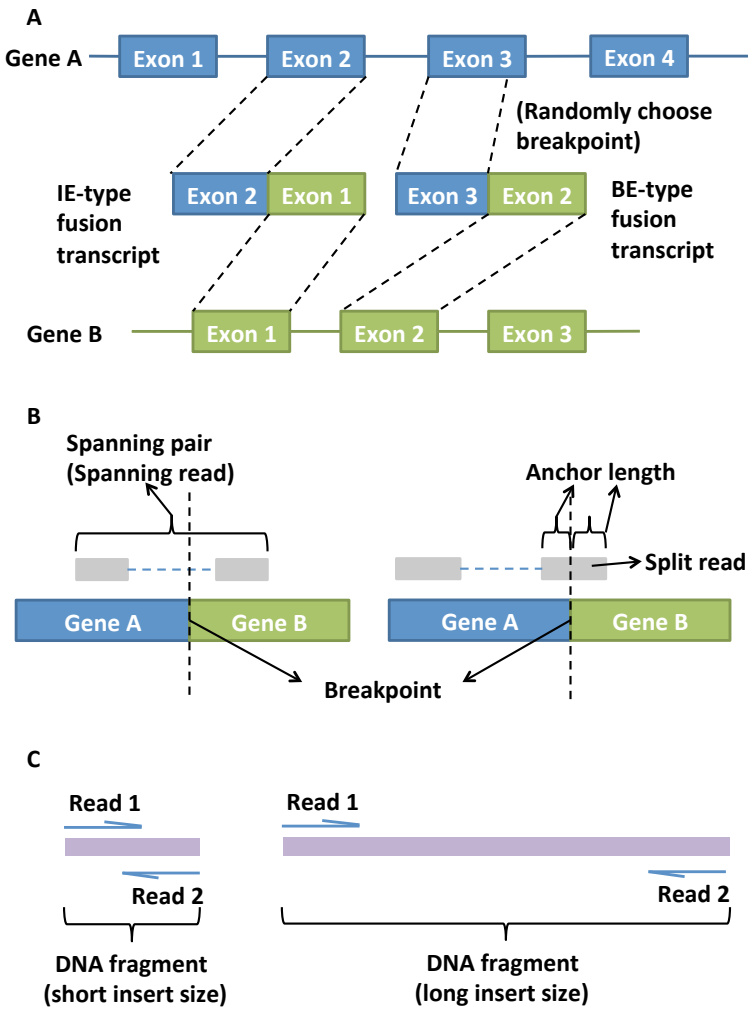


Figure 2.1: Figures to explain terminology. (A)IE-type and BE-type fusion transcripts; (B) spanning read, split read and anchor length; (C) short and long insert size of DNA fragment for sequencing.

precision. Result of the meta-caller was desirable to detect more candidate fusion transcripts with high confidence.

R package *FusionMetaCaller* is available on our website http://tsenglab.biostat.pitt.edu/software.htm.

2.2 MATERIALS AND METHODS

2.2.1 Overview of fusion transcript detection tools

To the best of our knowledge, we summarized 23 state-of-the-art fusion transcript detection tools in Table A.1, among which, 15 tools were examined in this study (Table A.2): MapSplice [Wang et al., 2010], ShortFuse [Kinsella et al., 2011], FusionHunter [Li et al., 2011b], FusionMap [Ge et al., 2011], deFuse [McPherson et al., 2011a], chimerascan [Iyer et al., 2011], FusionCatcher [Edgren et al., 2011; Nicorici et al., 2014], TopHat-Fusion [Kim and Salzberg, 2011], BreakFusion [Chen et al., 2012], EricScript [Benelli et al., 2012], SOAPfuse [Jia et al., 2013], FusionQ [Liu et al., 2013] , SnowShoes-FTD [Asmann et al., 2011], PRADA [Torres-García et al., 2014] and JAFFA [Davidson et al., 2015]. These detection tools differ in a variety of aspects, including read alignment methods [Li and Homer, 2010], criterion for determining fusions, advanced filtering criteria and final output information. In read alignment, for example, many tools (such as TopHat-Fusion, chimerascan, deFuse, FusionCatcher, FusionQ and SnowShoes-FTD) align all reads to the reference sequence using Bowtie [Langmead et al., 2009] or Bowtie2 [Langmead and Salzberg, 2012]. Other alignment tools such as EricScript, BreakFusion and PRADA use BWA [Li and Durbin, 2009], SOAPfuse uses SOAP2 [Li et al., 2009] and FusionMap has its own alignment algorithm. SOAPfuse, chimerascan, deFuse, EricScript, FusionCatcher, BreakFusion, PRADA and JAFFA use more than one alignment tool (combine with BLAT [Kent, 2012], STAR [Dobin et al., 2013] or BLAST [Camacho et al., 2009]) to increase the accuracy of alignment and fusion break point detection. In addition, some detection tools include assembly tools to construct new references with the alignment results. FusionQ, BreakFusion and FusionCatcher use cufflinks [Trapnell et al., 2010], TIGRA-SV [Mills et al., 2011] and velvet [Zerbino and Birney, 2008] respectively to improve the true positive rate with the expense of more computing times and memories. In our implementation, we adopted the most recent versions in May 2015 and used the default alignment settings in each of the 15 pipelines to have fair comparison (except that we fine-tuned the parameters of TopHat-Fusion which will be discussed later).

A second essential factor that affects fusion detection performance is the filtering criteria

since candidate fusion transcripts from preliminary alignment can easily generate thousands of false positives. Most pipelines require minimal threshold of spanning and split reads (see column 4 in Table A.2) that support the finding of a fusion transcript. Many also require a minimal thresholds of anchor length filtering (i.e. the minimum base pairs on either fused genes) for split reads (column 2 in Table A.2). In this paper, we set minimum supporting spanning and split reads to be 3 and 1 and minimum anchor length to be 10 bp whenever the pipeline allows the setting to be specified. Many tools also provide advanced filtering for read-through transcripts, PCR artifacts, gene homologs (e.g. homologous or repetitive regions, or pseudo genes) and checking against existing fusion transcript databases. Table A.2 provides all details of the parameters or availability of filtering criteria in each pipeline. In the final column, we also commented on any installation or application complexity of the tools.

Different fusion detection tools contain tremendously different sets of parameters and definitions. For example, FusionCatcher contains more than 40 parameters, including trimming options, search fusion gene options, filtering options and so on. On the other hand, BreakFusion has only several parameters that can be changed. In our experience, parameter settings can greatly influence the detection performance. For example, when we applied the default setting to TopHat-Fusion, no fusion transcript was detected in the Melanoma datasets (see real data section). But the performance improved significantly when we changed several key parameters (see Table A.8). How to set the best parameter setting for each tool and each dataset is obviously beyond the scope of this paper. As a result, we decided to only fix several key parameters whenever possible, otherwise we followed the default setting in each tool. In addition to minimum spanning reads (≥ 3), minimum split reads (≥ 1) and anchor length (≥ 10) described above, we allowed 1 mismatch per 25 bp (i.e. 2, 3 and 4 mismatches for 50, 75 and 100 bp reads, respectively) (see Table A.9 for parameter setting details for each tool). For the insert size parameters (mean and standard deviation) in the tools, we provided the truth for synthetic data and performed estimation using BWA [Li and Durbin, 2009] for real data. Among the 15 pipelines, we only fine-tuned TopHat-Fusion since TopHat tools are very popular in the field but TopHat-Fusion performed poorly in real data under the default setting (Table A.8). Whenever a tool cannot run in a specific dataset,

we attempted to debug and/or contact the authors to solve the problem. Table A.10 lists all remaining failure runs after all the efforts that lead to several incomplete results in Table 2.1. Specifically, FusionHunter failed for all synthetic data and ShortFuse also failed for most of them, so we could only effectively compare 13 tools in synthetic data.

Table 2.1: F-measure for three representative synthetic datasets and three real dataset. Type-1A: read 100 bp under 100X coverage for type-1A synthetic data; Type-1B: read 100 bp under 100X coverage for type-1B synthetic data; Type-3B: read 50 bp type-3B synthetic data (mean F-measure of the five control samples); Breast cancer: pool 4 samples of breast cancer datasets; Melanoma: pool 6 samples of melanoma datasets; Prostate cancer: pool 5 samples of prostate cancer datasets.

Tools	Type-1A	Type-1B	Type-3B	Breast cancer	Melanoma	Prostate cancer	Sum of syn data	Sum of real data	Sum of all data
SOAPfuse	0.882	0.883	0.850	0.421	0.169	0.148	2.615	0.738	3.353
FusionCatcher	0.777	0.791	0.750	0.405	0.300	0.209	2.318	0.914	3.232
JAFFA	0.693	0.672	0.702	0.543	0.267	0.006	2.067	0.816	2.883
EricScript	0.779	0.804	0.752	0.291	0.074	0.006	2.335	0.371	2.706
chimerascan	0.737	0.706	0.689	0.267	0.049	0.010	2.132	0.326	2.458
PRADA	0.545	0.543	0.540	0.469	0.334	0	1.628	0.803	2.431
deFuse	0.630	0.854	0.561	0.235	0.095	-	2.045	0.330	2.375
FusionMap	0.684	0.711	0.606	0.075	0.041	0.004	2.001	0.120	2.121
TopHat-Fusion	0.488	0.557	0.539	0.300	0.200	0	1.584	0.500	2.084
MapSplice	0.488	0.500	0.504	0.400	0.182	0	1.492	0.582	2.074
BreakFusion	0.707	0.569	0.454	0.016	0.004	0	1.730	0.020	1.750
SnowShoes-FTD	0.039	0.039	0.039	0.639	0.500	0.435	0.117	1.574	1.691
FusionQ	0.651	0.479	0.349	0.017	-	-	1.479	0.017	1.496
FusionHunter	-	-	-	0.520	0.421	-	-	0.941	0.941
ShortFuse	-	-	-	0.543	0.291	-	-	0.834	0.834

2.2.2 Description of evaluated datasets

2.2.2.1 Real data The real datasets in this study consisted of 4 breast cancer cell lines (BT-474, SK-BR-3, KPL-4 and MCF-7) [Edgren et al., 2011], 6 melanoma samples (M980409, M010403, M000216, M000921, M990802 and 501Mel) [Berger et al., 2010] and 5 prostate cancer specimen (171T, 165T, 158T, 49T and 159T) [Yu et al., 2014]. There were a total of 27 experimentally verified fusion events for breast cancer cell lines, 11 for melanoma samples and 12 for prostate cancer specimen that will serve as the underlying truth for evaluation. Table A.7 describes the details of the three real datasets.

2.2.2.2 Three synthetic data sets We first created two types of fusion transcripts for synthetic data in this study (as shown in Fig. 2.1A): (1) a fusion transcript with the associated fusion breakpoint formed by two intact exons (IE) from two different genes (called an IE-type fusion transcript); and (2) a fusion transcript with the left and/or right sides around the associated breakpoint being a broken exon(s) (BE) (called a BE-type fusion transcript). Here we simulated paired-end RNA-Seq data with synthetic fusion transcript events using the simulator in EricScript [Benelli et al., 2012]. Type-1A synthetic data were generated from the 5' and 3' end of the chimerical transcripts using wgsim (https://github.com/lh3/wgsim) with insert size 500 ± 50 bp. We generated datasets with five different coverages of 5X, 20X, 50X, 100X and 200X, each with three read lengths 50, 75 and 100 bp. The dataset with the largest coverage, i.e. 200X, was first simulated and then other datasets with smaller coverages (5X, 20X, 50X and 100X) were sequentially generated by subsampling (Table A.4). For each synthetic dataset, we simulated 80 IE-type fusion transcripts and 70 BE-type fusion transcripts (Table A.3). As a result, we generated 15 datasets in type-1A synthetic data and each dataset contained 150 true fusion transcripts.

In real experiments, the insert size (i.e. the DNA fragment size between paired-end adapters) can be pre-specified and designed by control reagent and fragmentation time in the protocol (TruSeq RNA Sample Preparation v2 Guide). Fig. 2.1C illustrates the short and long insert size DNA fragments with paired-end reads aligned to them. Left figure shows short insert size where paired-end reads cover most of the DNA fragment or even overlap in the middle; right figure shows long insert size where distance between the paired-ends is much larger. In the literature, reads with longer insert size help to detect long-range isoforms in paired-end RNA-seq and reads with shorter insert size and deeper coverage can fill in the gaps (An introduction to next-generation sequencing technology) [Katz et al., 2010]. Similarly, to detect fusion transcripts, library with longer insert size provides more spanning reads. Furthermore, some algorithms use the insert size of supporting reads as an criterion to filter out potential false positives. For example, FusionMap includes abnormal insert fragment size filtering, and this step can greatly influence the result [Ge et al., 2011]. In this paper, using BWA alignment tool [Li and Durbin, 2009], we estimated the insert sizes of three paired-end real data to be around 180 ± 80 bp, 400 ± 150 bp and 150 ± 40 bp in the

breast cancer, melanoma and prostate cancer data, respectively (Table A.7). As a result, we generated a second type of synthetic data (type-1B) using the same procedure as type-1A synthetic data except for smaller insert size at 250 ± 50 bp.

In type-2 data, we further used a control dataset from a normal lung tissue sample (SRR349695) [Zhang et al., 2012], in which we assumed no fusion transcript existed (though fusions may also exist in normal tissues). We randomly chose 2 million reads with read length 100 bp from this control sample and then trimmed the reads at 3' end to 75 bp and 50 bp to form the other two read length sets. This kind of dataset served as a negative control to benchmark whether the tools generate false positives from no-signal data. In type-3A data, we generated synthetic datasets with insert size 164 ± 48 bp (which was the insert size estimated from type-2 data) under 100X with length 50, 75 and 100 bp. Each dataset also contained the same 80 IE-type and 70 BE-type fusion transcripts. In type-3B data, we mixed type-2 and type-3A data together to test the background influence to the fusion detection tools. To increase the reliability of the comparison, we also used four additional normal samples – parathyroid (SRR479053) [Haglund et al., 2012], skeletal myocyte (SRR1693845) [Väremo et al., 2015], bladder (SRR400342) and T cell (SRR1909130) [Cao et al., 2015] samples – to generate type-2 data and combined with type-3A data (with their own insert size respectively) to generate type-3B synthetic data. Table A.5 and A.6 show details of type-2 and type-3 synthetic data. All these synthetic data sets contain the same 150 designed fusions (Table A.3).

2.2.2.3 Validation data set To evaluate the performance of meta-caller (will be introduced in Results section), an experimentally synthesized fusion sequencing dataset was used to serve as validation data (SRP043081, SRR1659964) [Tembe et al., 2014]. This paired-end dataset contains nine designed fusion transcripts as the underlying truth.

2.2.3 Performance benchmarks and evaluation criteria

We benchmarked different fusion detection tools using three evaluation criteria below. The first precision-recall plot and F-measure served as the primary benchmark for detection ac-

curacy performance which can be used for both synthetic and real data. The second criterion of supporting read identification was used only in synthetic data and mainly benchmarked the alignment efficiency. Finally, computational efficiency was evaluated to assess feasibility of the tools for big data sets with deep sequencing and/or large sample size.

2.2.3.1 Precision-recall plot In synthetic data, exactly 150 true fusion transcripts were known (Table A.3) to benchmark the performance of different methods. However, in real data, only a small set of validated fusion transcripts was available. Since a detection tool only reports the findings of possible fusion transcripts and the total positives were not entirely known in real data, popular receiver operating characteristic (ROC) curves for classification evaluation were not applicable.

Instead, the scenario was similar to information retrieval problems [Salton and McGill, 1986], in which the precision-recall curve was a better benchmark of the performance. Suppose TP, FP and FN are the true positives, false positives and false negatives of the findings from a detection tool. The precision rate (a.k.a. positive predictive value) is defined as TP/(TP+FP) that reflects the accuracy among the claimed fusion transcripts. High precision, however, does not guarantee good performance since one method can conservatively call only few fusion transcripts with high accuracy. As a result, we need the method to also have high recall rate (a.k.a sensitivity) defined as TP/(TP+FN). The precision-recall plot (precision on the y-axis and recall on the x-axis) seeks a method to have high precision and high recall near the (precision, recall)=(1,1) area. For a given result from a detection tool, we ranked the detected fusion transcripts according to the number of identified supporting reads (sum of spanning and split reads) and derived a precision-recall curve under different top numbers of detected fusions' thresholds. The classical F-measure simultaneously considers the effect of the precision and recall rates by taking the harmonic means of the precision and recall rates (i.e., F-measure = 2 $\times$ precision $\times$ recall / (precision + recall)), and was used to benchmark different methods.

2.2.3.2 Identification of supporting reads in synthetic data Identification of supporting spanning and split reads is a reflection of alignment accuracy and is the basis of

fusion transcript detection. Following the convention in the previous sub-section, we focused on the 150 true fusion transcripts in synthetic data and calculated the number of detected supporting reads in each true fusion transcript. In the distribution plot, a point (u, v) means that u out of the 150 true fusion transcripts have at least v detected supporting reads using the given detection pipeline. To better quantify and visualize similarity of identified supporting reads from different tools and the underlying true supporting reads, we applied multi-dimensional scaling plots where the dissimilarity measure between any two supporting read lists is defined as the sum squared differences of supporting reads (sum of spanning and split reads) of the 150 true fusion transcripts. The MDS plot helps quantify clusters of tools with similar alignment and supporting read detection performance.

2.2.3.3 Computational cost Recent reports have shown that sequencing depth is an important factor in detecting cancer related fusion transcripts due to tumor cell heterogeneity (i.e. a fusion transcript may only exist in partial tumor cells) [Yu et al., 2014; Luo et al., 2015]. In high coverage data, many pipelines demanding large memory and computing may become infeasible. We used four CPU cores for each fusion transcript detection tool on the type-1A synthetic data with read length 100 bp under coverage 50X, 100X and 200X to benchmark computing time for small datasets. Furthermore, to test the tool capacity to handle large datasets, we used 8 cores on the prostate cancer 171T dataset and its one-half, one-fourth and one-eighth subsamples (Table A.11) and attempted to characterize whether the computing time was increased at linear, sub-linear or super-linear rate. The machine is Linux-based, with AMD sixteen-core CPU 2.3GHz.

2.3 RESULTS

2.3.1 Evaluation in synthetic data

2.3.1.1 Type-1A and 1B Synthetic Data In type-1A synthetic data evaluation, all 15 fusion detection methods (Table A.2) were applied to 15 datasets of five coverages (5X,

20X, 50X, 100X, and 200X) and three read lengths (50, 75 and 100 bp). FusionHunter failed for all synthetic data and ShortFuse failed for most of them (see Table A.10, failed trials were excluded from further analysis). Fig. 2.2A indicates the numbers of true positives (bars shown on the y-axis, solid bars for IE-type and slashed bars for BE-type) and total numbers of fusion detection (the numbers marked on top of the bars) by each tool for read length 100 bp results (results for 50 bp and 75 bp are shown in Fig. A.1) for type-1A synthetic data. Fig. A.2 shows the 15 F-measures (as well as precisions and recalls) for five coverages and three read lengths in type-1A synthetic data (results of 100X and 100 bp read length are marked by red cross). For a representative demonstration, Fig. 2.2D shows the precision-recall curves in the 100X and 100 bp read length setting. In precision-recall plots, tools that generate higher recall rate under the same precision rate demonstrate better performance. In Fig. 2.2A, increasing coverages improved detection sensitivities for almost all tools. Most tools were equally powerful in detecting both IE and BE types of fusion transcripts except that PRADA and SnowShoes-FTD could not detect any BE-type fusions. When comparing impact of read length (Fig. 2.2A and Fig. A.1), increased read length under fixed coverage did not improve the detection sensitivity. This was probably because under fixed coverage, increasing read length decreased the total number of reads in the dataset (Table A.4). This finding was consistent with a previous report in bisulfite sequencing [Krueger et al., 2012]. By balancing precision and recall in Fig. 2.2D and Table A.12, we can visually identify SOAPfuse, FusionCatcher and EricScript to achieve high recall rate (up to 92.7% for SOAPfuse, 72.0% for FusionCatcher and 69.3% for EricScript) while maintaining high precision ($\approx$ 80% - 90%). JAFFA and PRADA appeared to be conservative but accurate tools that can achieve only 58.7% and 38.0% recall rate but maintained high precision rate (84.6% for JAFFA and 96.6% for PRADA). The complementary performance of these top performing tools motivated the development of the ensemble method to combine these methods in a later section.

Similar to type-1A evaluation, Fig. 2.2B and 2.2E show information of detected true positives and precision-recall curves at read length 100 bp for type-1B synthetic data (insert size 250 ± 50 bp)(Fig. A.3 shows results for 50 and 75 bp; Fig. A.4 shows F-measure of all 15 settings; Table A.13 shows F-measure of 100bp dataset). In these shorter insert size

data, tools were more sensitive to sequencing coverage. For example, BreakFusion detected (33-3)/3=10-fold more true fusions when increasing the coverage from 5X to 20X. Similarly, JAFFA and PRADA identified 4.8-fold and 4.6-fold more true fusions. Even SOAPfuse and FusionCatcher, which were not sensitive to low coverages at 500 bp insert size datasets, detected 65 and 40 more true positives (TPs) from 5X to 20X.

2.3.1.2 Type-2 and type-3 synthetic data with background noise In most cancer applications, tumor cells are often contaminated by adjacent normal cells to cause heterogeneity. To investigate the influence of such background noise, we first randomly generated type-2 synthetic data from normal lung tissues (SRR349695) [Zhang et al., 2012] (or parathyroid (SRR479053) [Haglund et al., 2012], skeletal myocyte (SRR1693845) [Väremo et al., 2015], bladder (SRR400342) and T cell (SRR1909130) [Cao et al., 2015] sample) that were assumed to contain no designed fusion event. We then generated synthetic data containing 150 true fusion transcripts in type-3A and then mixed type-2 and type-3A data to form type-3B synthetic data. Since the insert size for type-2 data is small (164 ± 48 bp for lung sample), we mainly focused on the results with read length 50 bp. Fig. 2.2C shows the result of type-2 (BG), type-3A (100X) and type-3B (100X+BG) lung tissue synthetic data at read length 50 bp (Fig. A.5 similarly shows results for 75 and 100 bp; Fig. A.6 shows F-measure of three read lengths; Fig. A.7 shows detection results for the other four tissues on 50 bp read length and Fig. A.8 shows their corresponding F-measures; Table A.14 shows F-measure of 100bp dataset and Table A.15 shows the correlation between five tissues by the F-measure of the 15 tools). From type-2 dataset (BG) in Fig. 2.2C, all tools detected almost none fusion transcripts as they were supposed to, except that FusionQ detected 28 false positives (FPs). Comparing results of type-3A and type-3B, BreakFusion increased the total number of detections significantly while the TPs remained almost the same. FusionQ was also sensitive to background influence, whose TPs increased significantly (from 9 to 37) with the sacrifice of increasing the total detections (from 10 to 74). DeFuse was also influenced by background noise with less TPs detected (decreased from 70 to 43). On the other hand, methods such as SOAPfuse, FusionCatcher, JAFFA, EricScript, chimerascan, PRADA, FusionMap, TopHat-Fusion and MapSplice were almost not influenced by the

background noises. Fig. 2.2F shows the precision-recall curves for type-3B synthetic data. Overall, FusionCatcher and EricScript performed the best to maintain high precision and stayed robust from background noise (Fig. 2.2C and 2.2F).

2.3.1.3 Alignment efficiency and detection similarity across pipelines To compare the alignment efficiency of each tool with the underlying truth, we analyzed the number of detected supporting reads for the 150 designed fusion transcripts (as well as 80 IE-type only and 70 BE-type only) via type-1A synthetic data. In Fig. 2.3A-C, for each tool, the y-axis of the distribution plot represents the number of detected designed fusion transcripts based on consideration of the fusion transcripts with the number of total identified supporting reads (sum of spanning and split reads) being larger than the specified values set on the x-axis. The black line represents the results of the ground truth and other color lines represent different tool results. The closer the lines of the tools to the ground truth, the better the ability of correctly aligning the supporting reads. Fig. 2.3A, 2.3B and 2.3C considered the total 150 designed fusions, 80 IE-type and 70 BE-type fusions, respectively. These figures show the results for type-1A synthetic datasets with 100X and 100 bp read length (the results with read lengths 50 and 75 bp under 100X coverage are shown in Fig. A.9). In Fig. 2.3A-C, we note that except for SOAPfuse, all the other tools missed some of the true fusions (e.g. all other tools missed 50-100 fusions in Fig. 2.3A). Of them, FusionCatcher (solid orange), EricScript (solid bright pink), JAFFA (solid bright green), TopHat-Fusion (dash dark green), FusionQ (dash red), deFuse (solid dark purple) and MapSplice (dash orange) seemed to have preferential alignment efficiency on a subset (50-80) of true fusion transcripts and can detect high supporting reads for partial of them (flat decreasing curves in Fig. 2.3A-C). Other callers tended to have sudden drops at 50-100 supporting reads, showing overall under-performance of alignment. SOAPfuse's superior alignment capability was consistent with the finding in a previous report [Ruffalo et al., 2011]. It required higher computational cost (see Computational Efficiency section) but it can also include modest number of false positive reads (number of supporting reads greater than the truth on the high end). This may explain SOAPfuse's high recall rate ($\approx 90\%$) and high precision rate ($\approx 80 - 90\%$) in Fig. 2.2D.

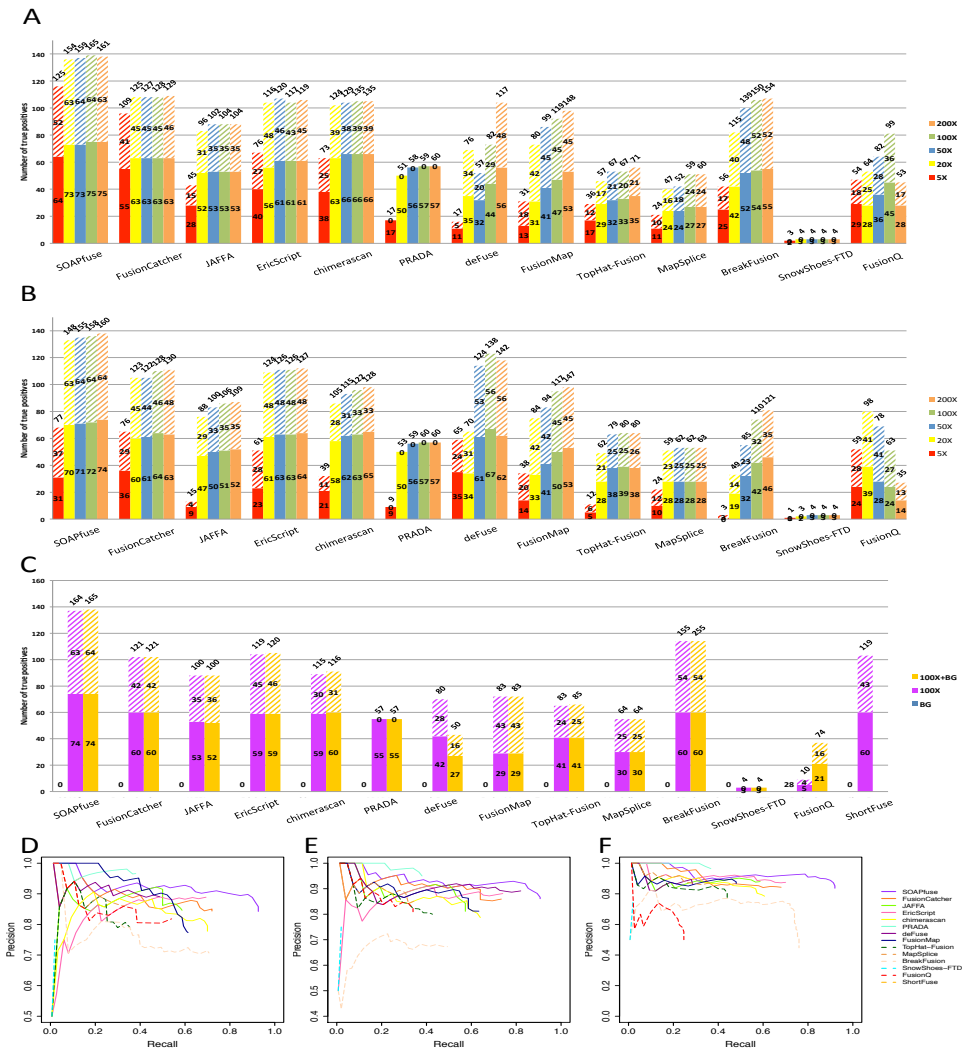


Figure 2.2: Fusion transcript detection results for synthetic datasets with 100 bp read lengths. (A)-(C): The y-axis bars show the number of true detected positives, among them IE-type and BE-type fusions are shown in solid and slashed rectangles. The total number of fusion detections are shown on the top of the bars. (A) Result for type-1A synthetic data (100 bp read length), (B) result for type-1B synthetic data (100 bp read length) and (C) result for type-2, type-3A and type-3B synthetic data (lung sample 50 bp read length). (D) Precision-recall plot for type-1A synthetic data (100 bp read length & 100X). (E) Precision-recall plot for type-1B synthetic data (100 bp read length & 100X). (F) Precision-recall plot for Type-3B synthetic data (lung sample 50 bp read length & 100X).

In Fig. 2.3D-F, we further examined the alignment similarity of the tools by multi-dimensional scaling (MDS) plots (tools closer to each other had more similar fusion supporting reads detection) in 100 bp read length. The result showed a close-to-the-truth performance of SOAPfuse, FusionCatcher and EricScript. FusionQ appeared to have very different alignment result from all other methods although its overall cumulative distribution did not much differ. The differential pattern of supporting reads detection provided the basis and rationale to combine multiple callers for improving fusion detection (discussed later). The results for 50 and 75 bp are shown in Fig. A.10.

2.3.1.4 Balance between precision and recall curve Precision and recall rates assess the tradeoff between true positives and false positives, measuring the tools' ability to detect more TPs with the cost of less FPs. A high recall rate indicates that the algorithm could detect most of the 150 true fusion transcripts while a high precision rate indicates that most of the fusion transcripts detected are true positives. In our analysis, we used precision-recall curves and calculated F-measure that balances between precision and recall (see Methods section) to benchmark the performance of different tools. The first three columns in Table 2.1 shows the F-measures of Type-1A, 1B and 3B results of different methods. As shown in Fig. 2.2D and Table A.12, the highest F-measure with 100 bp read lengths under 100X coverage in type 1A was SOAPfuse (92.7% recall rate, 84.2% precision and 0.882 F-measure), followed by EricScript (F=0.779), FusionCatcher (F=0.777), chimerascan (F=0.737) and BreakFusion (F=0.707). In type-3B data with background noise, SOAPfuse performed the best, followed by EricScript, FusionCatcher and JAFFA (Fig. 2.2F and Table A.14). Of special note was JAFFA and PRADA that maintained high precision rate while only had a comparatively low recall rate. Such complementary calling properties implied the possibility of combining FusionCatcher and SOAPfuse, as well as other top performing tools, for further improvement.

2.3.2 Evaluation in real datasets

In the three real datasets, we had 27, 11 and 12 wet-lab validated fusion transcripts but the full true fusion transcripts were not entirely known. As a result, we drew similar bar plots in

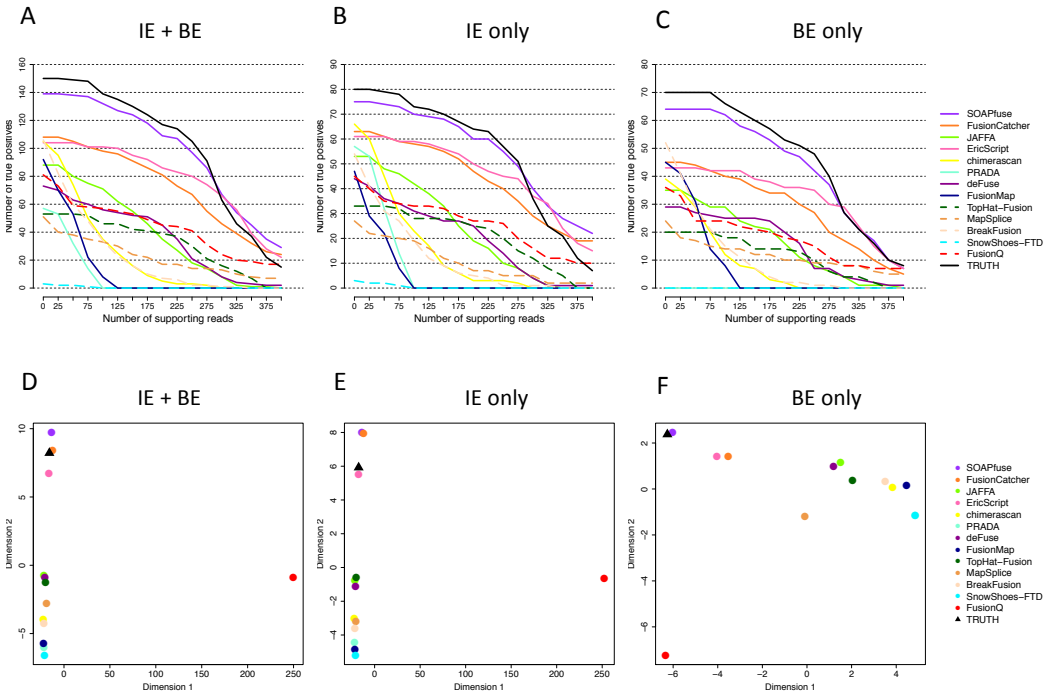


Figure 2.3: Illustration of alignment performance and similarity across tools for type-1A synthetic data with 100 bp read length & 100X. (A)-(C): Number of true positives (y-axis) with detected supporting reads greater than the threshold on the x-axis. (D)-(F): Multi-dimensional scaling (MDS) plots to demonstrate pairwise similarity of detection results from 15 tools and the underlying truth. (A) and (D): Results for all 150 true fusion transcripts. (B) and (E): Results for only IE-type fusion transcripts. (C) and (F): Results for only BE-type fusion transcripts.

Fig. 2.4A-C and used precision-recall plots and F-measure to benchmark the performance of the tools (Fig. 2.4D-F, Table A.16 A.17 and A.18). For example, in Fig. 2.4A SOAPfuse identified 35 candidate fusion transcripts in the BT-474 breast cancer cell line, among which 9 cases were validated. In total, SOAPfuse detected 68 fusion candidates across all four

breast cancer samples, of which 20 were validated (precision = 20/68 = 29.4% and recall = 20/27 = 74.1%). On the contrary, in prostate cancer example in Fig. 2.4C, EricScript detects 3809 fusion candidates, of which 11 were validated (precision=11/3809=0.3% and recall=11/12=91.7%). By comparing F-measure that balancing between precision and recall, we found that performance of methods varied greatly in different real datasets. Based on Table 2.1, SnowShoes-FTD, FusionHunter and FusionCatcher are better performers in real data. In these real data, several tools could not complete running in partial datasets. We had made our best effort to debug the pipelines, contacted authors and recorded all unfinished tasks in Table A.10 after all possible effort. Such cumbersome debugging processes are often encountered when using these pipelines.

2.3.3 Computational efficiency

Since fusion detection involves analysis of large sequencing datasets and complex analysis pipeline, computational efficiency is an important benchmark, especially for projects involving deep sequencing and large sample size, an expected trend in the field. Fig. 2.5A shows the computation time (log-scale on the y-axis) of small datasets using synthetic data with read length 100 bp and coverage 50X, 100X and 200X. FusionMap appeared to be the fastest algorithm, followed by similar speed of EricScript, JAFFA, SnowShoes-FTD, MapSplice, PRADA and TopHat-Fusion. SOAPfuse had good performance in alignment accuracy and precision-recall evaluation in synthetic data and real data but it apparently required much more computational resources. Each fusion detection pipelines had its own time-consuming steps based on its work-flow and tools involved [Kim and Salzberg, 2011]. We used the computing time at 200X and linearly projected to 1/2 and 1/4 computing time for 100X and 50X with the dashed lines. The result showed that computing time increased in a 'sub-linear' pattern for most methods in these datasets (i.e. doubling coverage took less than double computing time). This was reasonable because large percentage of the computing was spent on preliminary processing, library preparation and some post-processing steps for such small data sets. For example, after aligning the reads into BAM file, BreakFusion consists of five steps: identify breakpoint, assemble putative junctions, BLAT junctions to genome, esti-

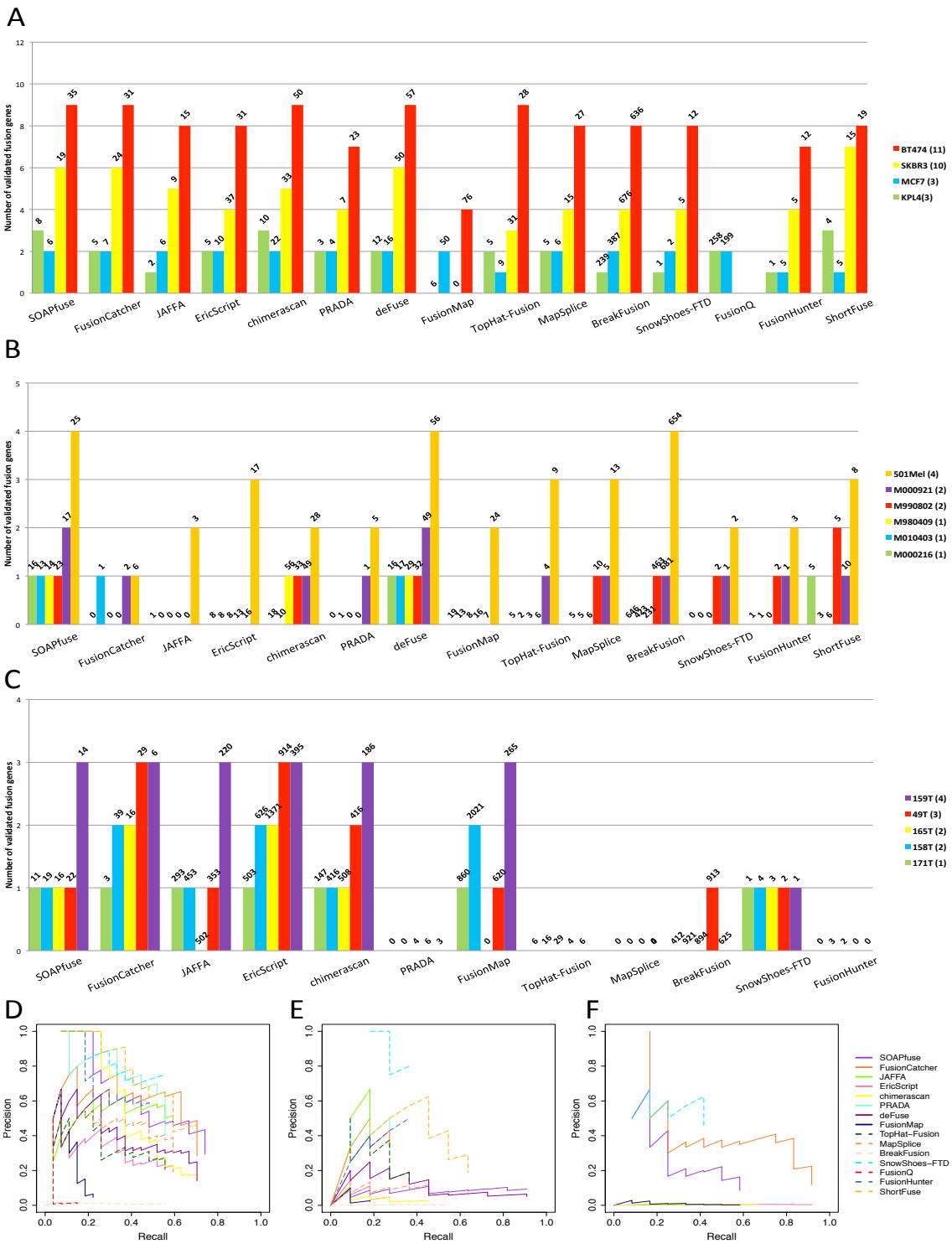


Figure 2.4: Fusion transcript detection results for three real datasets. Figures are similar to Figure 2. (A) and (D): Breast cancer dataset; (B) and (E) Melanoma dataset; (C) and (F): Prostate cancer dataset.

mate chimeric scores and annotate-and-filter [Chen et al., 2012]. We further tested another large dataset of prostate cancer sample 171T (118,742,381 reads with 100 bp read length) in Fig. 2.5B using the entire, 1/2, 1/4 and 1/8 randomly subsampled sequences (Table A.11). SOAPfuse remained computational costly while JAFFA, deFuse and MapSplice appeared to surpass computational needs of SOAPfuse. DeFuse even failed to complete for the entire sequencing dataset (did not stop after 16 days). FusionMap, FusionHunter and SnowShoes-FTD were the most computationally efficient methods. PRADA and deFuse required super-linear computing time for large datasets (i.e. doubling coverage required more than double of computing time). Practitioners should pay extra attention to plan enough computing power for these pipelines when running projects with deep sequencing and large sample size.

2.3.4 An ensemble algorithm by combining multiple top-performing fusion detection tools

Table 2.1 shows the F-measures of each detection method applied to each synthetic and real dataset. By ranking the sum of F-measures over three synthetic datasets and three real datasets, several methods such as SOAPfuse and FusionCatcher consistently performed well in most datasets but no method was always the top-performer. Strikingly, EricScript, chimerascan, deFuse and FusionMap performed well in synthetic data (sum of F-measures = 2.335, 2.132, 2.045 and 2.001) but performed poorly in real data (sum of F-measures=0.371, 0.326, 0.330 and 0.120). On the other hand, PRADA, SnowShoes-FTD, FusionHunter and ShortFuse performed well in real data (sum of F-measures=0.803, 1.574, 0.941 and 0.834) but performed poorly or failed to run in synthetic data (sum of F-measures=1.628, 0.117, failed and failed). Such a discrepancy may reflect the fact that the simulation model may be overly simplified. The three real datasets also shows some heterogeneity. Particularly, many methods could not run or almost detected nothing for the largest prostate cancer dataset because of the large size of the data and less validated fusion transcripts. Due to the limited availability of real data sets with enough amount of validations, we believe that the three real datasets may not reflect the comprehensive characteristics that users may encounter in

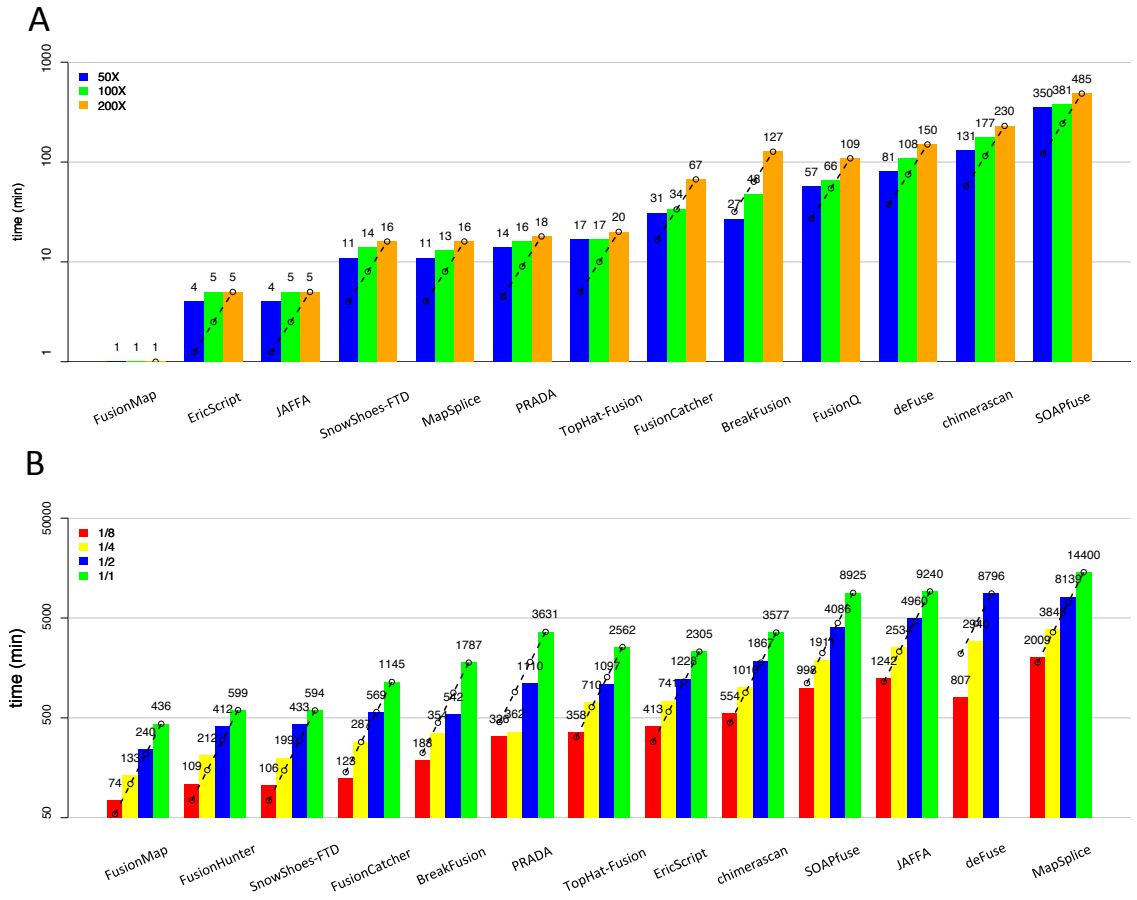


Figure 2.5: Computational cost comparison. The bar plots (y-axis) show the log-scaled computational time (min). Dashed lines project from the largest dataset with linear computing time decrease by coverage and can be used to determine linear, super-linear (bars for smaller coverages fall below the line) or sub-linear (bars for smaller coverages exceed the line) computing load. (A) Evaluation using type-1A synthetic data for read length 100 bp at 50X, 100X and 200X. (B) Evaluation using prostate cancer 171T sample.

their real data. As a result, we recommend users to apply SOAPfuse, FusionCatcher and JAFFA in order based on the sum of rank of the F-measures from Table 2.1.

In Fig. 2.2, Table A.12, A.13 and A.14, we have observed that SOAPfuse can achieve above 90% recall rate while FusionCatcher and JAFFA can reach high precision but low

recall rate. This created a possibility of combining results of these top three pipelines to improve detection performance provided that fusions detected by FusionCatcher were not all detected by SOAPfuse. In other words, top performing methods likely had complementary advantages to accurately detect different types of fusion events. To test this hypothesis, we combined the three top-performing methods (SOAPfuse, FusionCatcher and JAFFA) to construct a meta-caller. First of all, we selected fusion events detected by at least two out of the three methods (Step 1 of Fig. 2.6). We next ranked the detected fusion events from each method by the number of supporting reads, where larger number of supporting reads obtained larger rank (Step 2 of Fig. 2.6). Rank sums of the selected fusion events were calculated (where missing values of the ranks were ignored if the fusion event was not detected by one of the methods) and the fusion events were re-prioritized accordingly. To test validity of the new meta-caller, Fig. 2.7 shows the precision-recall performance of the three top-performing methods as well as the meta-caller (dash black) in different datasets: Fig. 2.7A-C for type 1A, 1B and 3B (lung sample) synthetic data with 100X coverage and read length 100, 100 and 50 bp respectively (Fig. A.11 shows the meta-caller performance of the other read lengths for synthetic dataset); Fig. 2.7D-F for pooled breast cancer, melanoma and prostate cancer real data. In all situations, the meta-caller performed better or at least equal to the best of the three top-performers. We have also tried to combine top 6 performer (ranked by Table 2.1, containing SOAPfuse, FusionCatcher, JAFFA, EricScript, chimerascan and PRADA) and re-ranked the fusion transcripts that detected by at least 3 tools. The precision and recall curve of the top 6 performer was shown in Fig. A.12 and its performance is slightly better than top-3 performer, but it takes larger computing efforts.

Admittedly, it's overfitting to use our synthetic and real data to validate the performance of meta-caller since the tools are evaluated and ranked from these datasets. So we used a new dataset sequenced from an experimentally-synthesized fusion transcripts library (nine designed underlying truth) [Tembe et al., 2014] as the validation data to evaluate the meta-caller performance. Table A.19 showed the performance summary of each tool. We also implemented top-3 (Fig. 2.8) and top-6 (Fig. A.13) meta-callers to combine the results from single tools and the performance still kept on top of single methods (except for equal or slightly worse than FusionCatcher). This provides a strong evidence to the hypothesis that

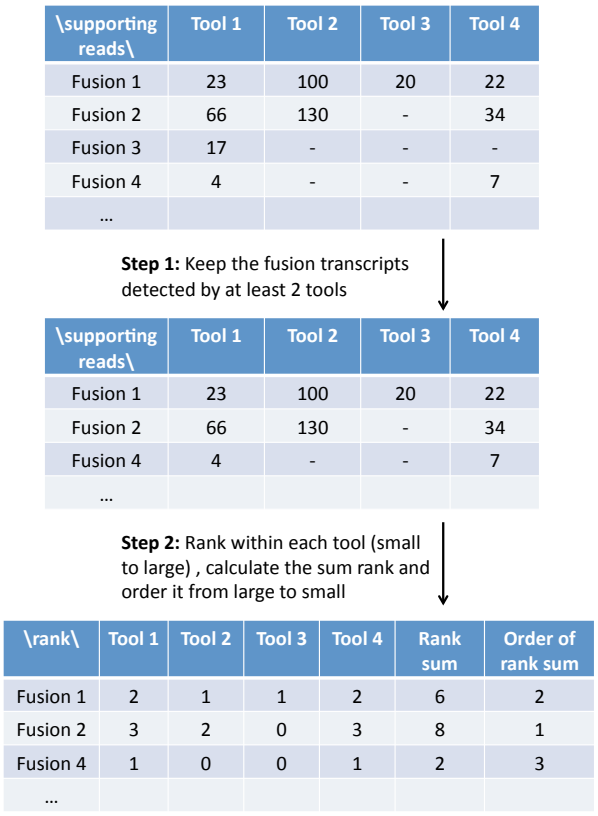


Figure 2.6: Illustration of the meta-caller workflow.

meta-caller improves detection result by combing multiple top-performing tools.

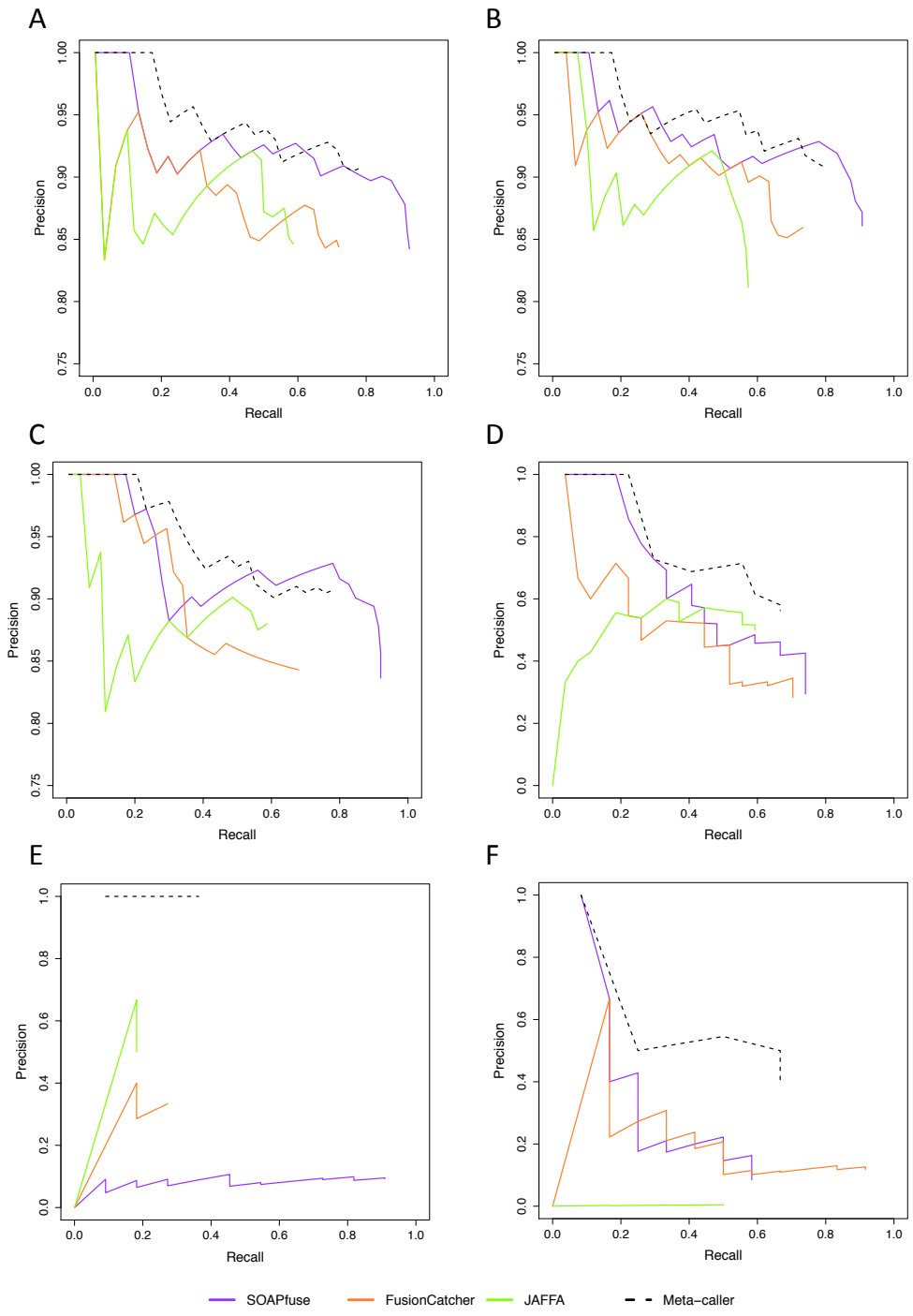


Figure 2.7: Precision-recall curves of top 3 performing tools and meta-caller. (A)-(C): Type-1A, type-1B and type-3B (lung sample) synthetic data with 100X coverage and 100, 100 and 50 bp read length respectively. (D)-(F): Three real datasets: breast cancer, melanoma and prostate cancer.

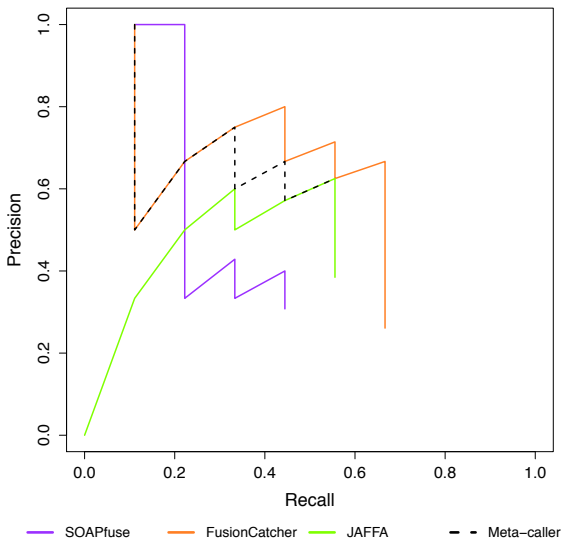


Figure 2.8: Precision-recall curves of top-3 performing tools and meta-caller (with majority vote=2) on validation data.

2.4 DISCUSSION AND CONCLUSION

In this paper, we performed a large-scale comparative study by applying 15 fusion transcript detection pipelines to three synthetic datasets and three real paired-end RNA-seq studies on breast cancer cell lines, melanoma samples and prostate cancer specimen. We used precision-recall plots and the associated F-measures to serve as the primary performance benchmark for both synthetic and real data (Fig. 2.2D-F, Fig. 2.4D-F and Table 2.1). In the synthetic data, the underlying truths are known so we further investigated the identified supporting reads of true fusions from each pipeline as the secondary benchmark to quantify alignment performance (Fig. 2.3). To evaluate computational cost of each tool for large sequencing projects, we evaluated running time as the third benchmark (Fig. 2.5). Finally, we developed a meta-caller algorithm to combine three top-performing methods (SOAPfuse, FusionCatcher and JAFFA) determined by F-measure (Fig. 2.6). The meta-caller was evaluated in the three synthetic and real datasets as well as an independent experimental

data set. The result provided a proof-of-concept justification that the meta-caller almost always performed better or at least equal to the best performer in each synthetic or real data scenario and should be recommended in daily applications (Fig. 2.7 and Fig. 2.8).

Fusion detection pipelines often include multiple complicated tools using different programming languages (e.g. Perl) and can be easily impacted by local machine setting and software versions. Unlike platform independent programming languages such as Java and R, fusion detection pipelines often require extensive script checking and debugging when the code is transported to a new machine or even rerun on the same machine after an extensive time period with possible software upgrades. In this paper, we have made our best effort to generate comparable evaluations by specifying versions of each tool, key parameters expected to impact the calling discrepancy (e.g. allowed alignment mismatches, minimal supporting split and spanning reads, minimal anchor lengths and etc.) and keep default settings whenever possible. When the tools failed to run after extensive effort, we have contacted the authors to improve but failures still remained in multiple situations (Table 2.1 and A.10). Such hurdles are probably still expected in a near foreseeable future and next-generation sequencing forums, such as SEQanswers, can often provide great help.

We summarize key conclusions from the comprehensive comparative study below.

1. No tool performed dominantly best in all synthetic and real datasets. SOAPfuse performed consistently among the best and followed by FusionCatcher, JAFFA and PRADA in both synthetic and real datasets. EricScript and chimerascan performed well in synthetic data but poor in the three real datasets we evaluated. The performance of each tool appeared to be data-dependent and not always consistent between synthetic and real data.

2. SOAPfuse, FusionCatcher and EricScript overall had the best alignment performance in the synthetic data evaluation.

3. SOAPfuse was one of the most computationally demanding tool. FusionCatcher and JAFFA had median computation load. All of the three methods required super-linear computing in deep-sequenced samples and computing resources should be planned ahead for large projects.

4. The meta-caller combining SOAPfuse, FusionCatcher and JAFFA generated better precision and recall performance than any single tool. Whenever possible, it is recommended to apply all three pipelines and combine the results in applications.

There are several limitations to our study design. First of all, the evaluation is limited (or potentially can be biased) by the simulation models, the three available data sets and the corresponding experimentally validated fusions. We particularly observed that several tools performed well in synthetic data but poorly in real data or vice versa. Due to limited number of datasets, we decided to aggregate performance benchmark of all results equally in Table 2.1. Collecting more real datasets and/or developing more realistic simulation models for a more conclusive evaluation is a future goal.

Secondly, demonstration of the meta-caller performance (Fig. 2.7, 2.8, A.11, A.12 and A.13) serves as a proof-of-concept, with only one independent data validation. If more real datasets and experimentally validated fusions become available in the future, systematic cross-validation assessment should be performed to evaluate the meta-caller. The increased information may further inspire new meta-caller methods.

Conclusions from this paper can provide guidelines or foster future research initiatives for different audience. Although no tool dominantly performed the best, for data analysts and practitioners the comparative study can guide to avoid using ineffective tools and recommend to select the top few best pipelines. Our proposed meta-caller framework allows users to effectively combine results of multiple top performers. For developers of existing tools, our evaluation can identify the subset of fusions with low detection accuracy in their pipelines and seek improvement. When a new fusion detection pipeline is developed in the future, our study will provide an open-source evaluation framework to benchmark the new method. For the large bioinformatics community, development of a high-performing (accurate and fast) fusion detection tool or methods to combine top-performing tools remains an important and open question.

2.5 ACKNOWLEDGEMENTS

This work is supported by NIH R21MH094862 and RO1CA190766 (to YD, SK and GCT), University of Pittsburgh Cancer Institute (to SL, JL and GCT), Ministry of Science and Technology (MOST) (MOST103-2221-E-010-015, to WHT and IFC), and National Yang-Ming University, Taiwan (a grant from Ministry of Education, Aim for the Top University Plan, to WHT and IFC). We would like to thank the AE and two reviewers for insightful comments and suggestions that significantly improved this paper.

3.0 AIM 2. META-ANALYTIC FRAMEWORK FOR LIQUID ASSOCIATION

This is a pre-copyedited, author-produced version of an article accepted for publication in *Bioinformatics (Oxford, England)* following peer review. The version of record [Wang et al., 2017] is available online at https://www.ncbi.nlm.nih.gov/pubmed/28334340.

3.1 INTRODUCTION

Gene co-expression analysis is vastly applied to study pairwise gene synchronization to elucidate potential gene regulatory mechanisms. For example, an unweighted gene co-expression network can be constructed from a transcriptomic study given a co-expression measure (e.g. Pearson correlation) and an edge cut-off (e.g. two nodes are connected if absolute correlation ≥ 0.6 and disconnected if < 0.6). In the literature, different measures such as Pearson correlation, Spearman correlation and mutual information [Butte and Kohane, 2000] have been used (see Song et al., 2012 for a comparative study). Alternatively, Zhang et al. [2005] developed a WGCNA framework using cluster analysis to construct gene co-expression modules and their associated weighted co-expression networks. Network properties and extended pathway analysis can then be studied to investigate disease related network alterations and mechanisms.

Although guilt-by-association heuristic assumed in gene co-expression network analysis is widely used in genomics [Wolfe et al., 2005], many complex regulatory mechanisms in the system cannot be readily captured by direct association because of multi-way interactions. The first column in Fig. 3.1A shows an example of liquid association first described in Li [2002]. Gene YCR005C and YPL262W are overall non-correlated in study GSE11452

(Spearman correlation = 0.239) but they exhibited high correlation (cor = 0.692) when a third gene YGR175C is low expressed (expression intensity < -0.424) and a much lower correlation (cor = -0.790) when expression of gene YGR175C is high (> 0.441). The simple interaction among the trio is biologically meaningful since the third gene YGR175C may serve as a surrogate of certain (hidden) cellular state or regulator that controls the presence and absence of co-regulation between gene YCR005C and YPL262W.

To quantify the conditional association in the triplet genes, Li (2012) proposed a liquid association (LA) measure to quantify the dynamic correlation of two variables depending on a third variable [Li, 2002; Li et al., 2004; Ho et al., 2011]. Li [2002] introduced this concept and proposed a computationally efficient three-product-moment measure (see Section 3.2.2). Zhang et al. [2007] adopted a simplified LA score based on z-transformed Pearson correlation conditional on discretized expression of the third gene. Ho et al. [2011] extended the trivariate dependency structure into a parametric Gaussian framework (called modified liquid association; MLA) to develop improved estimation frameworks and statistical test for the existence of the LA dependence. The computational complexity to screen all possible triplets is $O(n^3)$ and is generally too high for applying LA methods in a genome-wide scale. Gunderson and Ho [2014] introduced an efficient screening algorithm fastLA for the MLA method containing two steps: (1) screening the candidate triplets by difference between the correlations of the LA pair when the scouting gene is high and low; (2) fitting and estimating the model based on conditional normal distributions. The algorithm greatly improved the computing efficiency for genome-wide LA analysis.

Liquid association estimated from a single study is often unstable and not generalizable due to cohort bias, biological variation, and sample size limitation. With rapid accumulation of transcriptomic studies in the public domain, identifying LA triplets by combining multiple studies is likely to produce more stable and biologically reproducible results. For example, Fig. 3.1A shows an example of an LA triplet (gene YCR005C, YPL262W, and YGR175C) where the liquid association is statistically significant in the first yeast study GSE11452 but the LA association does not hold for the remaining four independent studies. Such an association is likely condition-specific for the first study or a false positive. On the other

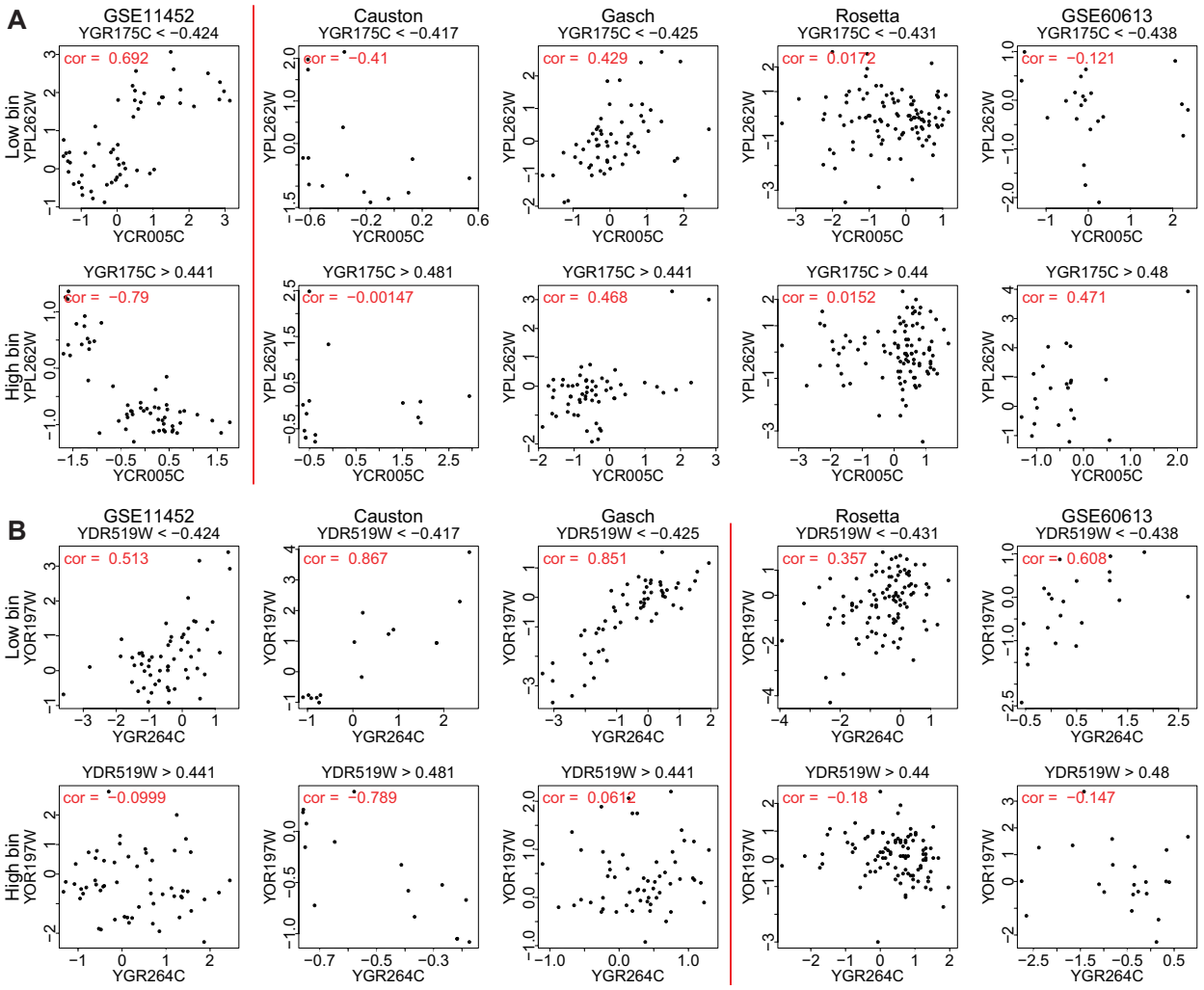


Figure 3.1: The scatter plot of the gene expressions in the high and low bins. (A) is for the triplet selected by GSE11452 through singleMLA and (B) is for the triplet selected by the studies GSE11452, Causton, and Gasch through MetaMLA.

hand, the LA triplet (YGR264C, YOR197W, and YDR519W) in Fig. 3.1B is obtained from the combined meta-analysis of the first three studies. The association is more likely to validate in the fourth and fifth studies. In this paper, we develop two meta-analytic frameworks for liquid association to accurately identify LA triplets that are consistent across multiple studies. The result shows that meta-analytic methods generate more stable LA triplets that are more reproducible in independent studies. The LA triplets also generate

better pathway enrichment results to better understand the biological insight and/or generate further hypothesis.

3.2 METHODS

3.2.1 Data sets and databases

We used five yeast (*Saccharomyces cerevisiae*) data sets – Causton [Causton et al., 2001], Gasch [Gasch et al., 2000], Rosetta [Hughes et al., 2000], GSE60613 [Chasman et al., 2014], and GSE11452 [Knijnenburg et al., 2009] – to illustrate our meta-analytic methods. In each study, yeast samples are exposed to a variety of environmental stress and the transcriptomic expression profiles are measured. Causton et al. includes a yeast gene expression series including yeasts treated with acid, alkali, heat, hydrogen peroxide, salt, sorbital, and during diauxic shift; Gasch et al. contains yeasts treated with amino acid starvation, diamide, DTT, exposure to peroxide, menadione, nitrogen depletion, osmolarity, and temperature shifts; Rosetta corresponds to 300 diverse mutations and chemical treatments; GSE60613 analyzes the stress-activated signaling network; GSE11452 corresponds to chemostat cultures under 55 different conditions. As shown in the data preprocessing step in Fig. 3.2, within each individual study we first deleted genes and samples with more than 10% and 30% missing values respectively, imputed the missing values with K-nearest neighbors algorithm [Altman, 1992], and quantile normalized the samples [Amaratunga and Cabrera, 2001]. We further performed unbiased filtering within each study to filter out non-expressed genes (lowest 35% of mean expression) and non-informative genes (lowest 35% of expression variances). Finally, our data sets include 1,770 overlapped genes across five studies and 45, 173, 300, 67, and 170 samples for study Causton, Gasch, Rosetta, GSE60613, and GSE11452, respectively.

As an *in silico* biological evaluation of the LA triplets, we downloaded yeast protein-protein interaction (PPI) database from Saccharomyces Genome database (SGD) [Cherry et al., 2011]. The database included 101,325 unique PPI pairs involving 5,706 genes. We

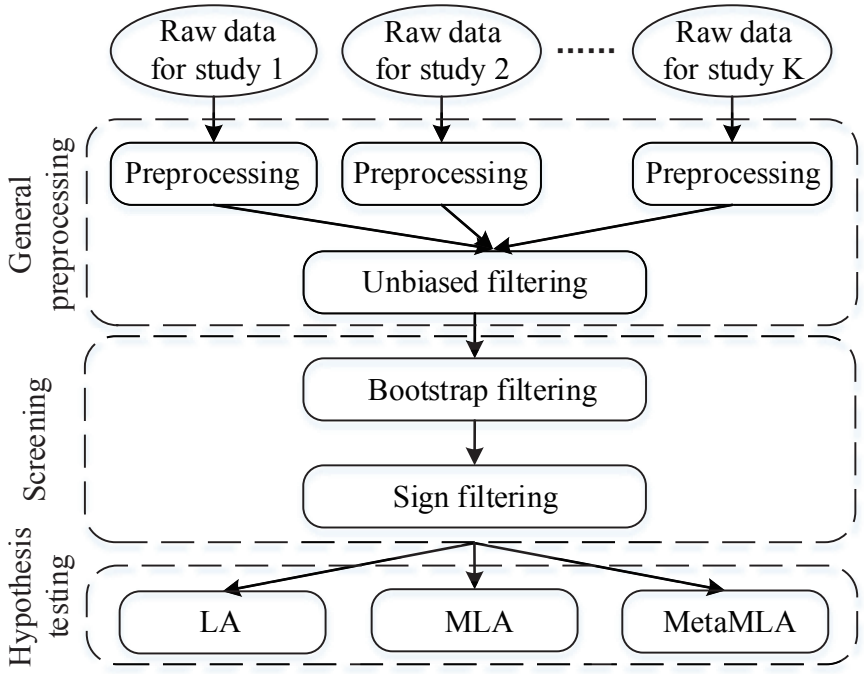


Figure 3.2: A process map of the genome-wide application of the MetaMLA algorithm.

applied pathway enrichment analysis on two databases: Gene Ontology (GO) [Cherry et al., 2011] and KEGG [Kanehisa et al., 2016] databases and obtained 1,398 GO terms and 95 KEGG pathways with at least five genes. Additionally in order to test how co-regulated genes are enriched in transcription factor (TF) binding data, we downloaded a TF binding gene sets from YEASTRACT database [Teixeira et al., 2013] and 96 gene sets with 5-200 validated genes were selected for further enrichment analysis. Fisher's exact test [Upton, 1992] was used for pathway enrichment analysis. The P-values were corrected by Benjamini-Hochberg (BH) algorithm [Benjamini and Hochberg, 1995] and the significance level was set to be $\alpha = 0.05$.

3.2.2 Liquid association methods (LA and MLA) for a single study

Li [2002] introduced the concept of "liquid association" and defined the LA score for a gene pair X_1 and X_2 given a scouting gene X_3 as $LA(X_1, X_2|X_3) = Eg'(X_3)$, where $g(x_3) =$

$E(X_1X_2|X_3 = x_3)$ and $g'(x)$ is the first derivative of $g(x)$. After standardizing the three gene expressions to fit Gaussian assumption and applying Stein's lemma, they proposed a computationally efficient estimator by $\widehat{LA} = \sum_{l=1}^{n} X_{1l}X_{2l}X_{3l}/n$, where n is the total number of observations (samples) and X_{1l}, X_{2l}, and X_{3l} are the lth observations for genes X_1, X_2, and X_3, respectively.

Ho et al. [2011] proposed a modified LA (MLA) method by $MLA(X_1, X_2|X_3) = Eh'(X_3)$, where $h(X_3) = \rho(X_1, X_2|X_3)$, $h'(x)$ is the first derivative of $h(x)$, and ρ is the Pearson correlation coefficient. They proposed a direct estimation of MLA score by $\widehat{MLA} = \sum_{j=1}^{M} \hat{\rho}_j \bar{X}_{3j}/M$, where M is the number of bins over X_3, $\overline{X}_{3j}$ is the sample mean of X_3 within bin j, and $\widehat{\rho}_j$ is the correlation of the LA pair X_1 and X_2 in bin j. A key advantage of the MLA estimator is the capability of performing hypothesis testing $H_0 : MLA(X_1, X_2|X_3) = 0$ by a Wald test statistics $T_{MLA} = \widehat{MLA}/SE(\widehat{MLA})$ to assess the P-value, where $SE(\widehat{MLA})$ is the standard error of $\widehat{MLA}$.

3.2.3 MetaMLA and MetaLA methods

In this section, we extend the original three-product-moment LA method [Li, 2002] and the model-based MLA method [Ho et al., 2011; Gunderson and Ho, 2014] into a meta-analytic scheme for combining information from multiple transcriptomic studies.

Suppose that we have K studies. For a gene triplet t: (X_1, X_2, X_3), if the LA scouting gene is $Z = X_i (i = 1, 2, 3)$, after standardizing all the three genes to have mean 0 and variance 1 and the scouting gene to follow normal distribution, the direct estimation of the MLA score [Ho et al., 2011] for the single study k $(k = 1, \cdots, K)$ is defined as $\widehat{MLA}_t^{(k,i)} = \sum_{j=1}^{M} \hat{\rho}_{t,j}^{(k,i)} \bar{z}_{t,j}^{(k,i)}/M$, where M is the number of bins, $\widehat{\rho}_{t,j}^{(k,i)}$ is the sample Pearson correlation coefficient of the LA pair in bin j when the scouting gene Z is X_i in triplet t, and $\overline{z}_{t,j}^{(k,i)}$ is the mean of Z in bin j. The test statistic for single study k is $T_{MLA,t}^{(k,i)} = \widehat{MLA}_t^{(k,i)}/SE(\widehat{MLA}_t^{(k,i)})$, where $SE(\widehat{MLA}_t^{(k,i)})$ is the standard error of $\widehat{MLA}_t^{(k,i)}$ for $k = 1, \cdots, K$ and $i = 1, 2, 3$. The MetaMLA statistic combines individual study MLA statistics $T_{MLA,t}^{(k,i)}$ and is defined as $mMLA_t^{(i)} = \overline{T}_{MLA,t}^{(i)}/(s_t^{(i)} + s_0)$ where $\overline{T}_{MLA,t}^{(i)}$ and $s_t^{(i)}$ are the sample mean and standard deviation of $\left\{T_{MLA,t}^{(k,i)}, k = 1, 2, \cdots, K\right\}$, respectively. $s_t^{(i)}$ provides standardization according

to the variance of MLA scores across studies. s_0 is a fudge parameter to avoid obtaining large mMLA score caused by very small $s_t^{(i)}$ values, which happens frequently in genome-wide screening. In our yeast data sets, suppose N is the total number of triplets for the hypothesis testing. We choose s_0 to be $10 \times med\{s_t^{(i)}, i = 1, 2, 3 \quad \text{and} \quad t = 1, \cdots, N\}$ (where $med(\cdot)$ means the median) to guarantee the stability of the test statistics, especially when sample size is small. The standardization by dividing the variance in the $T_{MLA,t}^{(k,i)}$ score considers both sample size and sample heterogeneity effects in single studies. For a study of large sample size, the standard deviation of MLA score is usually smaller and thus generates larger $T_{MLA,t}^{(k,i)}$ score. For a study containing large biological variation or considerable outliers in samples, the standard deviation of MLA score is large and results in smaller $T_{MLA,t}^{(k,i)}$ score.

The MetaLA statistic can be defined similarly with the MetaMLA statistic. The estimation of the LA score [Li, 2002] for the single study k ($k = 1, \cdots, K$) is defined as $\widehat{LA}_t^{(k)} = \sum_{l=1}^{n_k} X_{1l}^{(k)} X_{2l}^{(k)} X_{3l}^{(k)} / n_k$, where n_k is the total number of observations (samples) and $X_{1l}^{(k)}$, $X_{2l}^{(k)}$, and $X_{3l}^{(k)}$ are the lth observations for genes X_1, X_2, and X_3 in study k, respectively. The MetaLA statistic combines individual study LA scores $\widehat{LA}_t^{(k)}$ and is defined as $mLA_t = \overline{\widehat{LA}_t}^{(k)} / (s_t + s_0)$ where $\overline{\widehat{LA}_t}^{(k)}$ and s_t are the sample mean and standard deviation of $\left\{\widehat{LA}_t^{(k)}, k = 1, 2, \cdots, K\right\}$, respectively. s_t provides standardization according to the variance of LA scores across studies. s_0 is a fudge parameter to avoid obtaining large mLA score caused by very small s_t values.

3.2.4 Hypothesis testing and inference for MetaMLA and MetaLA

Based on MetaMLA, the hypothesis for liquid association in the gene triplet t: (X_1, X_2, X_3) is

$$H_0 : mMLA_t^{(i)} = 0, \forall i \in \{1, 2, 3\}$$

$$\leftrightarrow H_1 : \exists i \in \{1, 2, 3\}, s.t.\ MLA_t^{(i)} \neq 0,$$

where $i = 1, 2, 3$ corresponds to LA scouting gene $Z = X_i$ ($i = 1, 2, 3$). The null hypothesis represents all zero liquid associations no matter which one of X_1, X_2, and X_3 acts as the scouting gene Z. The test statistic is defined as

$$T_t = \max_{i=1,2,3} |mMLA_t^{(i)}|.$$

The distribution of T_t under the null hypothesis can be obtained by randomly permuting the samples of the LA scouting gene Z when calculating each $mMLA_t^{(i)}$ in the T_t statistics. We repeat the permutation for B times and use the resulting $B \times N$ permuted values of $T_t^{(b)}$ $(1 \leq b \leq B, 1 \leq t \leq N)$ as the null distribution. The P-value can be given by $P = (\sum_{b=1}^{B} \sum_{t=1}^{N} I(T_t^{(b)} \geq T_{obs})/(B \times N)$, where T_{obs} is the observed value of the test statistic. The P-values are corrected by Benjamini-Hochberg (BH) algorithm [Benjamini and Hochberg, 1995] and the false discovery rate is set to be $\alpha = 0.01$. Since the number of possible triplets N is usually very large, a small B is needed ($B = 40$) and used in the paper. We note that theoretically we should perform permutation for each triplet to form its own null distribution. The computation is, however, obviously not feasible (= number of permutations $\times$ number of triplets). In our approach, we imposed an assumption of common null distributions across all triplets to allow affordable computation.

Based on MetaLA, the hypothesis for liquid association in the gene triplet t: (X_1, X_2, X_3) is $H_0 : mLA_t = 0 \leftrightarrow H_1 : mLA_t \neq 0$. The test statistic can be defined as $T_t = |mLA_t|$. The distribution of T_t under the null hypothesis can be obtained by randomly permuting the samples inside gene X_1, X_2, or X_3 in turn. We repeat the permutation for B times and use the resulting $B \times N$ permuted values of $T_t^{(b)}$ $(1 \leq b \leq B, 1 \leq t \leq N)$ as the null distribution. The P-value can be given by $P = (\sum_{b=1}^{B} \sum_{t=1}^{N} I(T_t^{(b)} \geq T_{obs})/(B \times N)$, where T_{obs} is the observed value of the test statistic. The P-values are corrected by Benjamini-Hochberg (BH) algorithm [Benjamini and Hochberg, 1995] and the false discovery rate is set to be $\alpha = 0.01$. Similar to MetaMLA, $B = 40$ is used.

3.2.5 Filtering to reduce computation of MetaMLA

Genome-wide calculation of the liquid association is usually time-consuming and resource-intensive for a single study [Li, 2002; Ho et al., 2011]. This problem is further aggravated when combining multiple studies. In this section, we will develop a screening algorithm to perform a genome-wide MetaMLA analysis with higher efficiency. As illustrated in Fig. 3.2,

our algorithm seeks to reduce the number of triplets which need to be examined in depth in two screening steps: bootstrap filtering and sign filtering (Fig. 3.2).

In the first bootstrap filtering step, we filter out triplets with small correlation difference between the high and low bins. Define ρ_{diff} to be the difference of the LA pair correlations when scouting gene assigned to the highest and lowest bins. In the literature, the fastLA algorithm for single study [Gunderson and Ho, 2014] has used screening procedure for fast computing. In meta-analysis, we aim to detect triplets with consistently large or consistently small liquid associations across multiple studies. For the triplet t: (X_1, X_2, X_3), given the scouting gene $Z = X_i$ $(i = 1, 2, 3)$, we define $\rho^{(k,i)}_{diff,t} = \rho^{(k,i)}_{high,t} - \rho^{(k,i)}_{low,t}$, where $\rho^{(k,i)}_{high,t}$ and $\rho^{(k,i)}_{low,t}$ are the Pearson correlations when gene Z is in the high and low bins of study k, respectively. We use the score $\sum_{k=1}^{K} |\rho^{(k,i)}_{diff,t}|/K$ as the meta-filtering criteria. Since the scouting gene Z could be X_1, X_2, or X_3, we use $\max_{i=1,2,3}(\sum_{k=1}^{K} |\rho^{(k,i)}_{diff,t}|)/K$ to order and filter out triplets that are unlikely to have LA association. To avoid outlier effect when calculating correlations in the bins, we propose to bootstrap [Efron and Tibshirani, 1986] samples in each study for B times and get $\rho^{(meta,b)}_{diff,t} = \max_{i=1,2,3} \sum_{k=1}^{K} |\rho^{(k,i,b)}_{diff,t}|/K$, where $b = 1, 2, \cdots, B$. Finally, we can use

$$\rho^{(meta)}_{diff,t} = med\left(\rho^{(meta,b)}_{diff,t}, b = 1, \cdots, B\right)$$

to screen the triplets, where $med(\cdot)$ means taking the median. We set $\rho^{(meta)}_{diff,t} > 0.4$ as the cutoff to keep the triplets for further testing. $\rho^{(meta)}_{diff,t}$ can largely reduce computational complexity for two reasons: (1) calculating $\rho^{(meta)}_{diff,t}$ is computationally much simpler than the MetaMLA statistic; (2) $\rho^{(meta)}_{diff,t}$ can filter out a large percent of triplets and further reduce the computational cost of P-value calculation in the permutation step.

In the second sign filtering step, we filter out triplets with inconsistent signs of test statistics among meta and singleMLA. The scouting gene is chosen to maximize the test statistic of MetaMLA. In other words, we keep the triplets satisfying

$$\prod_{k=1}^{K} I\left(sign\left(mMLA^{(i_0)}_t\right) \cdot sign\left(T^{(k,i_0)}_{MLA,t}\right) = 1\right) = 1,$$

where $I(\cdot)$ is the indicator function and $i_0 = \arg\max_{i=1,2,3} |mMLA^{(i)}_t|$. For fair comparison, we use the same triplets filtered by MetaMLA to perform MetaLA and single-study MLA.

3.3 RESULTS

3.3.1 Computational reduction by filtering

Below we describe the screening result to avoid high computational load when evaluating all possible triplets in MetaMLA. After unbiased filtering of non-expressed and non-informative genes, we kept 1,770 genes, which led to a total number of $\begin{pmatrix} 1770 \\ 3 \end{pmatrix} \approx 9.23 \times 10^8$ triplets. The computing time is demanding if we perform hypothesis testing for all possible triplets. By applying bootstrap filtering with $\rho^{(meta)}_{diff,t} > 0.4$ with three bins, the number of triplets reduced to 2.18×10^7, approximately 2.36% of the original total number. Furthermore, the sign filtering step decreased the number of the remaining triplets to 1.21×10^7, which was only 1.32% of the total number.

Given the fact that our screening pipeline can dramatically reduce the number of triplets, we assessed whether the filtering procedures ignored statistically significant LA triplets. We performed MetaMLA on all the 9.23×10^8 triplets and reduced 1.32% triplets after filtering. As shown in Table B.1, our screening steps only missed 89, 219, 375, 520, and 690 of the top 2000, 4000, 6000, 8000, and 10000 triplets obtained from full analysis without filtering. P-values from Fisher's exact test are almost 0 and odds ratio are between 1,000-1,600 (Table B.1). In summary, we only missed about 5% significant triplets but saved almost 99% of computing time to make genome-wide LA triplet screening possible. Since filtering step also consumes computing time, we compared computing time of analyses with filtering versus non-filtering on a small dataset of 95 genes (using stringent selection criteria by removing genes with small means and small variances). By using five computing threads (Intel Xeon E7-2850), computing time for analyses with filtering versus non-filtering saved about 88% of computing time (16.3 minutes versus 134.6 minutes).

In general, filtering out potentially non-significant triplets will gain statistical power (van Iterson et al. [2010] and Bourgon et al. [2010]). In other words, we can detect more significant triplets under the same FDR control. To demonstrate the empirical effect of filtering in real data, we randomly selected 500 genes from the five Yeast studies and re-ran our MetaMLA algorithms by both filtering and non-filtering pipelines. Fig. B.1 shows that for a given

reasonable FDR (for example: 0.005 and 0.01), filtering pipeline can detect more significant triplets than full studies as we expected.

3.3.2 MetaMLA detects more over-represented pathways

We performed pathway enrichment analysis using GO and KEGG for all the genes from top m significant triplets ($m = 200, 300, \cdots, 1000$) selected by the single study MLA, MetaMLA, and MetaLA. Fig. 3.3 shows the numbers of enriched GO terms and KEGG pathways for different top numbers of triplets under FDR=0.05 threshold. MetaMLA (solid square line) consistently performed better than any single-study MLA (five dash lines) and MetaLA (solid rhombus line) method by detecting more enriched pathway. Jitter plots of q-values of the GO terms and KEGG pathways for the top 500 triplets at minus log 10 scale are further shown in Fig. B.2. Since single MLA and MetaMLA method can differentiate LA scouting gene Z, similar pathway enrichment analysis were done only for Z genes from the top triples (Fig. B.3 and B.4).

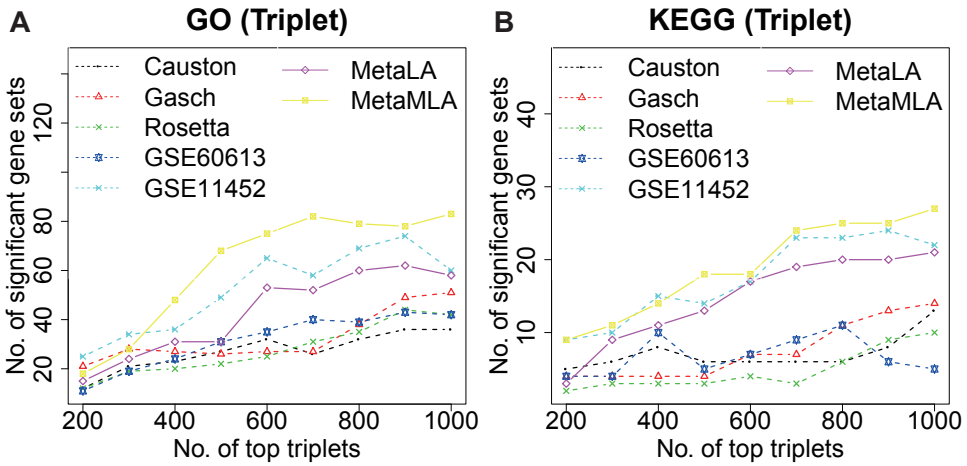


Figure 3.3: The number of enriched gene sets for all the genes from different numbers of top triplets detected by meta and single analysis. **(A)** is for GO terms and **(B)** is for KEGG pathways.

3.3.3 MetaMLA provides more consistent biomarker and pathway results with single study analyses

Fig. 3.1 (A) shows an example with LA association in the first study (correlation dropped from 0.692 to -0.79 for high and low expression groups of the LA scouting gene YGR175C) but fails to reproduce in the remaining four studies. Such an LA association with failed reproducibility is likely a false positive. Fig. 3.1 (B) demonstrates another example with consistent LA association in all five studies (correlation dropped significantly for high and low expression groups of YDR519W). In order to inspect agreement of top LA triplets across pairwise studies, Fig. B.5 and B.6 show scatter plots of test statistics and rank correlations of the pairwise top 1000 triplets. MetaMLA method combines information from all single studies. Conceptually, MetaMLA can provide more consistent results with single study MLA than results among single study MLA. In Fig. 3.4 (A), we examined pairwise overlap of detected top 1,000 triplets from the five single-study MLA and the MetaMLA. The result shows zero overlapping in all single-study MLA top triplets. (We also tried other top number of triplets in Fig. B.7 and they all shows out small overlap among single studies.) On the other hand, top triplets from MetaMLA have much higher percentage of overlapping with results from each single-study MLA.

We next calculated the number of overlaps of enriched GO terms and KEGG pathways when we used all the genes from the top 500 triplets from each MLA analysis for pathway enrichment. The results are shown in Fig. 3.4 (B) and (C). Numbers on the diagonal cells demonstrate the number of enriched GO or KEGG pathways from each single-study MLA and MetaMLA. (Similarly, overlapped pathways by only Z genes from the top 1000 triplets are shown in Fig. B.8). Similar to overlapped triplets in Fig. 3.4 (A), we observed much higher overlapped pathways between the MetaMLA result and each single-study MLA result than results between pair-wise single-study MLA. For example, study Causton detected 27 enriched GO terms, among which 8, 9, 6 and 9 pathways overlapped with results from the other four single-study MLA. Notably, it has 12 and 15 GO terms overlapped with MetaLA and MetaMLA. Comparing the two meta-analytic methods, MetaMLA performed much better than MetaLA.

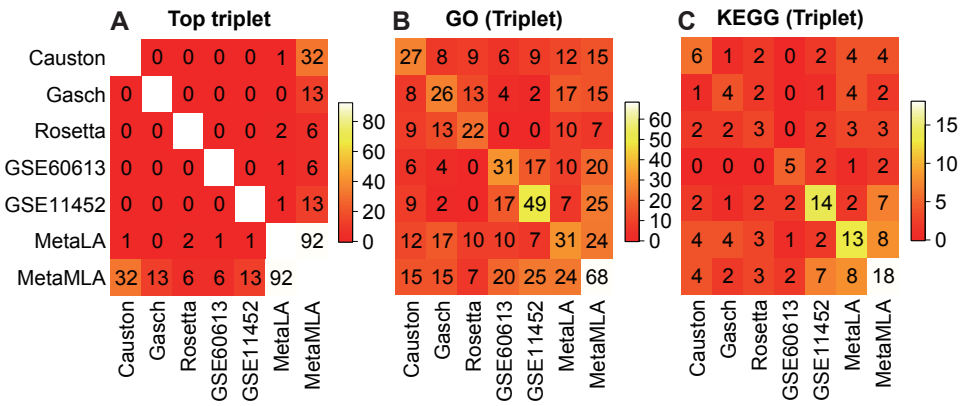


Figure 3.4: Overlap of meta and single analysis. **(A)** is for the number of overlapped triplets for the top 1000 significant triplets; **(B)** is for the number of overlapped enriched GO terms using all the genes from top 500 triplets for gene set enrichment analysis; **(C)** is for the number of overlapped enriched KEGG pathways using all the genes from top 500 triplets for gene set enrichment analysis.

3.3.4 MetaLA and MetaMLA provide more stable results

Below we apply subsampling and bootstrap techniques to compare stability for LA triplets detected by single-study MLA, MetaLA and MetaMLA. Fig. 3.5 (A) and (B) show the number of overlapped triplets between top triplets detected by original full dataset and subsampled data sets (90% and 80%, respectively). The numbers of top triplets are displayed on the x-axis and the y-axis is for the overlapping numbers. The result shows much better reproducibility of top triplets detected by subsampled data in MetaMLA (solid square line) and MetaLA (solid rhombus line) compared to single-study MLA (five dash lines). Similarly, comparison with bootstrapped data in Fig. 3.5 (C) shows similar trend. In summary, MetaMLA provides better stability in detecting top LA triplets, when compared to single-study MLA. MetaLA further outperforms MetaMLA.

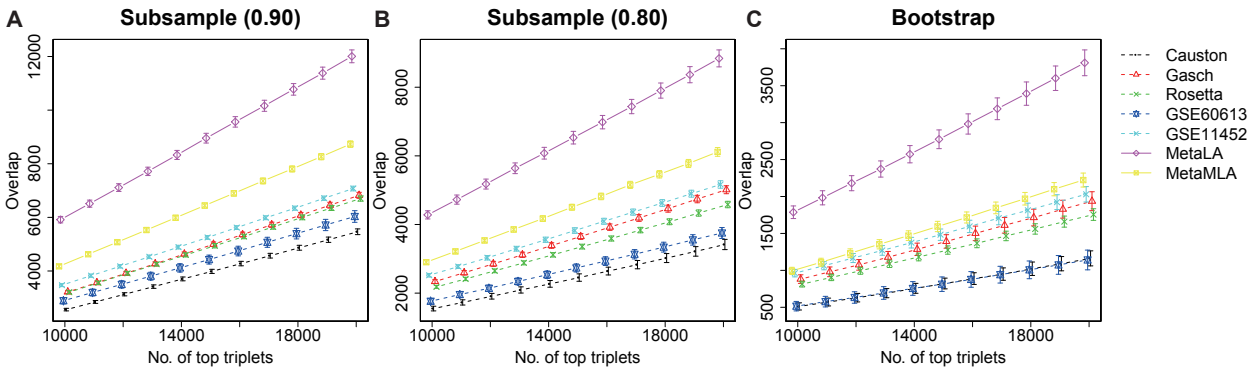


Figure 3.5: The number of overlapped top significant triplets between the original data set and the subsampled or bootstrap data sets. **(A)** and **(B)** are for the results of means and standard errors of ten times subsampling for the proportion of 0.90 and 0.80, respectively; **(C)** is for the results of means and standard errors of ten times bootstrap.

3.3.5 Pathway enrichment analysis and network visualization

In Section 3.3.2 - 3.3.4, although MetaLA provides more stable result than MetaMLA 3.3.4, it detects much fewer enriched pathways (Section 3.3.2) and generates less consistent biomarker and pathways with single studies (Section 3.3.3). As a result, we will focus on MetaMLA for further biological investigation in this subsection.

To test how the liquid association genes detected by MetaMLA method are consistent with transcription factor (TF) binding, we downloaded the TF binding gene sets from the YEASTRACT database [Teixeira et al., 2013] and selected 96 gene sets with 5-200 genes. Among these 96 TF genes, Hog1 (YLR113W) has the highest frequency among all the genes from the top 20000 triplets detected by MetaMLA method. Genes inside the same triplet as Hog1 are enriched in Hog1 binding gene sets ($p = 0.027$). More significantly, Hog1 is also the most frequent gene among the LA scouting gene Z in the top 100000 triplets. Genes regulated by Hog1 (inside the same triplets) are more significantly enriched in Hog1 binding gene sets ($p = 1.44E-5$). Table B.2 shows the top enriched TF binding gene sets. Among them, Hot1 is another enriched gene sets ($p = 7.67E-6$) and Alepuz et al. [2003] shows that Hot1 targets on Hop1p to osmostress responsive promoters and Hog1 mediates

recruitment/activation of RNAPII at Hot1p-dependent promoters. The analysis shows that top triplets selected by MetaMLA method are highly consistent with known TF regulation pattern.

Table 3.1 shows 18 significantly enriched KEGG pathways with hierarchical structure using all the genes from top 500 triplets selected by MetaMLA. Pathway enrichment using GO database identified 68 GO terms (Table B.3). Since the five transcriptomic studies contain yeast samples treated with different environmental conditions and mutations, we observed many pathways related to energy metabolism ($q = 5.67E-12$), carbohydrate, metabolism ($q = 1.40E-8$), amino acid metabolism ($q = 5.87E-8$), and translation ($q = 0.0065$).

Table 3.1: Enriched KEGG pathways and their hierarchical categories for all the genes from top 500 triplets selected by MetaMLA method.

Entry and category	*P*-value	*q*-value	Odds ratio	Count	Size	Name
Metabolism	1.53E-23	1.85E-21	2.69	200	835	
Energy metabolism	9.37E-14	5.67E-12	5.36	43	122	
sce00190	2.03E-12	8.21E-11	7.91	29	72	Oxidative phosphorylation
sce00680	0.007957	0.041109	3.49	8	28	Methane metabolism
Carbohydrate metabolism	4.64E-10	1.40E-08	2.86	62	229	
sce00620	1.91E-06	3.30E-05	5.53	17	39	Pyruvate metabolism
sce00630	3.15E-05	0.000423	6.14	12	26	Glyoxylate and dicarboxylate metabolism
sce00020	6.30E-05	0.000763	4.99	13	32	Citrate cycle (TCA cycle)
sce00010	0.000527	0.004249	2.99	17	58	Glycolysis / Gluconeogenesis
sce00051	0.005765	0.034881	3.76	8	25	Fructose and mannose metabolism
sce00030	0.007158	0.039369	3.23	9	28	Pentose phosphate pathway
Amino acid metabolism	2.42E-09	5.87E-08	3.06	51	178	
sce00260	6.40E-08	1.29E-06	8.99	16	32	Glycine, serine and threonine metabolism
sce00270	0.000146	0.001468	4.44	13	36	Cysteine and methionine metabolism
sce00250	0.002637	0.016791	3.60	10	30	Alanine, aspartate and glutamate metabolism
Lipid metabolism	0.000913	0.006501	2.11	29	126	
sce00100	0.000183	0.001705	6.89	9	17	Steroid biosynthesis
sce01040	0.000454	0.003927	12.21	6	11	Biosynthesis of unsaturated fatty acids
sce00062	0.002175	0.014623	10.16	5	8	Fatty acid elongation
Metabolism of cofactors and vitamins	0.011664	0.052272	1.80	24	117	
sce00670	0.008629	0.041763	4.57	6	15	One carbon pool by folate
Genetic Information Processing	0.214481	0.447452	1.10	114	1123	
Translation	0.000861	0.006501	1.59	70	682	
sce03010	9.42E-05	0.001036	2.22	37	181	Ribosome
Folding, sorting and degradation	0.105526	0.283748	1.27	41	263	
sce03050	0.008154	0.041109	2.91	10	35	Proteasome
Cellular Processes	0.718769	0.995721	0.92	45	382	
Transport and catabolism	0.010656	0.049591	1.61	36	194	
sce04145	2.96E-05	0.000423	5.07	14	36	Phagosome

To investigate further the identified LA association gene interactions, we chose among

the top 20,000 LA triplets ($q < 6.64E-5$) and included a total of 41 triplets with all three genes involved in the metabolism category ($q = 1.85E-21$ in Table 3.1) for network visualization (Fig. 3.6). Genes within one triplet are connected by edges in the same color. The dashed line represents reported interactions or regulations in the PPI database. In this network, there are totally four interactions validated by PPI database, more enriched than a randomly generated PPI database (0.69 random interactions on average, with P-value 0.00197 by Fisher's exact test).

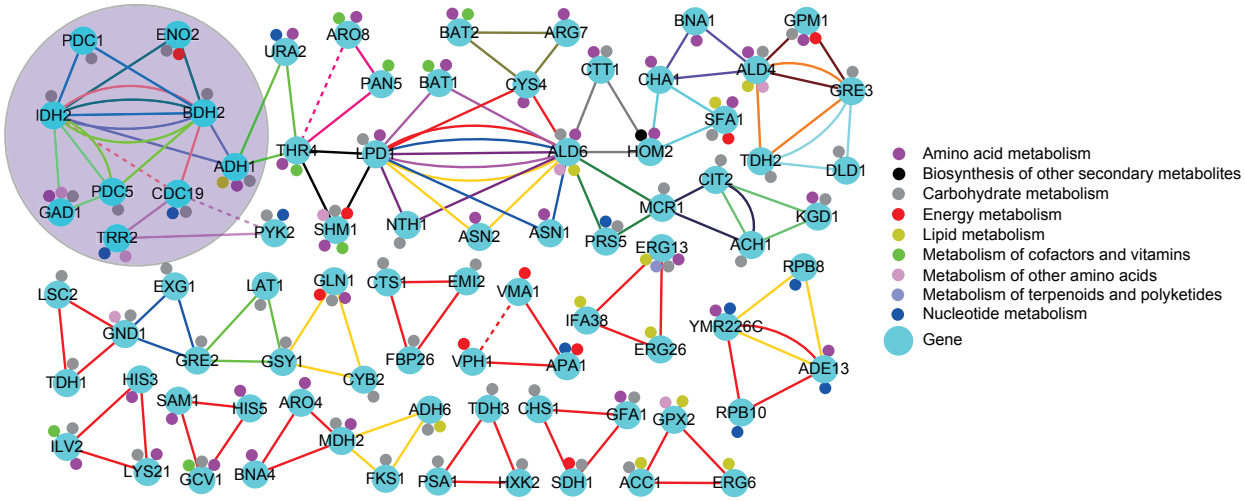


Figure 3.6: Gene network associated with metabolism. Genes within one triplet are connected by edges in the same color. The dash line means that the edge is in the PPI database. The small circle connected with the gene means that this gene is in the corresponding sub-category.

In Fig. 3.6, we observed a cluster of gene modules related to carbohydrate metabolism (purple background circle in Fig. 3.6; almost all genes annotated with gray dots). IDH2 and BDH2 are two notable hub genes that have many LA association with other neighboring genes. IDH2 is a subunit of mitochondrial NAD(+)-dependent isocitrate dehydrogenase, a key complex in tricarboxylic acid (TCA) cycle to catalyze the oxidation of isocitrate to alpha-ketoglutarate [Reinders et al., 2007]. BDH2 is a putative medium-cahin alcohol dehydrogenase [Dickinson et al., 2003]. In carbohydrate metabolism, pyruvate is the main input

for a series of chemical reactions for aerobic TCA cycle. In the subnetwork, CDC19 is a key pyruvate kinase, which coverts phosphoenolpyruvate to pyruvate [Xu et al., 2012; Byrne and Wolfe, 2005], and its physical protein-protein interaction with IDH2 has been previously reported [Gavin et al., 2006]. In addition, PDC5 is a minor isoform of pyruvate decarboxylase and PDC1 is a major of three pyruvate decarboxylase isozymes to decarboxylate pyruvate to acetaldehyde [Dickinson et al., 2003]. ENO2 is also a phosphopyruvate hydratase involved in pyruvate metabolism to catalyze 2-phosphoglycerate to phosphoenolpyruvate during glycolysis [McAlister and Holland, 1982; Byrne and Wolfe, 2005]. All these genes from the top MetaMLA triplets are potentially co-regulated with functional annotation from the carbohydrate metabolism pathway. However, if we examine the direct gene-gene Pearson correlations, the pair-wise correlations are low and the co-expression analysis will fail to identify association among these genes (see Table B.4).

3.4 CONCLUSION AND DISCUSSION

In this paper, we proposed two meta-analytic methods (MetaLA and MetaMLA) for liquid association analysis combining multiple studies. We used the mean of the singleMLA test statistics as the main part of the MetaMLA statistic and the standard deviation to penalize the inconsistent patterns among different studies. On the genome-wide application, we proposed to screen genes by bootstrap filtering and sign filtering (Fig. 3.2) to reduce the computation load. In the yeast data sets, we reduced more than 98% of the triplets for the hypothesis testing and captured 94-95% of the top triplets with large MetaMLA statistic. Compared with singleMLA method, MetaMLA can provide stronger pathway enrichment signal, more consistent results with single-study analysis, and more stable results with data subsampling or bootstrapping. Although MetaLA generated more stable results than MetaMLA, it detected less enriched pathways and is less consistent with single study analysis. Among the top significant triplets selected by MetaMLA, we constructed a gene regulatory network visualization to investigate the complex three-way conditional associations. The result identifies a subnetwork in carbohydrate metabolism network, which otherwise

cannot be identified by traditional pair-wise co-expression analysis. We identified validation in protein-protein interaction and focused functional annotation in TSA cycle.

The LA and MLA methods to detect liquid association triplets have their pros and cons. On one hand, the LA score by a three-product estimation on normalized gene intensities is much easier to compute than the model-free estimation of MLA score. However, MLA is more accurate when interdependency among the triplet (i.e. conditional mean and variance of two genes depend on the third gene) exist and such interdependency is theoretically ignored by the LA method. Additionally, MLA also provides systematic inference to assess p-values and false discovery rate control. To circumvent computational burden in MetaMLA, our proposed two-stage filtering can significantly reduce computing time. In this paper, we demonstrate genome-wide screening on all possible gene triplets. To further reduce computational load, one may apply pre-selected scouting genes from prior biological knowledge, transcription factor or protein-protein interaction databases.

Our meta-analytic framework has the advantage to stablely combine multiple studies from different microarray or next-generation sequencing platforms. Potential heterogeneity from platform, batch effect or measurement scaling issues are automatically standardized in the meta-analysis. In the literature, it has been well-acknowledged that simple correlation and co-expression analysis are not sufficient to describe the complex system of gene regulation. Applying advanced association models elucidates novel regulatory mechanisms and meta-analysis by combining multiple transcriptomic studies will greatly reduce false positive findings. Our proposed meta-analytic liquid association methods help accurately detect complicated three-way interactions and regulatory mechanisms.

3.5 ACKNOWLEDGEMENT

This work was supported by the National Institutes of Health NIH [R01CA190766 to S.L. and G.C.T.]; China Scholarship Council [201508110051 to L.W.]; National Nature Science Foundation of China [11526146 to L.W.]; Scientific Research Level Improvement Quota Project of Capital University of Economics an Business [to L.W.]; and University of Minnesota

Grant-In-Aid [to Y.Y.H.].

4.0 AIM 3. META-ANALYTIC PLAID MODEL FOR DETECTING BICLUSTERS WHEN COMBINING MULTIPLE TRANSCRIPTOMIC STUDIES

The majority of the text in this chapter comes from a manuscript prepared for submission.

4.1 INTRODUCTION

In the past two decades, high-throughput experimental technologies including microarray and next-generation sequencing have generated abundant high-dimensional data and introduced new statistical and computational challenges. In the analysis of transcriptomic data from microarray and RNA-seq, cluster analysis is a powerful unsupervised machine learning tool to group objects (i.e. genes or patients) by proximity of expression patterns when the underlying true class labels are not known [D'haeseleer, 2005]. When clustering genes, the purpose is to identify modules of highly co-expressed genes that likely are co-regulated or share common biological functions. When clustering patients, we aim to identify patient groups of similar expression patterns to form disease subtypes that are potentially of clinical significance with different disease mechanism, treatment response or survival outcome. Many classical clustering methods as well as new inventions have been applied or developed to the cluster analysis of expression profiles. Methods such as hierarchical clustering [Eisen et al., 1998], K-means [Hartigan and Wong, 1979], self-organizing map [Tamayo et al., 1999], model-based approaches [McLachlan et al., 2002] and Bayesian clustering [Medvedovic et al., 2004] have been widely used. Resampling approaches have been applied to improve clustering stability and consistency [Monti et al., 2003; Tibshirani and Walther, 2005; Tseng and Wong, 2005]. The concept of excluding scattered genes or samples from cluster assignment was introduced

to improve clustering tightness and quality [Tseng and Wong, 2005; Tseng, 2007; Maitra and Ramler, 2009]. Several comparative studies have been conducted to evaluate performance of different clustering methods in microarray data (see Datta and Datta [2003] and Thalamuthu et al. [2006] for gene clustering and de Souto et al. [2008] for sample clustering).

The aforementioned "one-way" clustering, however, has its own drawbacks. In most complex diseases, patients are heterogeneous and unknown disease subtypes often exist. Many gene modules may be co-expressed only in a subset of samples. To this end, biclustering methods have been developed for identifying clusters with subset of genes and subset of samples simultaneously. A variety of biclustering methods have been developed in the literature. Cheng and Church [2000] was the earliest to employ biclustering in gene expression data. They used a greedy algorithm to find biclusters assuming uniformly expressed intensities in the background. Lazzeroni and Owen [2002] introduced a plaid model on the assumption that the matrix expression levels are the superposition of each bicluster layers. Martella and Vermunt [2013] developed a mixture of structural equation models (SEMs), which is a more general model to detect both clustering and biclustering patterns in gene expression matrices [Martella et al., 2008]. Among many other popular algorithms, Spectral [Kluger et al., 2003], xMOTIFs [Murali and Kasif, 2003] and Bayesian Biclustering [Gu and Liu, 2008] were developed with different bicluster pattern targets and for different biological purposes. Multiple reviews and comparative studies have comprehensively evaluated different biclustering methods [Eren et al., 2013; Bozdağ et al., 2010; Madeira and Oliveira, 2004; Prelić et al., 2006].

As microarray and NGS experimental costs continue to drop over the years, tremendous amount of transcriptomic data are available in public databases. As each single study usually contain only moderate number of samples, its cluster analysis is deemed unstable and may be biased by study-specific features such as cohort bias or experimental protocol. An increasing trend of combining multiple transcriptomic studies for meta-analysis not only increases statistical power but also generates more robust and consistent results that more likely can be validated in independent cohorts [Tseng et al., 2012; Richardson et al., 2016]. In the literature, transcriptomic meta-analysis have mostly focused on detecting differentially expressed genes and associated pathways. In this paper, we propose a meta-analytic plaid

model to integrate multiple transcriptomic studies for bicluster detection. From simulations and a real example of five breast cancer expression profiles (Table 4.1), we show that single study biclustering generates unstable results that are difficult to validate in independent studies. Our meta-biclustering method produces more stable and consistent results that reveal interesting biological insights in potential disease subtypes.

Table 4.1: Five breast cancer expression data information

BRCA sample	GSE2034	GSE7390	GSE11121	TCGA	METABRIC
Reference	Wang et al. [2005]	Desmedt et al. [2007]	Schmidt et al. [2008]	Network et al. [2012]	Curtis et al. [2012]
Platform	Affymetrix	Affymetrix	Affymetrix	Agilent	Illumina
Number of genes	12704	12704	12704	17814	19396
Number of samples	260	164	161	533	1981
Mean intensity	6.797 ± 1.71	5.523 ± 1.84	6.552 ± 1.79	0.003 ± 1.34	6.960 ± 1.70

4.2 METHODS

4.2.1 Data sets and databases

In order to evaluate the performance of biclustering, we generated simulation data with known underlying truth. First of all, we confined our study to four types of genes. They are (1) *consistent genes*: genes that are relevant to biclusters of all the studies; (2) *prevalent genes*: genes that are relevant to biclusters in all the studies except for the last one; (3) *study-specific genes*: genes that are relevant to biclusters in only one study; (4) *irrelevant genes*: genes that are irrelevant to biclusters in all studies. For each simulation setting, we synthesized 4 studies with 500 genes and 100, 100, 80 and 70 samples, respectively. The simulated biclusters are a superposition to the background value with 50 genes and 40, 30, 25 and 25 samples for each study. Table C.1 shows information of simulation parameters and details.

Five breast cancer (BRCA) data sets in Table 4.1 will be used to test our meta-analytic method: GSE2034 [Wang et al., 2005], GSE7390 [Desmedt et al., 2007], GSE11121 [Schmidt et al., 2008], TCGA [Network et al., 2012] and METABRIC [Curtis et al., 2012]. These

five studies were generated by different platforms and their intensity ranges differ from each other. It is worth noting that METABRIC, among all the data sets has access to rich clinical and survival information, which is particularly helpful for our evaluation of detected biclusters. For pre-processing, we filtered out genes with missing values and only kept the overlapped genes in all five studies. Then we quantile normalized the samples within each single study [Amaratunga and Cabrera, 2001]. For the next step, we performed meta-filtering method used in Wang et al. [2012b] and filtered out non-expressed genes (lowest 50% of mean expression) and non-informative genes (lowest 50% of expression variances). As a result, our data sets include 2648 genes across all the studies and 260, 164, 161, 533 and 1981 samples for the five studies, respectively.

4.2.2 The plaid model for single study bicluster detection

Plaid model is a popular and good-performing bicluster detection algorithm to detect overlapped biclusters from transcriptomic data [Lazzeroni and Owen, 2002] . It can discover higher proportion of enriched biclusters compared with other algorithms [Eren et al., 2013]. Assume Y_{ij} is the expression intensity of gene i in sample j, where $i = 1, \ldots, n$, $j = 1, \ldots, p$, n and p are the total numbers of genes and samples. In plaid model, it assumes that the expression intensities are the superposition of K layers. Mathematically,

$$\begin{aligned} Y_{ij} =& \theta_{ij0} + \sum_{k=1}^{K} \rho_{ik}\kappa_{jk}\theta_{ijk} + \epsilon_{ij} \\ =& (\mu_0 + \alpha_{i0} + \beta_{j0}) + \sum_{k=1}^{K} \rho_{ik}\kappa_{jk}(\mu_k + \alpha_{ik} + \beta_{jk}) + \epsilon_{ij}, \end{aligned}$$

where k is the layer index and $k = 0$ refers to the overall background layer; the formula $\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$ represents estimated intensity to be the sum of mean (μ_k), gene (α_{ik}) and sample (β_{jk}) effects for gene i in sample j of the kth layer; ρ_{ik} and κ_{jk} are binary membership partition parameters for gene i and sample j respectively, with 1 indicating inside the layer and 0 for outside the layer; ϵ_{ij} refers to the error term.

To estimate the parameters of the kth layer, the algorithm first subtracts the previous $(k-1)$ estimated layers. The remaining expression intensity can be written as, $Z_{ijk} =$

$Y_{ij} - \theta_{ij0} - \sum_{k'=1}^{k-1} \theta_{ijk'}\rho_{ik'}\kappa_{jk'}$. Then the objective function for the k^{th} layer is expressed as,

$$Q = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left[Z_{ij} - (\mu + \alpha_i + \beta_j)\rho_i\kappa_j\right]^2,$$

subject to normalizing conditions

$$\sum_{i=1}^{n}\rho_i^2\alpha_i = \sum_{j=1}^{p}\kappa_j^2\beta_j = 0,$$

where the subscript k is eliminated for short.

Lazzeroni and Owen [2002] proposed this plaid model and estimated the parameters by straightforward Lagrange multiplier. As a further step, Turner et al. [2005a,b] improved this model and implemented binary least square algorithm to update the parameters. Then they further pruned the biclusters to screen out non-informative genes or samples by pruning parameter ν_1 and ν_2,

$$\begin{aligned}
\tilde{\rho}_i &= \begin{cases} 1 & \text{if } \hat{\rho}_i = 1 \text{ and } \sum_{j:\hat{\kappa}_j=1}(\hat{Z}_{ij} - \hat{\theta}_{ij})^2 < (1-\nu_1)\sum_{j:\hat{\kappa}_j=1}\hat{Z}_{ij}^2, \\ 0 & \text{otherwise,} \end{cases} \\
\tilde{\kappa}_j &= \begin{cases} 1 & \text{if } \hat{\kappa}_j = 1 \text{ and } \sum_{i:\hat{\rho}_i=1}(\hat{Z}_{ij} - \hat{\theta}_{ij})^2 < (1-\nu_2)\sum_{i:\hat{\rho}_i=1}\hat{Z}_{ij}^2, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned} \tag{4.1}$$

Plaid model performs well in detecting biclusters within single study. In this paper, we propose a meta-biclustering method to extend this model to meta-analytic framework to detect biclusters among multiple cohorts simultaneously.

4.2.3 Penalized objective function for regularization and meta-biclustering

First of all, we improved the plaid model by integrating the pruning step into the objective function from a regularization and feature selection perspective. Consider the penalized objective function,

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{p}\left[Z_{ij} - \theta_{ij}\rho_i\kappa_j\right]^2 + \tau_1\sum_{i=1}^{n}\left[\rho_i\left(\sum_{j=1}^{p} Z_{ij}^2\right)\right] + \tau_2\sum_{j=1}^{p}\left[\kappa_j\left(\sum_{i=1}^{n} Z_{ij}^2\right)\right]. \quad (4.2)$$

We modify the pruning step in Eq. 4.1 from Turner et al. [2005a,b] by substituting $j : \hat{\kappa}_j = 1$ to $j \in \{1, 2, \ldots, p\}$, that is,

$$\tilde{\rho}_i = \begin{cases} 1 & \text{if } \hat{\rho}_i = 1 \text{ and } \sum_{j=1}^{p}(\hat{Z}_{ij} - \hat{\theta}_{ij})^2 < (1-\nu_1)\sum_{j=1}^{p}\hat{Z}_{ij}^2, \\ 0 & \text{otherwise,} \end{cases}$$
$$\tilde{\kappa}_j = \begin{cases} 1 & \text{if } \hat{\kappa}_j = 1 \text{ and } \sum_{i=1}^{n}(\hat{Z}_{ij} - \hat{\theta}_{ij})^2 < (1-\nu_2)\sum_{i=1}^{n}\hat{Z}_{ij}^2, \\ 0 & \text{otherwise,} \end{cases} \quad (4.3)$$

It can be proved that the penalized objective function in Eq. 4.2 is equivalent to the modified form of the pruning steps in Equ. 4.3 (see Appendix for proof).

Next, we extend the penalized objective function in Eq. 4.2 into a meta-analytic framework. Specifically, we consider the following objective function:

$$Q = \sum_{s=1}^{S}\sum_{i=1}^{n}\sum_{j=1}^{p^{(s)}}\left[Z_{ij}^{(s)} - \theta_{ij}^{(s)}\rho_i^{(s)}\kappa_j^{(s)}\right]^2 + \tau_1\sum_{s=1}^{S}\sum_{i=1}^{n}\left[\rho_i^{(s)}\left(\sum_{j=1}^{p^{(s)}} Z_{ij}^{(s)2}\right)\right]$$
$$+ \tau_2\sum_{s=1}^{S}\sum_{j=1}^{p^{(s)}}\left[\kappa_j^{(s)}\left(\sum_{i=1}^{n} Z_{ij}^{(s)2}\right)\right] + \lambda\cdot\Omega. \quad (4.4)$$

Here Ω is a meta-term to borrow information across studies and control consistency of gene selection and/or gene effect sizes across studies. λ is a tuning parameter for weight of the meta-term where larger λ gives higher penalty to the inconsistent gene partitions among multiple studies, and vice versa. In this paper, we have compared seven options of the meta-term Ω (see Table 4.2) and compared their performance in Section 4.3.1 using extensive simulations.

Table 4.2: Details of seven proposed meta-terms Ω.

No.	Formula	Note
1	$\Omega = \sum_{i=1}^{n} \sum_{s:\rho_i^{(s)} \neq 0} \left[\alpha_i^{(s)} \rho_i^{(s)} - \frac{1}{S} \sum_{s':\rho_i^{(s')} \neq 0} \alpha_i^{(s')} \rho_i^{(s')} \right]^2$	variance of $\alpha_i^{(s)} \rho_i^{(s)}$ for all the non-zero $\rho_i^{(s)}$
2	$\Omega = \sum_{i=1}^{n} \sum_{s=1}^{S} \left[\alpha_i^{(s)} \rho_i^{(s)} - \frac{1}{S} \sum_{s'=1}^{S} \alpha_i^{(s')} \rho_i^{(s')} \right]^2$	variance of $\alpha_i^{(s)} \rho_i^{(s)}$ for all the studies
3	$\Omega = \sum_{i=1}^{n} \left[\text{MAD}_{s:\rho_i^{(s)} \neq 0} \left(\alpha_i^{(s)} \rho_i^{(s)} \right) \right]$	MAD of $\alpha_i^{(s)} \rho_i^{(s)}$ for all the non-zero $\rho_i^{(s)}$
4	$\Omega = \sum_{i=1}^{n} \left[\text{MAD}_{s=1:S} \left(\alpha_i^{(s)} \rho_i^{(s)} \right) \right]$	MAD of $\alpha_i^{(s)} \rho_i^{(s)}$ for all the studies
5	$\Omega = \sum_{i=1}^{n} \sum_{s:\rho_i^{(s)} \neq 0} \left[sign\left(\alpha_i^{(s)} \right) \rho_i^{(s)} - \frac{1}{S} \sum_{s':\rho_i^{(s')} \neq 0} sign\left(\alpha_i^{(s')} \right) \rho_i^{(s')} \right]^2$	variance of $sign(\alpha_i^{(s)}) \rho_i^{(s)}$ for all the non-zero $\rho_i^{(s)}$
6	$\Omega = \sum_{i=1}^{n} \sum_{s=1}^{S} \left[sign\left(\alpha_i^{(s)} \right) \rho_i^{(s)} - \frac{1}{S} \sum_{s'=1}^{S} sign\left(\alpha_i^{(s')} \right) \rho_i^{(s')} \right]^2$	variance of $sign(\alpha_i^{(s)}) \rho_i^{(s)}$ for all the studies
7	$\Omega = \sum_{i=1}^{n} \sum_{s=1}^{S} \left[\rho_i^{(s)} - \frac{1}{S} \sum_{s'=1}^{S} \rho_i^{(s')} \right]^2$	variance of $\rho_i^{(s)}$ for all the studies

4.2.4 Optimization of proposed objective function

Parameters of the proposed MetaBiclust objective function (Eq. 4.4) can be estimated by using binary least square algorithm, which is similar to the single-study plaid model proposed by Turner et al. [2005a,b].

Step 1: Pre-processing. Normalize each study matrix to mean 0 and standard deviation 1. Mathematically, $Z^{(s,normalized)} = \frac{Z^{(s,raw)} - mean(Z^{(s,raw)})}{SD(Z^{(s,raw)})}$.

Step 2: Initialization. Implement K-means clustering with $k = 2$ and select all the genes (or sample) in the smaller cluster for gene (or sample) initialization.

Step 3: Expectation-maximization (EM) iteration. Estimate the parameters by EM algorithm, regarding the gene and sample selection as a missing data problem. The objective function is first optimized within each single cohort ($\tau_1 = \tau_2 = \lambda = 0$), then optimized without gene and sample selection penalization ($\tau_1 = \tau_2 = 0$), and finally updated using the full objective function. The parameters can be updated iteratively as follows,

M-step:

$$
\begin{aligned}
Z^{(s)*} &= Z^{(s)}[\rho_i == 1, \kappa_j == 1], \\
\mu^{(s)} &= mean(Z^{(s)*}), \\
\alpha_i^{(s)} &= mean(Z_{i.}^{(s)*}) - \mu^{(s)}, \\
\beta_j^{(s)} &= mean(Z_{.j}^{(s)*}) - \mu^{(s)}.
\end{aligned}
$$

E-step:

$$\{\rho_i^{(1)}, \ldots, \rho_i^{(s)}\} = \arg \min_{\{\rho_i^{(1)}, \ldots, \rho_i^{(s)}\} \in \{0,1\}^S} \sum_{s=1}^{S} \sum_{j=1}^{p^{(s)}} \left[Z_{ij}^{(s)} - \theta_{ij}^{(s)} \rho_i^{(s)} \kappa_j^{(s)} \right]^2 + \tau_1 \sum_{s=1}^{S} \left[\rho_i^{(s)} \left(\sum_{j=1}^{p^{(s)}} Z_{ij}^{(s)2} \right) \right] + \lambda \cdot \Omega,$$

$$\kappa_j^{(s)} = I \left[\sum_{i=1}^{n} (Z_{ij}^{(s)} - \rho_i \theta_{ij})^2 + \tau_2 \kappa_j^{(s)} \left(\sum_{i=1}^{n} Z_{ij}^{(s)2} \right) < \sum_{i=1}^{n} \left(Z_{ij}^{(s)} \right)^2 \right].$$

That is to say, main, gene and sample effect can be updated by the linear regression model (M-step), and partition parameters ρ and κ are updated by selecting the best binary value combination that can minimize the objective (E-step). We update $(\rho_i^{(s)}, \kappa_j^{(s)})$ and $(\mu^{(s)}, \alpha_i^{(s)}, \beta_j^{(s)})$ iteratively until convergence or reaching the maximum updating step cutoff. Multiple initials for K-means in Step 2. are tested to generate the best objective score and to alleviate local minimum problem.

4.2.5 Selection of parameters

In this section, we discuss selection of τ_1, τ_2 and λ. τ_1 and τ_2 are two pruning parameters that can be selected by maximizing the gap statistic, borrowing the idea from cluster number selection [Tibshirani et al., 2001]. We define,

$$\max_{(\tau_1, \tau_2)} Gap(\tau_1, \tau_2) = E_{\text{null}}[obj(\tau_1, \tau_2)] - obj(\tau_1, \tau_2),$$

where $obj(\tau_1, \tau_2)$ is the objective function score given τ_1 and τ_2 in Eq. 4.4. The null expected objective score $E_{\text{null}}[obj(\tau_1, \tau_2)]$ is calculated as the average of B times permutations under null. The algorithm performs grid search of τ_1 and τ_2 by balancing computing and resolution, and finally identifies the best $(\hat{\tau}_1\ \hat{\tau}_2)$ that maximizes the gap statistic.

Parameter λ helps to adjust the consistency of the genes selection from multiple studies. High λ means large penalty on the inconsistency, while low λ gives less penalty. To the extreme, $\lambda = \infty$ can guarantee exactly the same gene selection, while $\lambda = 0$ represents single study analysis with equal feature selection parameters τ_1 and τ_2 in each study.

4.2.6 Bicluster evaluation

We evaluate biclustering results from two aspects. For synthetic data with known underlying truth, we use Jaccard coefficient [Eren et al., 2013] to define the similarity between the detected bicluster b_1 and the true bicluster b_2,

$$s(b_1, b_2) = \frac{|b_1 \bigcap b_2|}{|b_1 \bigcup b_2|},$$

where $|b_1 \bigcap b_2|$ and $|b_1 \bigcup b_2|$ represents element numbers of the intersection and union of the two biclusters, respectively.

For real data without known biclustering truth, we implemented indirect evaluation for both gene and sample selection. We apply pathway enrichment analysis on curated pathway databases collected by MSigDB (version 5.2), including: chemical and genetic perturbations, Canonical pathways, BioCarta, KEGG and Reactome. We filter out pathways with less than 5 or greater than 200 genes. Finally, 3384 gene sets are used for pathway enrichment analysis by Fisher's exact test [Upton, 1992]. Enrichment p-values are then adjusted by Benjamini-Hochberg (BH) algorithm [Benjamini and Hochberg, 1995] and the false discovery rate (FDR) is set to be 0.05. For METABRIC study [Curtis et al., 2012], both clinical and survival information are available for cluster evaluation. For a given bicluster, we test association of sample selection with with clinical information (e.g. ER positive versus negative) by Fisher's exact test. Kaplan Meier curve and log-rank test [Mantel, 1966] are used in the METABRIC cohort for testing survival difference between samples inside and outside the bicluster.

4.3 RESULTS

4.3.1 Selection of meta-term Ω

For the MetaBiclust algorithm, we expect to select similar bicluster gene sets among multiple cohorts. Seven meta-terms (Table 4.2) were proposed in the objective function to penalize inconsistent bicluster gene selection. Fig. 4.1 compares the performance of different meta-terms in two scenarios (with pruning parameter $\tau_1 = \tau_2 = 0.05$). In the consistent +

prevalent gene simulation (Fig. 4.1A, see simulation details in section 4.2.1), three types of gene sets can be used as truth to evaluate the Jaccard similarity: truth 1 only contains the simulated consistent genes; truth 2 consists of the true consistent and prevalent gens for each study; and truth 3 includes both consistent and prevalent genes for all the studies. In this scenario, we aimed to detect biclusters composed by both consistent and prevalent genes in all the cohorts, so truth 1 and truth 3 are preferable. Fig. 4.1 shows that the 2^{nd} and 7^{th} meta-term overall performs better than the others. In the second scenario, consistent + study-specific genes were simulated to evaluate the performance (Fig. 4.1B). Similar to previous simulation, three types of gene set selections can play roles as underlying truth, except that now we substitute the prevalent genes by study-specific genes. In order to detect consistent gene sets but not influenced by study-specific genes, we prefer to compare the results using truth 1. Fig. 4.1 shows that the 7^{th} meta-term performs best for all different λ settings. Alternatively, Fig. C.1 compared the results without pruning steps ($\tau_1 = \tau_2 = 0$) and found similar performance trend. These two simulation results lead to our final decision of using the 7^{th} meta-term ($\sum_{i=1}^{n}\sum_{s=1}^{S}\left[\rho_i^{(s)} - \frac{1}{S}\sum_{s'=1}^{S}\rho_i^{(s')}\right]^2$) in the meta-biclustering objective function to encourage consistent and prevalent genes but discourage study-specific genes.

4.3.2 Meta-analysis increases bicluster detection accuracy

In order to compare bicluster detection performance between single-study and meta-analysis, we simulated 4 studies with consistent bicluster gene partition (described in section 4.2.1) and tried a series of λ value to evaluate the accuracy. When λ equals to zero, each single study detected their biclusters independently without meta-term penalty. Small λ value penalizes less on the inconsistent bicluster gene selection across studies, while larg λ value leads to more consistent gene selection results. Fig. 4.2A shows the Jaccard value of the biclusters detected from 4 simulation data matrices when given different λ values (0, 50, 100, 1000 and 10000). The mean and SD Jaccard values were calculated by 100-times simulation repeats. The x-axis σ_0 represents the background noise. Given that the true biclusters are only composed by consistent gene sets, Fig. 4.2A shows that meta-analysis

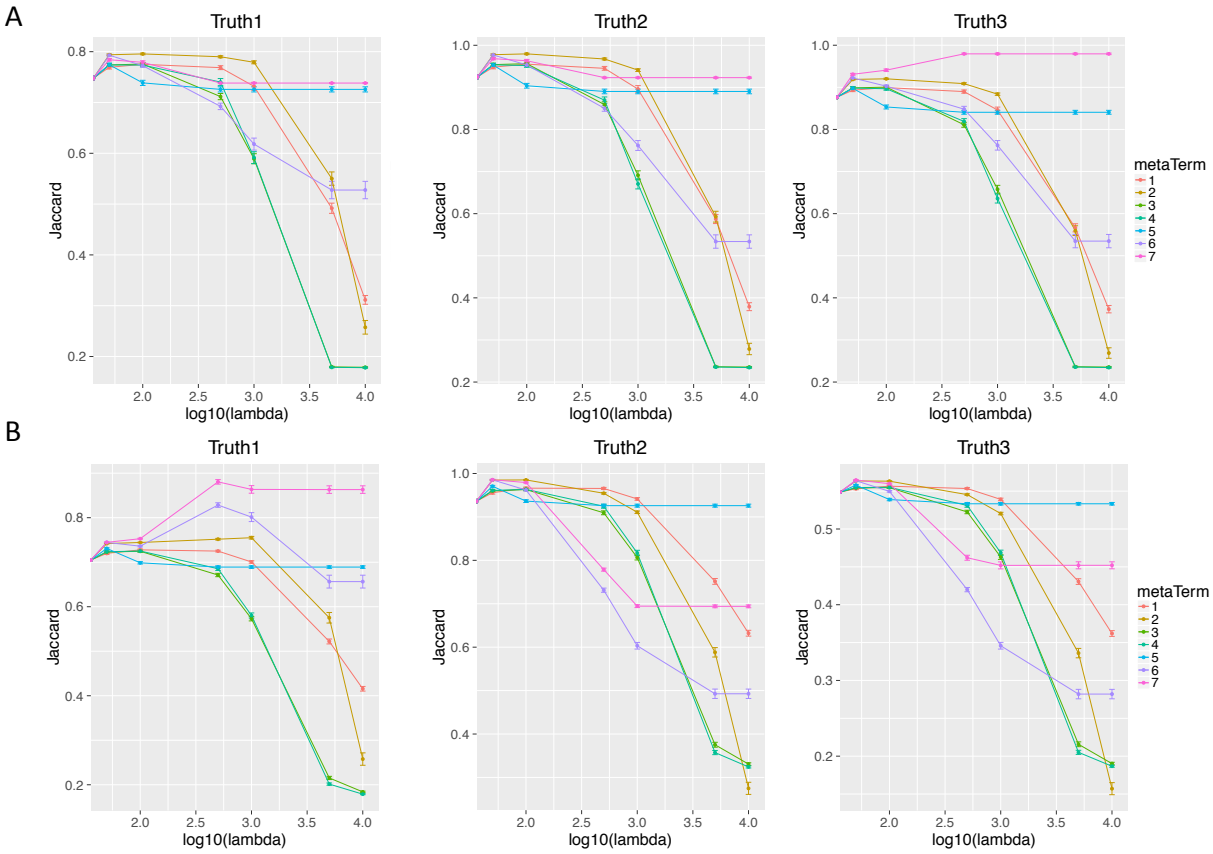


Figure 4.1: Comparison of seven meta terms performance on (A) consistent + prevalent gene simulation; (B) consistent + study-specific gene simulation with pruning steps. For consistent + prevalent gene simulation, truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and prevalent genes inside each study, and truth 3 means using consistent + prevalent genes as the truth for all studies. For meta analysis, truth 3 is preferred that both consistent and prevalent genes are preferred to be detected. For consistent + study-specific gene simulation, truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and study-specific genes inside each study, and truth 3 means using consistent + study-specific genes as the truth for all studies. For meta analysis, truth 1 is preferred that only common genes are preferred but not the study-specific genes.

($\lambda > 0$) overall performed better than single-study ($\lambda = 0$), especially when the noise was large. Larger λ led to higher Jaccard value because gene selections were controlled by more strict consistency penalties among multiple studies. For example, when $\sigma_0 = 1.5$, biclusters detected from single-study (red line, $\lambda = 0$) only reached Jaccard values that were smaller than 0.6, while meta-analysis performed much better. When setting an extreme λ value, for example, 10000 (purple line), the Jaccard value was higher than 0.85. Notice that, when λ hit a certain threshold, gene selections for all the studies were exactly the same and the performance would not be improved if λ increased further. For example, in Fig. 4.2A, the $\lambda = 1000$ line (blue) were almost identical to $\lambda = 10000$ line (purple). It is also worth noting that, larger lambda does not necessarily guarantee better performance if prevalent genes and study-specific genes are of interest. For example, Fig. 4.1 and C.1 shows the performance of consistent + prevalent gene and consistent + study-specific gene simulations for different lambda settings. In these simulations, Jaccard performance may increase and then drop as λ becomes large. This is because very large λ decreases power to detect study-specific genes. In the following analyses of this paper, we concentrate on detection of only consistent genes to maximize chance of independent validation. As a result, $\lambda = 10000$ is set to encourage consistent genes selection.

We further assessed detection power of single study analysis or MetaBiclust when smaller sample sizes are available. Fig. 4.2C shows the Jaccard accuracy of detected biclusters with different percentage of subsampling. For example, there were 100, 100, 80 and 70 samples in the four simulated studies. When subsampling percentage is 30%, sample sizes reduced to 30, 30, 24 and 21. Jaccard index of single-study analysis dramatically reduced to $\sim$50% while Jaccard of MetaBiclust remained as high as 0.85.

4.3.3 Pruning parameter selection by gap statistic and gene size control

In the meta-analytic framework, pruning step is integrated into the objective function in order to estimate the parameters and to avoid large non-informative biclusters. Fig. 4.2 compares the performances of bicluster discovery with and without pruning steps. When studying the biclustering performance, pruning step ($\tau_1 = \tau_2 = 0.05$, Fig. 4.2B) can increase

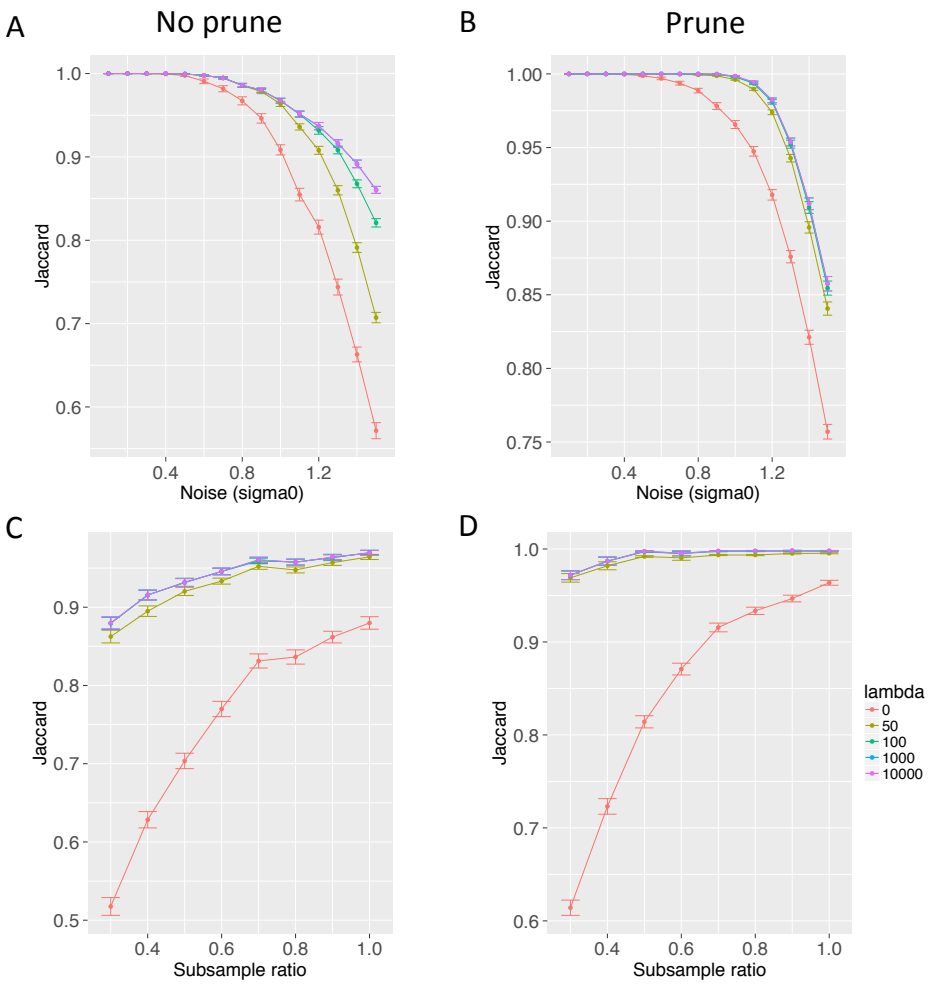


Figure 4.2: Performance of biclustering detection by single-study versus meta-analysis towards noise and sample size. (A) Jaccard similarity over simulation noise without pruning steps; (B) Jaccard similarity over simulation noise with pruning steps; (C) Jaccard similarity over sample subset ratio without pruning steps; (D) Jaccard similarity over sample subset ratio with pruning steps.

the Jaccard accuracy in a large scale when compared to non-pruning analysis($\tau_1 = \tau_2 = 0$, Fig. 4.2A). Similarly, in the subsampling analysis, pruning results outperform the non-pruning ones (Fig. 4.2D over Fig. 4.2C).

When pruning step in Eq. 4.1 was first proposed by [Turner et al., 2005a,b], the param-

eters ν_1 and ν_2 were arbitrarily recommended in the paper. Here to provide an automatic pipeline for pruning parameter selection for (τ_1 and τ_2) in MetaBiclust, we borrowed the concept of gap statistic from cluster number selection [Tibshirani et al., 2001]. Fig. 4.3A compares three settings: red for Jaccard similarity for bicluster detection without pruning step, green for results with pruning parameters seleceted by gap statistic, and blue the best performance we can achieve within the searching space. The result shows that gap statistic can successfully choose the pruning parameters to help increase biclustering performance, especially when the noise is large. Besides consistent genes, we also simulated consistent + prevalent genes and consistent + study-specific genes(Fig. C.2 and Fig. C.3). All of them prove a better perfromance of gap statistic over non-pruning analysis.

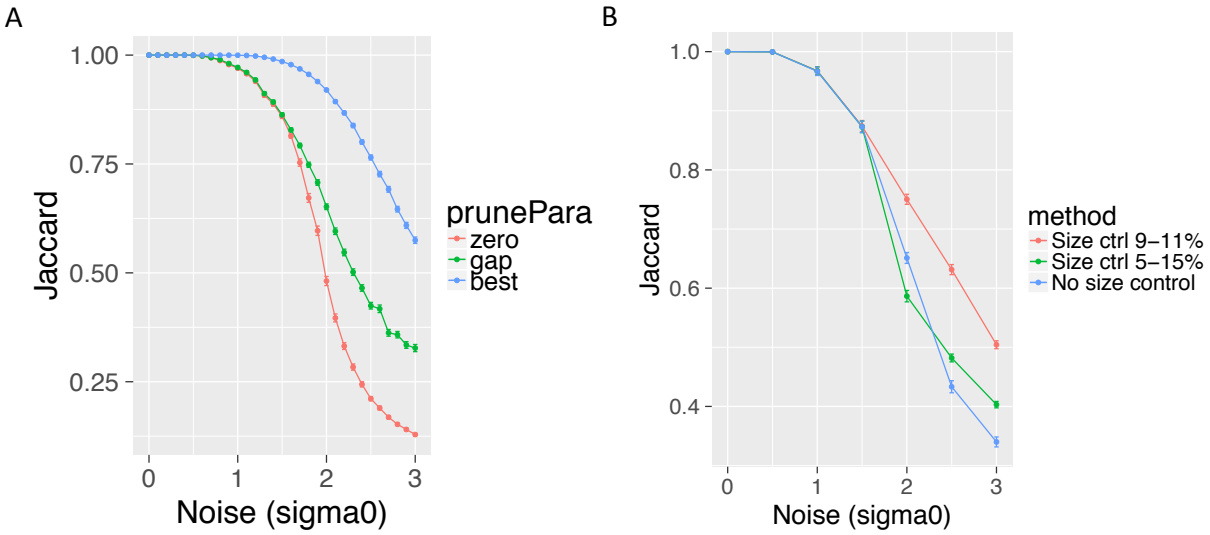


Figure 4.3: Performance of MetaBiclust detection. (A) Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by gap statistic (green line), and the best performance within the searching space (blue line). (B) Performance of bicluster detection with strong (9% to 11%, red line), week (5% to 15%, green line), and without gene size control (blue line). The underlying truth of gene selection rate is 10%.

Alternatively we tried to maximize the log of gap statistic ($\max_{(\tau_1,\tau_2)} \log(Gap(\tau_1,\tau_2)) = \log(E_{\text{null}}[obj(\tau_1,\tau_2)]) - \log(obj(\tau_1,\tau_2))$). Fig. C.4 shows a very similar result to Fig. 4.3

for consistent gene simulations. Similarly, Fig. C.5 and Fig. C.6 illustrate the log of gap statistic performance on consistent + prevalent and consistent + study-specific gene simulations. With clear evidence that both gap statistic and log of gap statistic lead to similar performance, we finally decide to implement gap statistic for self-evaluation of pruning parameters.

It has been reported that gap statistic tends to over-estimate the number of clusters [Yan and Ye, 2007]. In our situation, the algorithm tends to identify biclusters with very large gene sets when noise level is high. To rank genes for controlling gene size, we proposed the concept of gene information, which is defined as $\frac{\sum_{j:\hat{\kappa}_j=1}\left(\hat{Z}_{ij}-\hat{\theta}_{ij}\right)^2}{\sum_{j:\hat{\kappa}_j=1}\hat{Z}_{ij}^2}$. For example, if the gene size range is set by user to be $[50, 200]$, for each EM iteration step, if the number of selected genes exceeds 200, only the top 200 genes will be selected based on their gene information ranking. We simulated true bicluster gene size to be 10% of the total genes. Fig. 4.3B shows the performance of bicluster detection with strong (9% to 11%, red line), week (5% to 15%, green line), and without (blue line) gene size control. Correct range of gene size control will improve the biclsuter detection compared to no control because prior knowledge is introduced. However, incorrect gene size control will decrease the performance when the noise is low, but will perform better than no control when the noise is large (Fig. C.7). In our real data application, since the data signals are often noisy and complex, we decided to control the gene size to a reasonable range to the best of our knowledge (for example, between 200 to 500).

4.3.4 Applying bicluster genes selected from multiple training studies to an independent testing cohort

To evaluate accuracy of applying genes selected from MetaBiclust to an independent testing cohort, we simulated four studies as before, treated the first three studies as training data and used the fourth study for independent testing. Firstly, biclusters are discovered from the three training datasets using MetaBiclust. Genes selected by MetaBiclust are applied to the testing cohort. When discovering biclusters in the testing cohort, selected genes from training data are fixed in the EM iterations while only sample selection was performed to minimize

the objective function. Fig. 4.4 illustrates the training and testing bicluster performance when given different background noise. The result showed that testing bicluster discovery only performed slightly worse than the training bicluster detection when the noise level was low, and very close performance was found for high noise level. It proved that genes selected from training data are generalizable to a new testing cohort in bicluster detection.

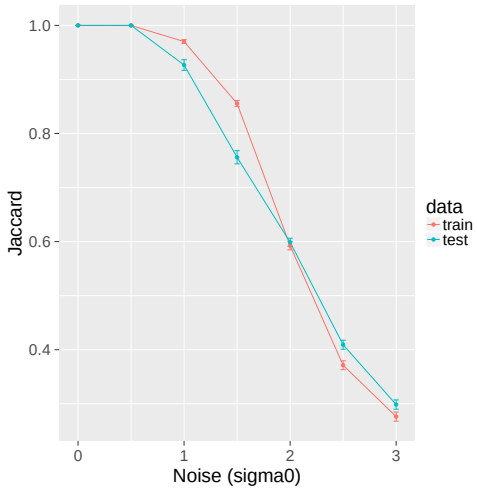


Figure 4.4: Performance evaluation of biclusters detecting from training studies (red line) and testing studies (green line). Genes from training biclsuter are applied and fixed in testing bicluster detection.

4.3.5 Breast cancer application

The MetaBiclust pipeline was applied to five breast cancer cohorts using different platforms (Table 4.1). We set a very large λ value such that biclusters detected from each single study contained exactly the same gene selection. The number of genes and samples for the detected biclusters are shown in Table C.2. As expected, the first several biclusters included all the samples, showing gene modules co-expressed across the whole population. Starting from the 9th bicluster, we were able to obtain biclusters with subset of samples. Taking the 12^{th} bicluster as an example, Fig. C.8 shows the heatmap of the biclusters detected, where each row represents a selected gene and columns represent the samples. Samples inside the bicluster

were marked by orange color and otherwise by grey color. Fig C.8 illustrates that biclusters can successfully divide samples with different expression pattern among a selected subset of genes. Breast cancer patients are traditionally classified by their estrogen receptor (ER) status (ER positive or ER negative). With clinical information available for METABRIC, samples in many biclusters are highly associated with ER status (Table C.3 by Fisher's exact test). Currently, breast cancer patients are often classified into five subtypes: Luminal A, Luminal B, Her2-enriched, Basal and normal-like. For METABRIC cohort, samples inside the 12th bicluster are more enriched in Her and Basel subtypes (Table C.4) using the PAM50 classification rule [Parker et al., 2009; Tibshirani et al., 2002]. Further, Fig 4.5 and Table C.3 shows that samples selected by the 12^{th} bicluster has significantly lower survival curve compared to patients not selected by the bicluster (p=7.13E-11 using log-rank test). Pathway enrichment analysis is applied to genes selected by each bicluster (from bicluster 9 to 15). Table C.5 lists the number of significantly enriched pathways (or gene sets) under FDR control at 5%. For the 12^{th} bicluster, top enriched pathways are listed in Table 4.3. In chemical and genetic perturbation (CGP) category, gene lists from many previous breast cancer experiments have shown extremely high association. In Reactome and Gene Ontology, many enriched pathways are fundamental cell cycle regulations and chromosomal organization or helicase activity. Both evaluations on sample and gene selections of detected biclusters indicate that the proposed MetaBiclust pipeline integrates multiple transcriptomic studies to detect biclusters that form basis to characterize clinically meaningful disease subtypes.

4.4 CONCLUSION AND DISCUSSION

In this paper, we extended the plaid model algorithm to meta-analytic framework aiming to detect biclusters from multiple cohorts at the same time. With known underlying truth in synthetic data, meta-analysis can increase the bicluster detection accuracy compared to single-study analysis, especially when the simulated background noise is large. For data with small sample size, meta-analysis was also able to beat single-study analysis to detect

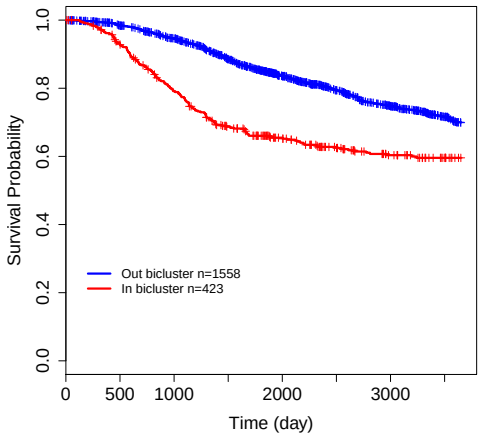


Figure 4.5: Kaplan-Meier survival curves for samples inside (red line) and outside (blue line) the bicluster of METABRIC cohorts.

Table 4.3: Top significant enriched pathways by the 12th bicluster genes.

Category	Pathway / Gene sets	p-value	q-value	# pathway genes	# bicluster genes inside pathway	Odds ratio
CGP	SMID BREAST CANCER BASAL UP	3.13E-61	8.11E-58	196	110	16.00
	VANTVEER BREAST CANCER ESR1 DN	8.54E-42	1.11E-38	106	68	18.56
	BENPORATH PROLIFERATION	5.29E-26	4.57E-23	53	38	23.39
	SOTIRIOU BREAST CANCER GRADE 1 VS 3 UP	1.43E-22	9.28E-20	49	34	20.61
	ZHOU CELL CYCLE GENES IN IR RESPONSE 24HR	2.64E-22	1.37E-19	44	32	24.08
Reactome	REACTOME G2 M CHECKPOINTS	9.40E-08	2.82E-05	11	9	37.46
	REACTOME CELL CYCLE	1.56E-07	2.82E-05	86	27	3.98
	REACTOME CELL CYCLE MITOTIC	3.03E-07	3.64E-05	68	23	4.41
	REACTOME ACTIVATION OF ATR IN RESPONSE TO REPLICATION STRESS	1.31E-06	1.18E-04	8	7	57.99
	REACTOME CELL CYCLE CHECKPOINTS	5.81E-06	4.18E-04	26	12	7.19
GO BP	GO CHROMOSOME ORGANIZATION	5.46E-10	1.49E-06	161	45	3.53
	GO MITOTIC CELL CYCLE	1.33E-09	1.82E-06	148	42	3.58
	GO CELL CYCLE PROCESS	3.94E-09	3.60E-06	194	49	3.09
	GO DNA METABOLIC PROCESS	3.42E-08	2.34E-05	112	33	3.69
	GO DNA DEPENDENT DNA REPLICATION	6.21E-08	3.40E-05	19	12	14.41
GO CC	GO CHROMOSOMAL REGION	2.01E-07	6.69E-05	48	19	5.60
	GO CHROMOSOME TELOMERIC REGION	2.00E-06	3.17E-04	24	12	8.39
	GO NUCLEAR CHROMOSOME	2.87E-06	3.17E-04	98	27	3.29
	GO NUCLEAR CHROMOSOME TELOMERIC REGION	5.46E-06	4.53E-04	22	11	8.36
	GO CHROMOSOME	3.71E-05	2.20E-03	148	33	2.49
GO MF	GO DNA HELICASE ACTIVITY	5.36E-06	2.50E-03	9	7	28.95
	GO DNA DEPENDENT ATPASE ACTIVITY	3.22E-05	7.51E-03	14	8	11.06
	GO HELICASE ACTIVITY	2.81E-04	4.38E-02	22	9	5.75

more robust biclusters in multiple studies (Fig. 4.2). We introduced gap statistic method and controlled the bicluster gene size in order to prune non-informative genes and samples automatically (Fig. 4.3). The generalizability of the bicluster genes was also proved by

applying bicluster genes detected from training data into testing data (Fig. 4.4). In real data application, we discovered biclusters from 5 breast cancer expression datasets (Table C.5).The results showed that our meta-biclustering algorithm succeeded in detecting genes enriched in breast cancer and cell cycle related pathways (Table 4.3 and Table C.5) and samples showing significantly differential survival trend (Fig. 4.5 and Table C.3).

We summarized several key discussions for applying the meta-biclustering pipeline below,

1. Both setting pruning parameters (τ_1 and τ_2) manually and automatically have their pros and cons. On one hand, selecting reasonable parameters needs prior knowledge available; while an arbitrary setting though simple, tend to lose accuracy. On the other hand, automatic setting by gap statistic permutation is able to improve the performances without additional information provided, however, it requires larger computing to do simultaneous search for (τ_1, τ_2) combinations. In our application, instead of fixing a large searching space, the algorithm starts by searching along a large interval and then zoom in gradually to balance the computing cost and searching resolution.
2. In real-data whole-genome application, we suggest to control the gene size to a reasonable range between 200 to 500 based on the following considerations. Large gene set might introduce a decent number of non-informative genes; small gene size setting, on the other hand, might over-penalize the genes and thus miss the main signals.
3. Real data are often noisier and more complex than synthetic data. For this reason, the algorithm will select all the samples for the first several biclusters in real application. But after removing these noisy signals, the algorithm is able to discover biclusters successfully with both reasonable gene and sample sizes for all the studies.
4. Meta term parameter λ will control the consistency of gene selection in multiple studies. An extremely large λ value can guarantee exactly the same gene set selection, while a zero setting represents no consistency control. In order to discovery prevalent or study-specific genes, a reasonable λ value can be set between zero and infinity to control the penalty (Fig. 4.1 and Fig. C.1).

Some future directions are proposed to improve the algorithm further. (1) Several large and non-informative biclsuters can be first subtracted from the data matrix in order to

remove background noise. Then significant biclusters with both reasonable gene and sample sizes can be detected from the remaining data matrix. (2) The plaid model can also be extended into vertical meta-analysis direction, that is, simultaneously detecting biclusters from the same cohort but different types of omics data.

4.5 APPENDIX

To prove that the first objective function is equivalent to the improved pruning step.

For a given gene i, when $\rho_i = 1$, the objective score for gene i can be written as

$$Q_{(\rho_i=1)} = \sum_{j=1}^{p} (Z_{ij} - \theta_{ij})^2 + \tau_1 \sum_{j=1}^{p} Z_{ij}^2 + \tau_2 \sum_{j=1}^{p} \left[\kappa_j \left(\sum_{i=1}^{n} Z_{ij}^2 \right) \right].$$

When $\rho_i = 0$, the objective score is

$$Q_{(\rho_i=0)} = \sum_{j=1}^{p} Z_{ij}^2 + \tau_2 \sum_{j=1}^{p} \left[\kappa_j \left(\sum_{i=1}^{n} Z_{ij}^2 \right) \right].$$

It can be proved that

$$\begin{aligned}
\rho_i =& I\left[Q_{(\rho_i=1)} < Q_{(\rho_i=0)}\right] \\
=& I\left[\sum_{j=1}^{p} (Z_{ij} - \theta_{ij})^2 + \tau_1 \sum_{j=1}^{p} Z_{ij}^2 + \tau_2 \sum_{j=1}^{p} \left[\kappa_j \left(\sum_{i=1}^{n} Z_{ij}^2 \right) \right] \right. \\
& \left. < \sum_{j=1}^{p} Z_{ij}^2 + \tau_2 \sum_{j=1}^{p} \left[\kappa_j \left(\sum_{i=1}^{n} Z_{ij}^2 \right) \right] \right] \\
=& I\left[\sum_{j=1}^{p} (Z_{ij} - \theta_{ij})^2 < (1 - \tau_1) \sum_{j=1}^{p} Z_{ij}^2 \right]
\end{aligned}$$

where $I(expr)$ is the indicator function that equals to one if $expr$ is true, otherwise zero.

Similar to sample partition pruning. As a whole, it proves that the integrated objective function is equivalent to the improved form of pruning steps.

5.0 CONCLUSIONS

This dissertation majorly includes three genomic integrative studies to improve the performance of fusion transcripts detection from paired-end RNA-seq data, gene three-way liquid association analysis, and bicluster discovery.

In chapter 1, we introduced two genomic high-throughput technologies (microarray and next generation sequencing) and briefly reviewed fusion transcripts, gene expression and regulation, and biclustering algorithms. The work can be divided into two types of integration. On one hand, multiple tools (or algorithms) have been developed for the same purpose while in general no single one can over-perform the others in each evaluation criteria. This motivates the idea to combine the results of top performing tools to increase the overall performance. Chapter 2 compared 15 fusion transcript detection tools using RNA-seq data and developed a meta-caller to combine the results from multiple pipelines. On the other hand, taking the advantage of high-throughput genomic data from many public accessible database, researchers are able to collect multiple types of omics data from different cohorts or platforms. Instead of limited to one single cohort, combining multiple cohorts can increase the stability and produce more generalizable results for future new studies. Chapter 3 and Chapter 4 aimed to developed meta-analysis pipelines to detect liquid association and biclustering from multiple cohorts.

In chapter 2 for the first project, we evaluated 15 tools developed to detect fusion transcripts from paired-end RNA-seq data. Three types of data are used for the comparison: *in silico* synthetic read sequences with known fusions, real data RNA-seq (breast cancer, melanoma and prostate cancer) with validated fusion transcripts, and an experimentally synthesized data with designed fusions. Tools are evaluated in terms of precision, recall, F-measure, number of supporting reads and computing time for data with different sequencing

depth, read length, insert size and background noise. Though we can still rank the overall performance of these tools, but no single pipeline was able to perform better than the others in all the criteria. In order to improve the performance, we proposed a meta-caller to combine the results from several top performing pipelines to re-prioritize the candidate fusion transcripts. Our results showed that the fusion meta-caller can overall improve the balance between precision and recall rates, and thus provide a better ranked fusion candidate list for experimental validation.

In chapter 3 for the second project, we extended the liquid association algorithm into two meta-analytic frameworks: MetaLA for liquid association and MetaMLA for modified liquid association. Liquid association quantifies the three-way gene regulation where the correlation between two genes depends on the expression level of a third gene (scouting gene). In order to improve the performance of liquid association detection, we extend the algorithm to multiple cohorts to detect more robust and generalizable results. Our methods were applied into five Yeast datasets treated with different environmental stimuli. Compared to single-analysis, genes from the top triplets of both the MetaLA and MetaMLA pipelines were enriched into more significant pathways. Besides, enriched pathways from meta-analysis were more consistent with each single-study, while there is almost no overlap between pairwise single studies. In addition, genes from the top triplets can contribute to gene network construction, where many predicted associations can be validated by known protein-protein interactions. The results showed the promising future that our MetaLA and MetaMLA pipeline can be used for three-way gene regulation predictions.

In chapter 4 for the third project, we aimed to detect biclusters from multiple cohorts with consistent gene selection. Plaid model is one of the most popular biclustering algorithms that assumes the expression matrix to be the superposition of multiple layers (biclusters), and each bicluster can be represented as the sum of mean, row (gene) and column (sample) effects. In order to gain power, we extended this single-study algorithm into meta-analysis framework. In simulation studies (with known biclusters), compared to single-study analysis, our meta-biclustering algorithm was able to increase the accuracy especially for large simulation noise and small sample size. Gap statistic was introduced into the algorithm to estimate pruning parameters for more informative gene and sample selection. In real data application, our

algorithm was applied to five breast cancer cohorts. The bicluster genes were highly enriched into many breast cancer related pathways, and the bicluster sample splitting can group samples with different survival behaviors. All these results showed that our meta-biclsutering provides a better method to group genes and samples from multiple cohorts.

As a whole, this dissertation improved the fusion transcript detection, liquid association analysis and biclustering by the integration of multiple methods or cohorts. These provide better predictions of the genome structure variation, complex gene regulation pattern, and disease subtype detection, and thus will contribute to the better understanding of disease occurrence or carcinogenesis.

APPENDIX A

SUPPLEMENTARY MATERIAL FOR AIM 1

A.1 SUPPLEMENTARY TABLES

Table A.1: Summary of computational tools published since 2010.

Tools	Fusion point	Platform	Published
MapSplice	No restriction	Illumina	Wang et al. [2010]
Trans-ABySS	Canonical splicing pattern	Illumina	Robertson et al. [2010]
FusionSeq	No restriction	Illumina/SOLiD	Sboner et al. [2010]
ShortFuse	No restriction	Illumina/SOLiD	Kinsella et al. [2011]
Comrad	No restriction	Illumina	McPherson et al. [2011b]
FusionHunter	Canonical splicing pattern	Illumina	Li et al. [2011b]
FusionMap	Canonical splicing pattern	Illumina	Ge et al. [2011]
deFuse	No restriction	Illumina	McPherson et al. [2011a]
SnowShoes-FTD	Within exon boundary	Illumina	Asmann et al. [2011]
chimerascan	No restriciton	Illumina	Iyer et al. [2011]
FusionCatcher	No restriction	Illumina/SOLiD	Nicorici et al. [2014]
TopHat-Fusion	No restriction	Illumina	Kim and Salzberg [2011]
BreakFusion	No restriction	Illumina	Chen et al. [2012]
FusionAnalyser	Within exon boundary	Illumina	Piazza et al. [2012]
LifeScope		SOLiD	Sakarya et al. [2012]
Bellerophontes	No restriction	Illumina	Abate et al. [2012]
FusionFinder	Within exon boundary	Illumina	Francis et al. [2012]
EricScript	Within exon or exon boundary	Illumina	Benelli et al. [2012]
SOAPfuse	No restriction	Illumina	Jia et al. [2013]
FusionQ	No restriction	Illumina/SOLiD	Liu et al. [2013]
SOAPfusion	Within exon or exon boundary	Illumina	Wu et al. [2013]
PRADA	Within exon boundary	Illumina	Torres-García et al. [2014]
JAFFA	Within exon or exon boundary	Illumina	Davidson et al. [2015]

Table A.2: Description of fifteen fusion detection tools and their default (or available) detection and filtering parameters.

Tool (version)	Anchor length filter	Read-through transcript filter	Supported reads filter (spanning / split)	PCR artifact filter	Homology based filter	Alignment tools	Assembly (#) / Machine learning (o)	Fusion db	Description of installation
MapSplice (V 2.1.9)	N	N	N	N	N	bowtie	N	N	Python script. Easy to install.
ShortFuse (V 0.2)	N	N	Y	N	N	bowtie	N	Y	C++ script. Easy to install.
Fusion Hunter (V 1.4)	10	Y	1-Mar	Y	Y	bowtie	N	Y	Perl script. Easy to install.
FusionMap (V 20150331)	Y*	Y	Y	Y	Y	N	o (GSNAP)	Y	Excutatable file. Easy to install.
deFuse (V 0.6.2)	10	Y	1-Mar	N	Y	bowtie/ BLAT	N	N	C++ script. Easy to install.
chimerascan (V 0.4.5)	10	Y	4 (total)	N	N	bowtie/ BWA	N	Y	Python script. Easy to install.
Fusion Catcher (V 0.99.4b)	10	Y	1-Mar	N	Y	bowtie/ STAR/ BLAT/ bowtie2	# (velvet)	Y	Python script. easy to install.
TopHat-Fusion (V 2.0.14)	10	Y	1-Mar	N	Y	bowtie	N	Y	Python script. Easy to install.
BreakFusion (V 1.0.1)	N	N	N	N	N	BWA/ BLAT	# (TIGRA-SV)	N	Need to install supporing tools.
EricScript (V 0.5.1)	N	Y	3/1 (self filter)	Y	Y	BWA/ BLAT	o	N	Perl and R script. Easy to install.
SOAPfuse (V 1.26)	10	N	1-Mar	N	N	Soap2/ BWA/ BLAT	N	N	Perl script. Easy to install
FusionQ (V 5)	10	N	1-Mar	N	Y	bowtie	# (TIGRA-SV)	N	Perl script.
SnowShoes-FTD (V 2.0)	N	Y	2/N	Y	Y	bowtie/ BWA	N	Y	Perl script. Easy to install.
PRADA (V 1.1)	N	N	N	N	N	BWA/ BLAST	N	N	Python script. Easy to install.
JAFFA (V 1.06)	N	Y	3/1 (self filter)	N	Y	bowtie2/ BLAT	N	N	Java script. Easy to install.

Table A.3: 150 designed fusions in the synthetic data.

Gene1ID	Gene2ID	Gene1Name	Gene2Name
ENSG00000008282	ENSG00000075790	SYPL1	BCAP29
ENSG00000011405	ENSG00000244165	PIK3C2A	P2RY11
ENSG00000022556	ENSG00000100504	NLRP2	PYGL
ENSG00000028310	ENSG00000173273	BRD9	TNKS
ENSG00000055609	ENSG00000100364	MLL3	KIAA0930
ENSG00000062038	ENSG00000044115	CDH3	CTNNA1
ENSG00000069493	ENSG00000102024	CLEC2D	PLS3
ENSG00000070501	ENSG00000196873	POLB	CBWD3
ENSG00000072041	ENSG00000182944	SLC6A15	EWSR1
ENSG00000077549	ENSG00000143549	CAPZB	TPM3
ENSG00000084710	ENSG00000026103	EFR3B	FAS
ENSG00000095539	ENSG00000181090	SEMA4G	EHMT1
ENSG00000100239	ENSG00000105810	PPP6R2	CDK6
ENSG00000100744	ENSG00000142541	C14orf129	RPL13A
ENSG00000101138	ENSG00000115946	CSTF1	PNO1
ENSG00000101945	ENSG00000116703	SUV39H1	PDC
ENSG00000105568	ENSG00000126070	PPP2R1A	EIF2C3
ENSG00000106991	ENSG00000105856	ENG	HBP1
ENSG00000109072	ENSG00000139372	SEBOX	TDG
ENSG00000110786	ENSG00000146013	PTPN5	GFRA3
ENSG00000112414	ENSG00000100276	GPR126	RASL10A
ENSG00000114120	ENSG00000171307	SLC25A36	ZDHHC16
ENSG00000114626	ENSG00000169439	ABTB1	SDC2
ENSG00000115392	ENSG00000173614	FANCL	NMNAT1
ENSG00000115935	ENSG00000163430	WIPF1	FSTL1

ENSG00000115966	ENSG00000115446	ATF2	UNC50
ENSG00000116809	ENSG00000140577	ZBTB17	CRTC3
ENSG00000118557	ENSG00000051382	PMFBP1	PIK3CB
ENSG00000119139	ENSG00000188554	TJP2	NBR1
ENSG00000119772	ENSG00000130640	DNMT3A	TUBGCP2
ENSG00000121577	ENSG00000205981	POPDC2	DNAJC19
ENSG00000124596	ENSG00000198363	C6orf130	ASPH
ENSG00000125122	ENSG00000186231	LRRC29	KLHL32
ENSG00000125630	ENSG00000119535	POLR1B	CSF3R
ENSG00000128512	ENSG00000186952	DOCK4	TMEM232
ENSG00000130038	ENSG00000150527	EFCAB4B	CTAGE5
ENSG00000130844	ENSG00000104325	ZNF331	DECR1
ENSG00000132321	ENSG00000170236	IQCA1	USP50
ENSG00000132323	ENSG00000173715	ILKAP	C11orf80
ENSG00000134375	ENSG00000140854	TIMM17A	KATNB1
ENSG00000135269	ENSG00000134940	TES	ACRV1
ENSG00000136541	ENSG00000048740	ERMN	CELF2
ENSG00000136546	ENSG00000171109	SCN7A	MFN1
ENSG00000137404	ENSG00000184708	NRM	EIF4ENIF1
ENSG00000142494	ENSG00000164941	SLC47A1	INTS8
ENSG00000146263	ENSG00000158552	MMS22L	ZFAND2B
ENSG00000146909	ENSG00000174236	NOM1	REP15
ENSG00000147044	ENSG00000185305	CASK	ARL15
ENSG00000148296	ENSG00000123700	SURF6	KCNJ2
ENSG00000151726	ENSG00000012048	ACSL1	BRCA1
ENSG00000154556	ENSG00000143774	SORBS2	GUK1
ENSG00000156502	ENSG00000106686	SUPV3L1	SPATA6L
ENSG00000157600	ENSG00000108961	TMEM164	RANGRF

ENSG00000157601	ENSG00000124181	MX1	PLCG1
ENSG00000158055	ENSG00000160310	GRHL3	PRMT2
ENSG00000161835	ENSG00000165138	GRASP	ANKS6
ENSG00000162174	ENSG00000104936	ASRGL1	DMPK
ENSG00000163281	ENSG00000164896	GNPDA2	FASTK
ENSG00000165219	ENSG00000148660	GAPVD1	CAMK2G
ENSG00000165275	ENSG00000153029	TRMT10B	MR1
ENSG00000165802	ENSG00000137207	NELF	YIPF3
ENSG00000166333	ENSG00000129219	ILK	PLD2
ENSG00000167822	ENSG00000164032	OR8J3	H2AFZ
ENSG00000168000	ENSG00000141564	BSCL2	RPTOR
ENSG00000168214	ENSG00000149571	RBPJ	KIRREL3
ENSG00000170525	ENSG00000188859	PFKFB3	FAM78B
ENSG00000171055	ENSG00000183474	FEZ2	GTF2H2C
ENSG00000172939	ENSG00000169239	OXSR1	CA5B
ENSG00000173482	ENSG00000156990	PTPRM	RPUSD3
ENSG00000173638	ENSG00000139233	SLC19A1	LLPH
ENSG00000175691	ENSG00000120910	ZNF77	PPP3CC
ENSG00000178882	ENSG00000165799	FAM101A	RNASE7
ENSG00000186431	ENSG00000048828	FCAR	FAM120A
ENSG00000186523	ENSG00000154429	FAM86B1	CCSAP
ENSG00000188493	ENSG00000182934	C19orf54	SRPR
ENSG00000196104	ENSG00000185627	SPOCK3	PSMD13
ENSG00000203685	ENSG00000148426	C1orf95	C10orf47
ENSG00000205327	ENSG00000256061	OR6C68	DYX1C1
ENSG00000215454	ENSG00000100918	KRTAP10-4	REC8
ENSG00000255501	ENSG00000001084	CARD18	GCLC
ENSG00000007264	ENSG00000122642	MATK	FKBP9

ENSG00000007350	ENSG00000137501	TKTL1	SYTL2
ENSG00000011347	ENSG00000136247	SYT7	ZDHHC4
ENSG00000015133	ENSG00000205583	CCDC88C	STAG3L1
ENSG00000037474	ENSG00000189308	NSUN2	LIN54
ENSG00000065150	ENSG00000110077	IPO5	MS4A6A
ENSG00000074319	ENSG00000082516	TSG101	GEMIN5
ENSG00000078618	ENSG00000076554	NRD1	TPD52
ENSG00000085377	ENSG00000196642	PREP	RABL6
ENSG00000087460	ENSG00000140299	GNAS	BNIP2
ENSG00000092421	ENSG00000167173	SEMA6A	C15orf39
ENSG00000102125	ENSG00000049618	TAZ	ARID1B
ENSG00000102243	ENSG00000102271	VGLL1	KLHL4
ENSG00000103363	ENSG00000172366	TCEB2	FAM195A
ENSG00000103591	ENSG00000075426	AAGAB	FOSL2
ENSG00000109339	ENSG00000179698	MAPK10	KIAA1875
ENSG00000111271	ENSG00000174175	ACAD10	SELP
ENSG00000112624	ENSG00000107438	KIAA0240	PDLIM1
ENSG00000116254	ENSG00000083454	CHD5	P2RX5
ENSG00000116761	ENSG00000116830	CTH	TTF2
ENSG00000118729	ENSG00000113312	CASQ2	TTC1
ENSG00000118997	ENSG00000151779	DNAH7	NBAS
ENSG00000119042	ENSG00000137948	SATB2	BRDT
ENSG00000119844	ENSG00000152413	AFTPH	HOMER1
ENSG00000119986	ENSG00000115685	AVPI1	PPP1R7
ENSG00000122145	ENSG00000127993	TBX22	RBM48
ENSG00000122741	ENSG00000102359	DCAF10	SRPX2
ENSG00000123552	ENSG00000106078	USP45	COBL
ENSG00000126107	ENSG00000013288	HECTD3	MAN2B2

ENSG00000130787	ENSG00000002726	HIP1R	ABP1
ENSG00000133026	ENSG00000163328	MYH10	GPR155
ENSG00000133985	ENSG00000181135	TTC9	ZNF707
ENSG00000134324	ENSG00000088305	LPIN1	DNMT3B
ENSG00000134343	ENSG00000101158	ANO3	TH1L
ENSG00000134627	ENSG00000204580	PIWIL4	DDR1
ENSG00000137070	ENSG00000088298	IL11RA	EDEM2
ENSG00000138069	ENSG00000144908	RAB1A	ALDH1L1
ENSG00000138386	ENSG00000158639	NAB1	PAGE5
ENSG00000138395	ENSG00000123130	CDK15	ACOT9
ENSG00000140285	ENSG00000100353	FGF7	EIF3D
ENSG00000144820	ENSG00000108292	GPR128	MLLT6
ENSG00000146828	ENSG00000168275	SLC12A9	C1orf31
ENSG00000147133	ENSG00000127989	TAF1	MTERF
ENSG00000149636	ENSG00000134453	DSN1	RBM17
ENSG00000150477	ENSG00000075914	KIAA1328	EXOSC7
ENSG00000150995	ENSG00000142920	ITPR1	ADC
ENSG00000151136	ENSG00000177106	BTBD11	EPS8L2
ENSG00000154134	ENSG00000131503	ROBO3	ANKHD1
ENSG00000160867	ENSG00000134986	FGFR4	NREP
ENSG00000162909	ENSG00000168569	CAPN2	TMEM223
ENSG00000163541	ENSG00000163166	SUCLG1	IWS1
ENSG00000168509	ENSG00000102158	HFE2	MAGT1
ENSG00000169230	ENSG00000158517	PRELID1	NCF1
ENSG00000169914	ENSG00000173077	OTUD3	1-Dec
ENSG00000170166	ENSG00000092203	HOXD4	TOX4
ENSG00000172007	ENSG00000141540	RAB33B	TTYH2
ENSG00000174899	ENSG00000139182	C3orf55	CLSTN3

ENSG00000176641	ENSG00000148985	RNF152	PGAP2
ENSG00000177311	ENSG00000164129	ZBTB38	NPY5R
ENSG00000183963	ENSG00000153237	SMTN	CCDC148
ENSG00000184497	ENSG00000099204	FAM70B	ABLIM1
ENSG00000185189	ENSG00000239713	NRBP2	APOBEC3G
ENSG00000185303	ENSG00000125676	SFTPA2	THOC2
ENSG00000188282	ENSG00000162191	RUFY4	UBXN1
ENSG00000188343	ENSG00000104824	FAM92A1	HNRNPL
ENSG00000189157	ENSG00000073417	FAM47E	PDE8A
ENSG00000197561	ENSG00000125944	ELANE	HNRNPR
ENSG00000198130	ENSG00000166428	HIBCH	PLD4
ENSG00000205726	ENSG00000188130	ITSN1	MAPK12
ENSG00000239857	ENSG00000180822	GET4	PSMG4

Table A.4: The read numbers of type-1A and type-1B synthetic datasets.

Coverage	50 bp	75 bp	100 bp
5X	16,832	11,221	8,416
20X	67,328	44,885	33,664
50X	168,320	112,213	84,160
100X	336,641	224,427	168,320
200X	673,282	448,854	336,641

Table A.5: Read numbers for type-2, type-3A and type-3B synthetic datasets.

Type	Composition	50 bp	75 bp	100 bp
Type-2	Background data	2,000,000	2,000,000	2,000,000
Type-3A	Synthetic data	336,641	224,426	168,284
Type-3B	Synthetic + Background data	2,336,641	2,224,426	2,168,284

Table A.6: Insert sizes for type-2, type-3A and type-3B synthetic datasets.

Tissue	SRA ID	Insert size mean value	Insert size SD value
Lung	SRR349695	164 bp	48 bp
Parathyroid	SRR479053	192 bp	85 bp
Skeletal myocyte	SRR1693845	353 bp	116 bp
Bladder	SRR400342	248 bp	26 bp
T cell	SRR1909130	290 bp	120 bp

Table A.7: Data description for three real datasets.

Cancer	Sample	Read length (bp)	Insert size (bp)	Read number	Validated fusions
Breast Cancer	BT474	50	180 ± 80	21,423,697	ACACA - STAC2, RPS6KB1 - SNF8, VAPB - IKZF3, ZMYND8 - CEP250, RAB22A - MYO9B, SKA2 - MYO19, DIDO1 - KIAA0406, STARD3 - DOK5, LAMP1 - MCF2L, GLB1 - CMTM7, CPNE1 - PI3
	KPL4	50	180 ± 80	6,796,443	BSG - NFIX, PPP1R12A - SEPT10, NOTCH1 - NUP214
	MCF7	50	180 ± 80	8,409,785	BCAS4 - BCAS3, ARFGEF2 - SULF2, PRPS6KB1 - TMEM49
	SKBR3	50	180 ± 80	18,140,246	TATDN1 - GSDMB, CSE1L - ENSG00000236127, RARA - PKIA, ANKHD1 - PCDH1, CCDC85C - SETD3, SUMF1 - LRRFIP2, WDR67 - ZNF704, CYTH1 - EIF3H, DHX35 - ITCH, NFS1 - PREX1
Melanoma	501Mel	50	351 ± 139	14,857,046	CCT3 - C1orf61, GNA12 - SHANK2, SLC12A7 - C11orf67, PARP1 - MIXL1
	M000216	50	393 ± 115	13,868,165	KCTD2 - ARHGEF12
	M000921	50	627 ± 564	14,468,771	TMEM8B - TLN1, RECK - ALX3
	M010403	50	374 ± 159	8,168,750	SCAMP2 - WDR72
	M980409	50	334 ± 119	15,768,555	GCN1L1 - PLA2G1B
	M990802	50	355 ± 125	16,066,999	ANKHD1 - C5orf32, RB1 - ITM2B
Prostate Cancer	158T	100	140 ± 27	221,206,388	SLC45A3 - ELK4, MTOR - TP53BP1
	159T	100	153 ± 33	159,766,465	TRMT11 - GRIK2, MAN2A1 - FER, KDM4B - AC011523.2, CCNH - C5orf30
	165T	100	158 ± 40	243,191,643	SLC45A3 - ELK4, TMEM135 - CCDC67
	171T	100	143 ± 30	118,742,381	TMPRSS2 - ERG
	49T	100	158 ± 37	250,071,864	SLC45A3 - ELK4, TMPRSS2 - ERG, LRRC59 - SLC22A10

Table A.8: Parameter setting for TopHat-Fusion.

Options	Explanation
–no-coverage-search	Disable the coverage based search for junctions
–fusion-min-dist 1000000	Default minimum distance
–fusion-ignore-chromosomes chrM	Ignore chromosome M

Table A.9: Parameter setting for anchor length and spanning/split reads of all the fifteen tools.

Detection tools	Anchor length	Spanning read	Split read	Note
MapSplice	-	-	-	User self-filter for spanning +split as 4
ShortFuse	-	-	-	A score to rank the detected fusion transcripts
FusionHunter	10	3	1	
FusionMap	Min(25, Max(17, floor(ReadLength/3)))	Distinct definition	Distinct definition	
deFuse	10	3	1	
chimerascan	10	-	-	Set spanning+split as 4; maximum mismatch is 3
FusionCatcher	10	3	1	
TopHat-Fusion	10	3	1	
BreakFusion	-	-	-	User self-filter for spanning +split as 4
EricScript	-	-	-	User self-filter for spanning /split read on the output
SOAPfuse	10	3	1	
FusionQ	10	3	1	
PRADA	-	-	-	User self-filter for spanning +split as 4
SnowShoes-FTD	-	-	-	Set spanning +split as 4
JAFFA	-	3	1	User self-filter for spanning /split read on the output

Table A.10: Completeness of the fifteen tools on the synthetic and real datasets.

Tools	Type-1 synthetic dataset	Type-2 synthetic dataset	Type-3 synthetic dataset	Breast cancer dataset	Melanoma dataset	Prostate dataset	Validation dataset	Note
MapSplice	Y	Y	Y	Y	Y	49T failed	Y	Running time exceed 14 days and larger than 2TB temp files
ShortFuse	100 bp data failed	75 bp and 100 bp data failed	0 output for most of the datasets	Y	Y	failed	failed	Error with 'parsing the discordant reads'
FusionHunter	failed	failed	failed	Y	Y	Y	Y	Stack smashing for synthetic data
FusionMap	Y	Y	Y	Y	Y	Y		
deFuse	Y	Y	Y	Y	Y	failed	0 output	Running time exceed 14 days
chimerascan	Y	Y	Y	Y	Y	Y	Y	
FusionCatcher	Y	Y	Y	Y	Y	Y	Y	
Tophat-Fusion	Y	Y	Y	Y	Y	Y	Y	Changing parameter setting
BreakFusion	Y	Y	Y	Y	Y	Y	Y	
Ericscript	Y	Y	Y	Y	Y	0 output	Y	
SOAPfuse	Y	Y	Y	Y	Y	Y	Y	
FusionQ	Y	Y	Y	Partially could not complete (CNC)	CNC	CNC	0 output	Tool cannot stop at 3rd step if data are large
PRADA	Y	Y	Y	Y	Y	Y	Y	
SnowShoes -FTD	Y	Y	Y	Y	Y	Y	Y	Trim 100 bp reads to 50 bp
JAFFA	Y	Y	Y	Y	Y	Y	Y	

Table A.11: The read numbers of prostate cancer 171T dataset and its subsamples.

Subsample	Read numbers
1/1	118,742,381
1/2	59,371,192
1/4	29,685,596
1/8	14,842,798

Table A.12: The summary of the recall rates, precision rates and F-measures for type-1A with read 100 bp & 100X.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	139	26	11	165	0.927	0.842	0.882
FusionCatcher	108	20	42	128	0.72	0.844	0.777
JAFFA	88	16	62	104	0.587	0.846	0.693
EricScript	104	13	46	117	0.693	0.889	0.779
chimerascan	105	30	45	135	0.7	0.778	0.737
PRADA	57	2	93	59	0.38	0.966	0.545
deFuse	73	9	77	82	0.487	0.89	0.63
FusionMap	92	27	58	119	0.613	0.773	0.684
TopHat-Fusion	53	14	97	67	0.353	0.791	0.488
MapSplice	51	8	99	59	0.34	0.864	0.488
BreakFusion	106	44	44	150	0.707	0.707	0.707
SnowShoes-FTD	3	1	147	4	0.02	0.75	0.039
FusionQ	81	18	69	99	0.54	0.818	0.651
FusionHunter	-	-	-	-	-	-	-
ShortFuse	-	-	-	-	-	-	-

Table A.13: The summary of the recall rates, precision rates and F-measures for type-1B with read 100 bp & 100X.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	136	22	14	158	0.907	0.861	0.883
FusionCatcher	110	18	40	128	0.733	0.859	0.791
JAFFA	86	20	64	106	0.573	0.811	0.672
EricScript	111	15	39	126	0.74	0.881	0.804
chimerascan	96	26	54	122	0.64	0.787	0.706
PRADA	57	3	93	60	0.38	0.95	0.543
deFuse	123	15	27	138	0.82	0.891	0.854
FusionMap	95	22	55	117	0.633	0.812	0.711
TopHat-Fusion	64	16	86	80	0.427	0.8	0.557
MapSplice	53	9	97	62	0.353	0.855	0.5
BreakFusion	74	36	76	110	0.493	0.673	0.569
SnowShoes-FTD	3	1	147	4	0.02	0.75	0.039
FusionQ	51	12	99	63	0.34	0.81	0.479
FusionHunter	-	-	-	-	-	-	-
ShortFuse	-	-	-	-	-	-	-

Table A.14: The summary of the recall rates, precision rates and F-measures for type-3B lung sample with read 50 bp & 100X.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	138	27	12	165	0.92	0.836	0.876
FusionCatcher	102	19	48	121	0.68	0.843	0.753
JAFFA	88	12	62	100	0.587	0.88	0.704
EricScript	105	15	45	120	0.7	0.875	0.778
chimerascan	91	25	59	116	0.607	0.784	0.684
PRADA	55	2	95	57	0.367	0.965	0.532
deFuse	43	7	107	50	0.287	0.86	0.43
FusionMap	72	11	78	83	0.48	0.867	0.618
TopHat-Fusion	66	19	84	85	0.44	0.776	0.562
MapSplice	55	9	95	64	0.367	0.859	0.514
BreakFusion	114	141	36	255	0.76	0.447	0.563
SnowShoes-FTD	3	1	147	4	0.02	0.75	0.039
FusionQ	37	37	113	74	0.247	0.5	0.331
FusionHunter	-	-	-	-	-	-	-
ShortFuse	-	-	-	-	-	-	-

Table A.15: The correlation between five normal tissues by the F-measure of the fifteen tools on the type-3B dataset.

	Lung	Parathyroid	Bladder	Skeletal myocyte	T cell
Lung	1	0.95	0.78	0.94	0.92
Parathyroid	-	1	0.79	0.99	0.79
Bladder	-	-	1	0.87	0.56
Skeletal myocyte	-	-	-	1	0.76
T cell	-	-	-	-	1

Table A.16: The summary of the recall rates, precision rates and F-measures for breast cancer data.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	20	48	7	68	0.741	0.294	0.421
FusionCatcher	19	48	8	67	0.704	0.284	0.405
JAFFA	16	16	11	32	0.593	0.5	0.543
EricScript	16	67	11	83	0.593	0.193	0.291
chimerascan	19	96	8	115	0.704	0.165	0.267
PRADA	15	22	12	37	0.556	0.405	0.469
deFuse	19	116	8	135	0.704	0.141	0.235
FusionMap	6	126	21	132	0.222	0.045	0.075
TopHat-Fusion	15	58	12	73	0.556	0.205	0.3
MapSplice	16	37	11	53	0.593	0.302	0.4
BreakFusion	15	1923	12	1938	0.556	0.008	0.016
SnowShoes-FTD	15	5	12	20	0.556	0.75	0.639
FusionQ	4	453	23	457	0.148	0.009	0.017
FusionHunter	13	10	14	23	0.481	0.565	0.52
ShortFuse	19	24	8	43	0.704	0.442	0.543

Table A.17: The summary of the recall rates, precision rates and F-measures for melanoma data.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	10	98	1	108	0.909	0.093	0.169
FusionCatcher	3	6	8	9	0.273	0.333	0.3
JAFFA	2	2	9	4	0.182	0.5	0.267
EricScript	3	67	8	70	0.273	0.043	0.074
chimerascan	5	189	6	194	0.455	0.026	0.049
PRADA	3	4	8	7	0.273	0.429	0.334
deFuse	10	189	1	199	0.909	0.05	0.095
FusionMap	2	85	9	87	0.182	0.023	0.041
TopHat-Fusion	4	25	7	29	0.364	0.138	0.2
MapSplice	5	39	6	44	0.455	0.114	0.182
BreakFusion	6	3092	5	3098	0.545	0.002	0.004
SnowShoes-FTD	4	1	7	5	0.364	0.8	0.5
FusionQ	-	-	-	-	-	-	-
FusionHunter	4	4	7	8	0.364	0.5	0.421
ShortFuse	7	30	4	37	0.636	0.189	0.291

Table A.18: The summary of the recall rates, precision rates and F-measures for prostate cancer data.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	7	75	5	82	0.583	0.085	0.148
FusionCatcher	11	82	1	93	0.917	0.118	0.209
JAFFA	6	1815	6	1821	0.5	0.003	0.006
EricScript	11	3798	1	3809	0.917	0.003	0.006
chimerascan	8	1665	4	1673	0.667	0.005	0.01
PRADA	0	13	12	13	0	0	0
deFuse	-	-	-	-	-	-	-
FusionMap	7	3759	5	3766	0.583	0.002	0.004
TopHat-Fusion	0	61	12	61	0	0	0
MapSplice	0	4	12	4	0	0	0
BreakFusion	1	3764	11	3765	0.083	0	0
SnowShoes-FTD	5	6	7	11	0.417	0.455	0.435
FusionQ	-	-	-	-	-	-	-
FusionHunter	0	5	12	5	0	0	0
ShortFuse	-	-	-	-	-	-	-

Table A.19: The summary of the recall rates, precision rates and F-measures for validation dataset.

Detection tools	TP	FP	FN	TP+FP	Recall	Precision	F-measure
SOAPfuse	4	9	5	13	0.444	0.308	0.364
FusionCatcher	6	17	3	23	0.667	0.261	0.375
JAFFA	5	8	4	13	0.556	0.385	0.455
EricScript	5	358	4	363	0.556	0.014	0.027
chimerascan	6	127	3	133	0.667	0.045	0.084
PRADA	3	4	6	7	0.333	0.429	0.375
deFuse	0	0	0	0	0	0	0
FusionMap	6	324	3	330	0.667	0.018	0.035
TopHat-Fusion	1	5	8	6	0.111	0.167	0.133
MapSplice	4	8	5	12	0.444	0.333	0.381
BreakFusion	2	142	7	144	0.222	0.014	0.026
SnowShoes-FTD	4	3	5	7	0.444	0.571	0.5
FusionQ	0	0	0	0	0	0	0
FusionHunter	2	4	7	6	0.222	0.333	0.266
ShortFuse	-	-	-	-	-	-	-

A.2 SUPPLEMENTARY FIGURES

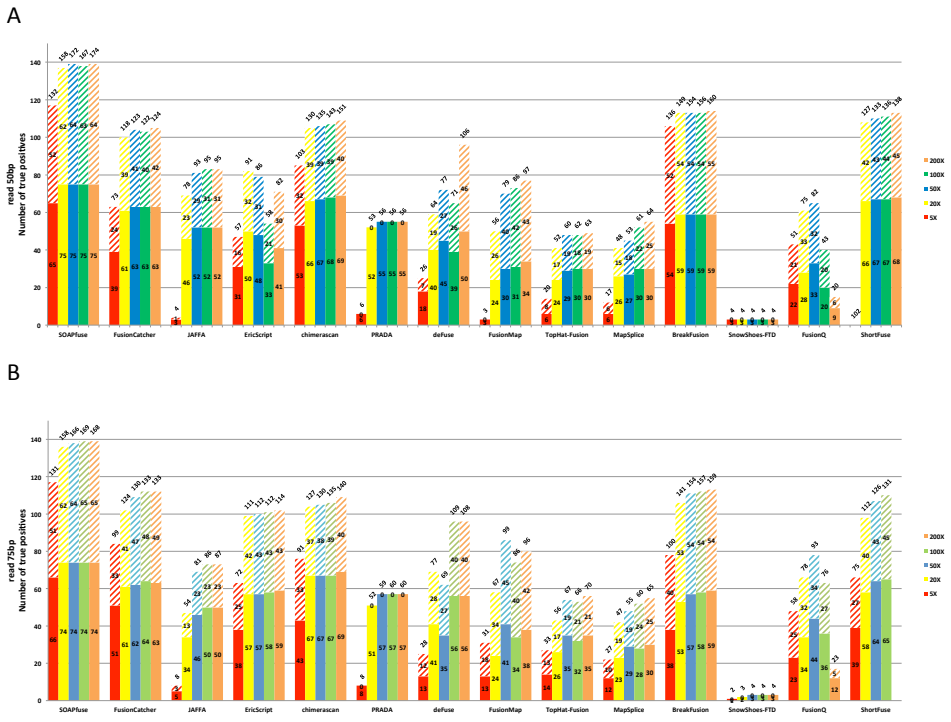


Figure A.1: Fuison transcript detection results for type-1A synthetic datasets. The y-axis bars show the number of true detected positives, among them IE-type and BE-type fusions are shown in solid and slashed rectangles. The total numbers of fusion detections are shown on top of the bars. (A) results for read length 50 bp (B) results for read length 75 bp.

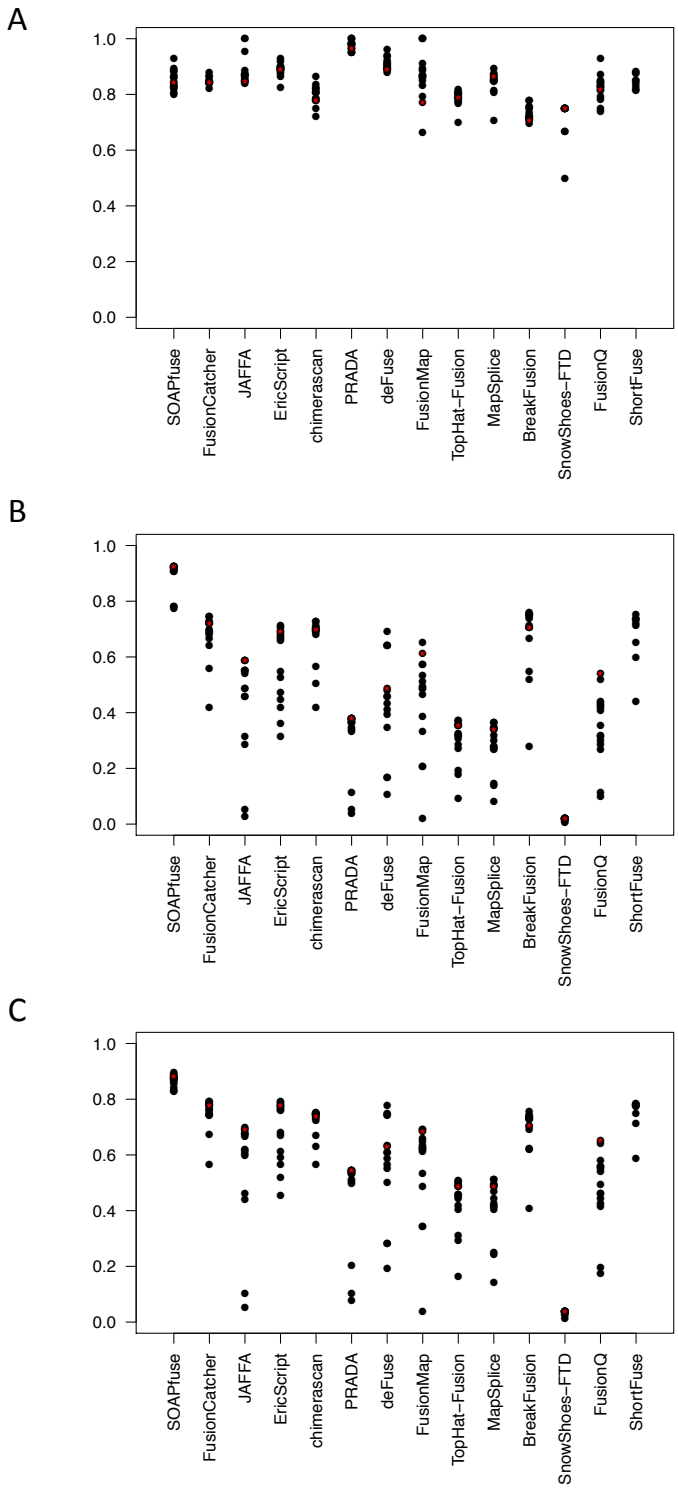


Figure A.2: Precision, recall and F-measure for type-1A synthetic data. Results for read 100 bp & 100X are marked as red star.

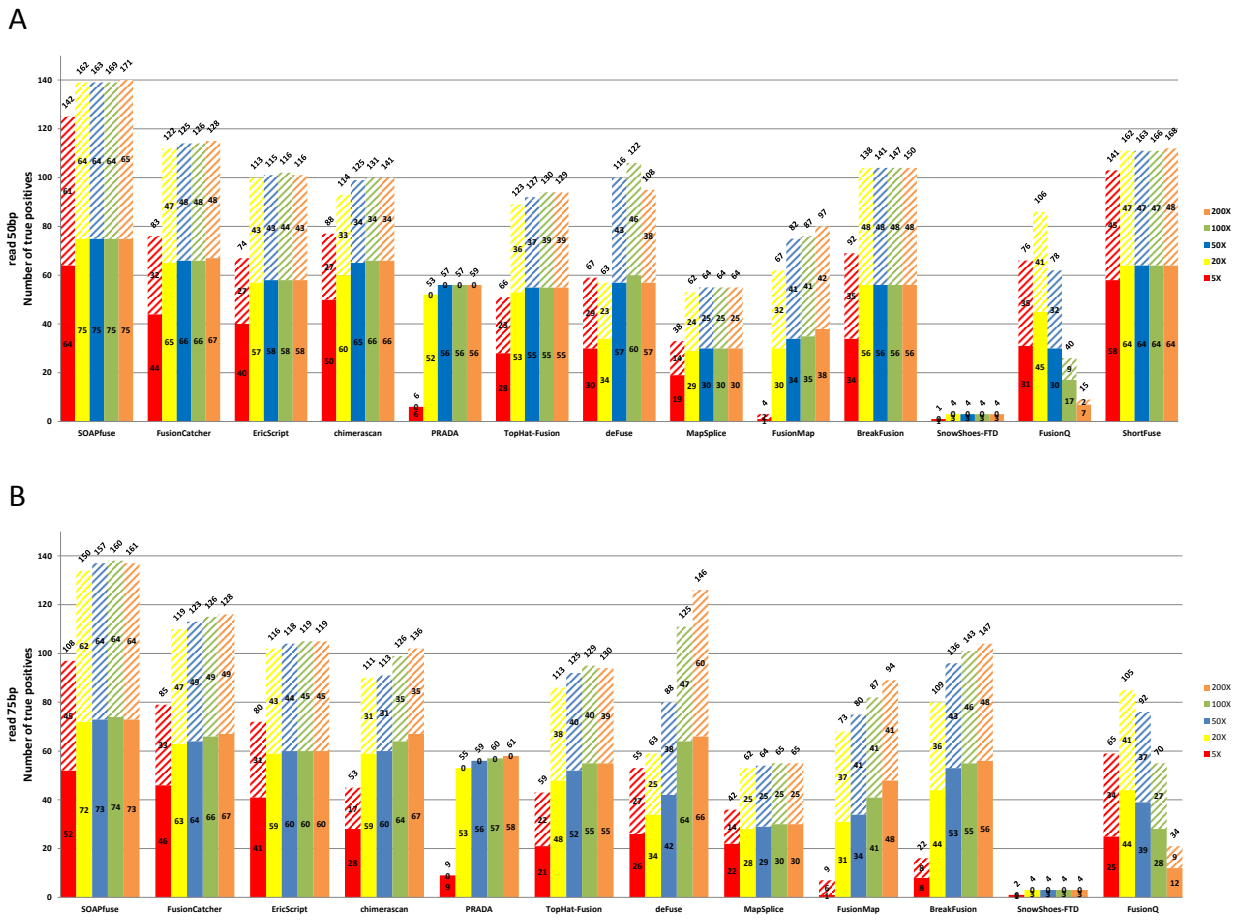


Figure A.3: Fuison transcript detection results for type-1B synthetic datasets. The y-axis bars show the number of true detected positives, among them IE-type and BE-type fusions are shown in solid and slashed rectangles. The total numbers of fusion detections are shown on top of the bars. (A) results for read length 50 bp (B) results for read length 75 bp.

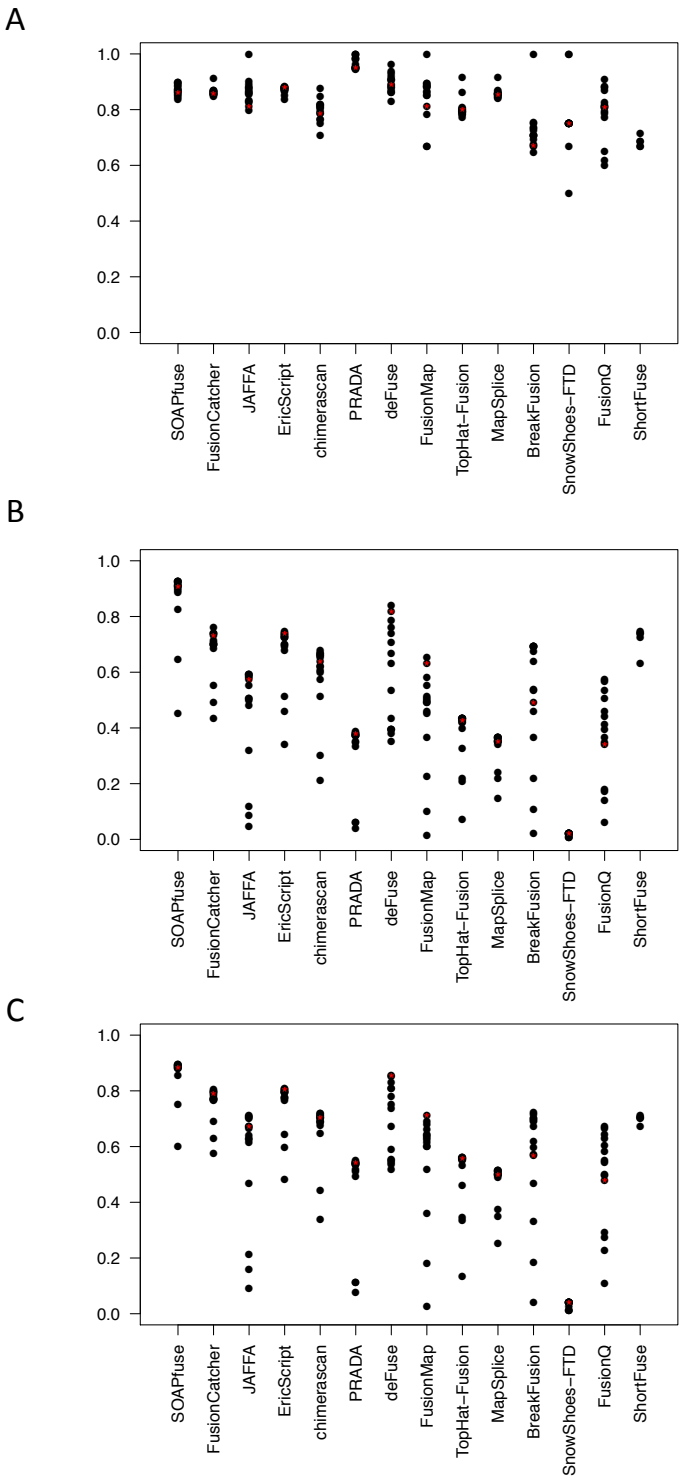


Figure A.4: Precision, recall and F-measure for type-1B synthetic data. Results for read 100 bp & 100X are marked as red star.

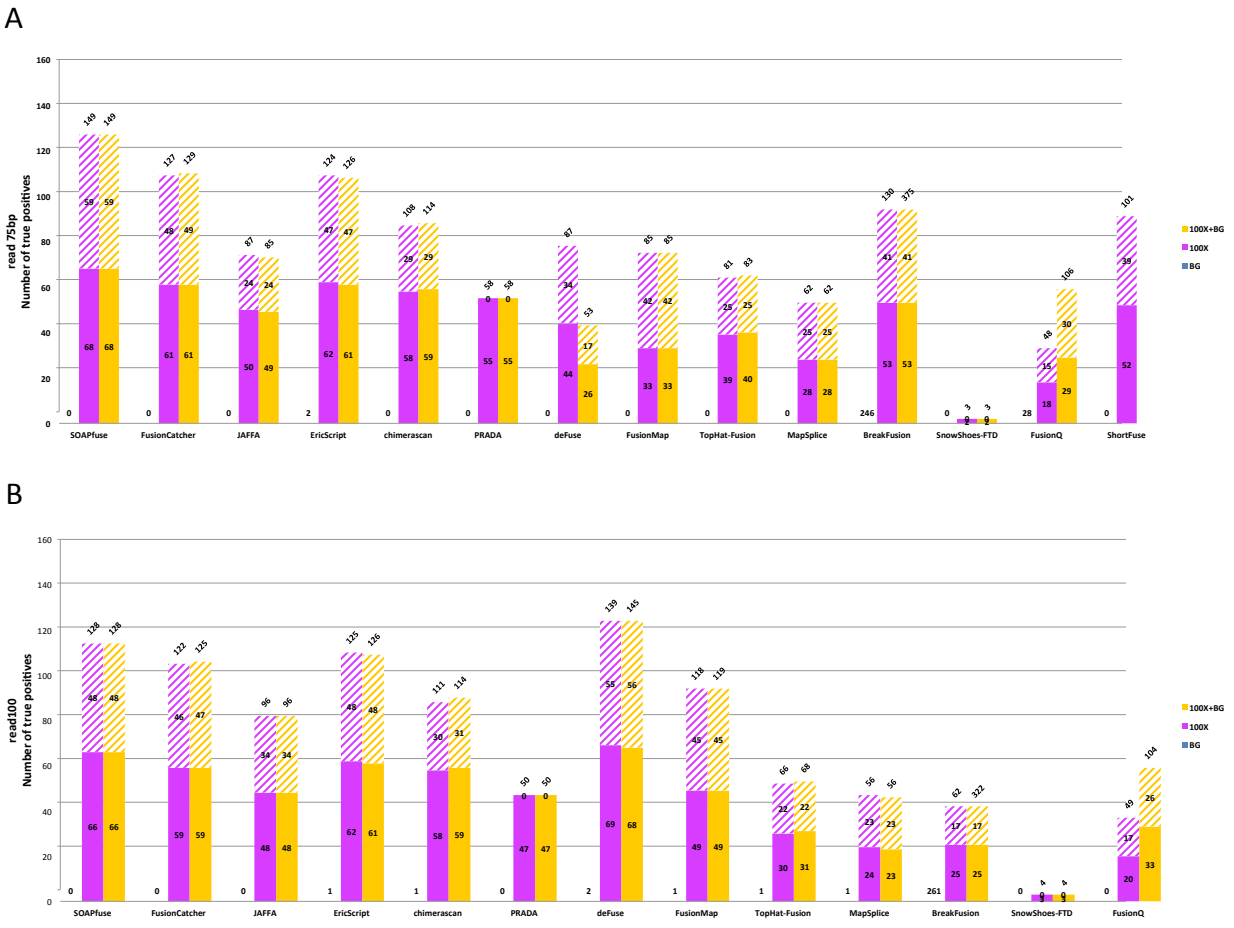


Figure A.5: Fusion transcript detection results for type-2, type-3A and type-3B (lung sample) synthetic datasets on lung sample. The y-axis bars show the number of true detected positives, among them IE-type and BE-type fusions are shown in solid and slashed rectangles. The total numbers of fusion detections are shown on top of the bars. (A) results for read length 75 bp (B) results for read length 100 bp.

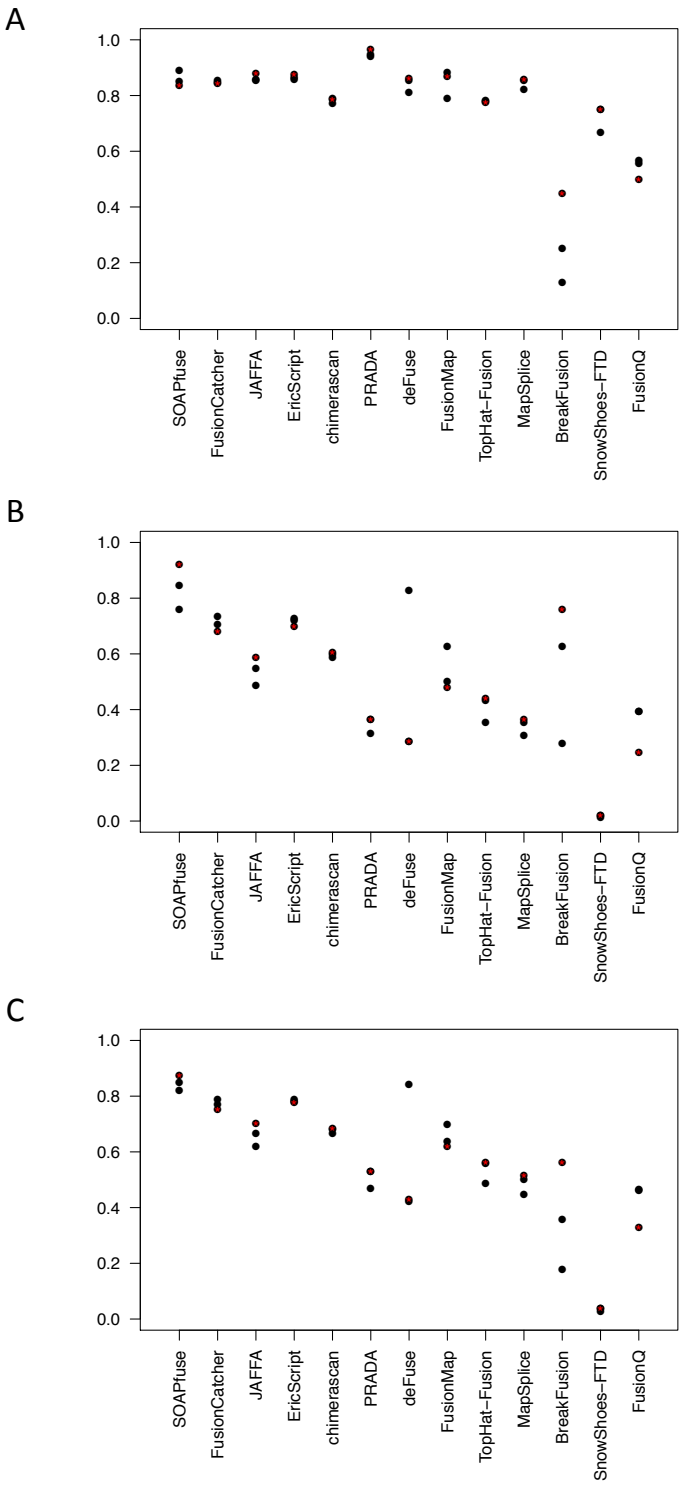


Figure A.6: Precision, recall and F-measure for type-3B (lung sample) synthetic data. Results for read 50 bp are marked as red star.

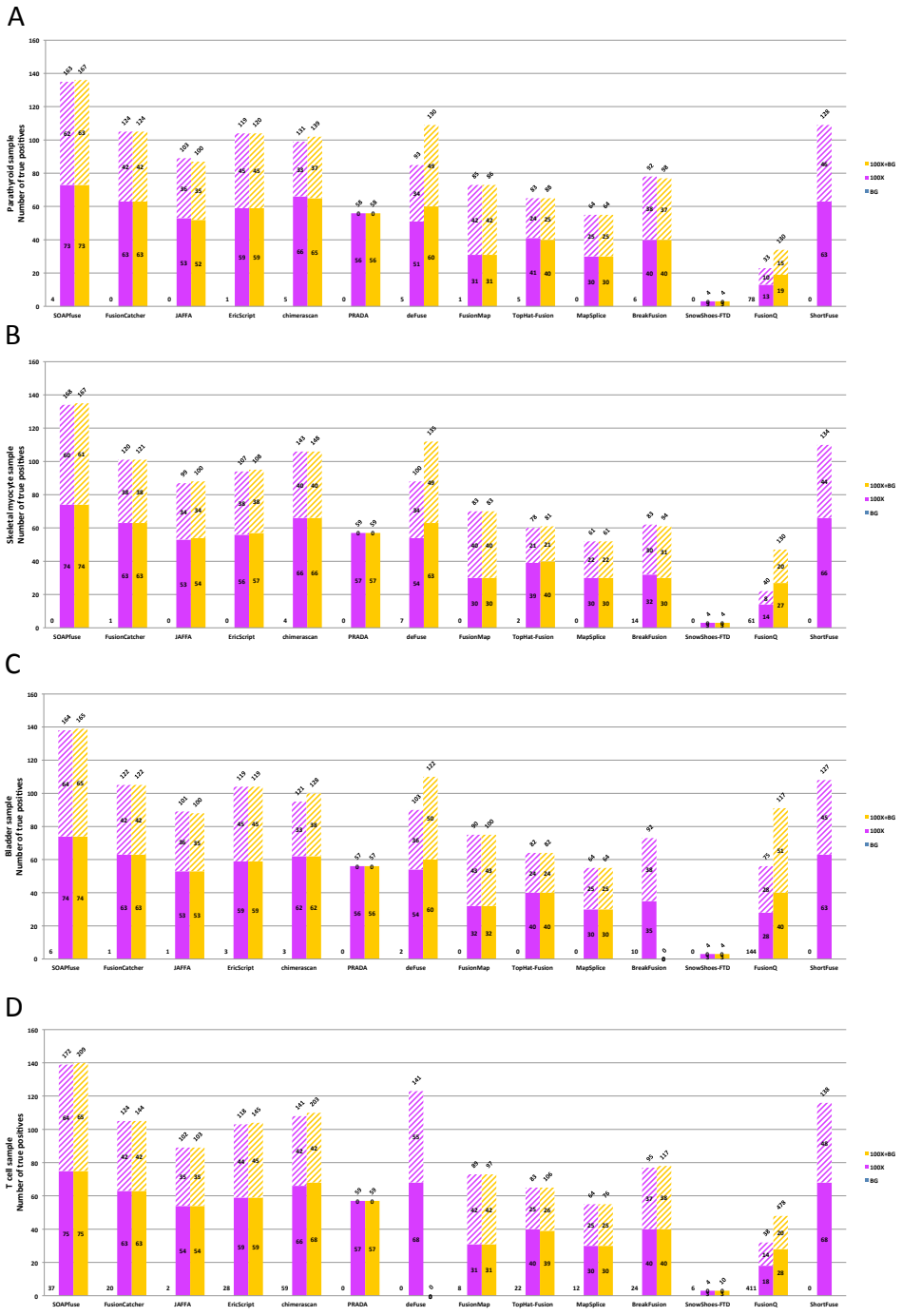


Figure A.7: Fuison transcript detection results for type-2, type-3A and type-3B (lung sample) synthetic datasets on (A) Parathyroid sample (B) Skeletal myocyte sample (C) Bladder sample and (D) T cell sample for read length 50 bp. The y-axis bars show the number of true detected positives, among them IE-type and BE-type fusions are shown in solid and slashed rectangles. The total numbers of fusion detections are shown on top of the bars.

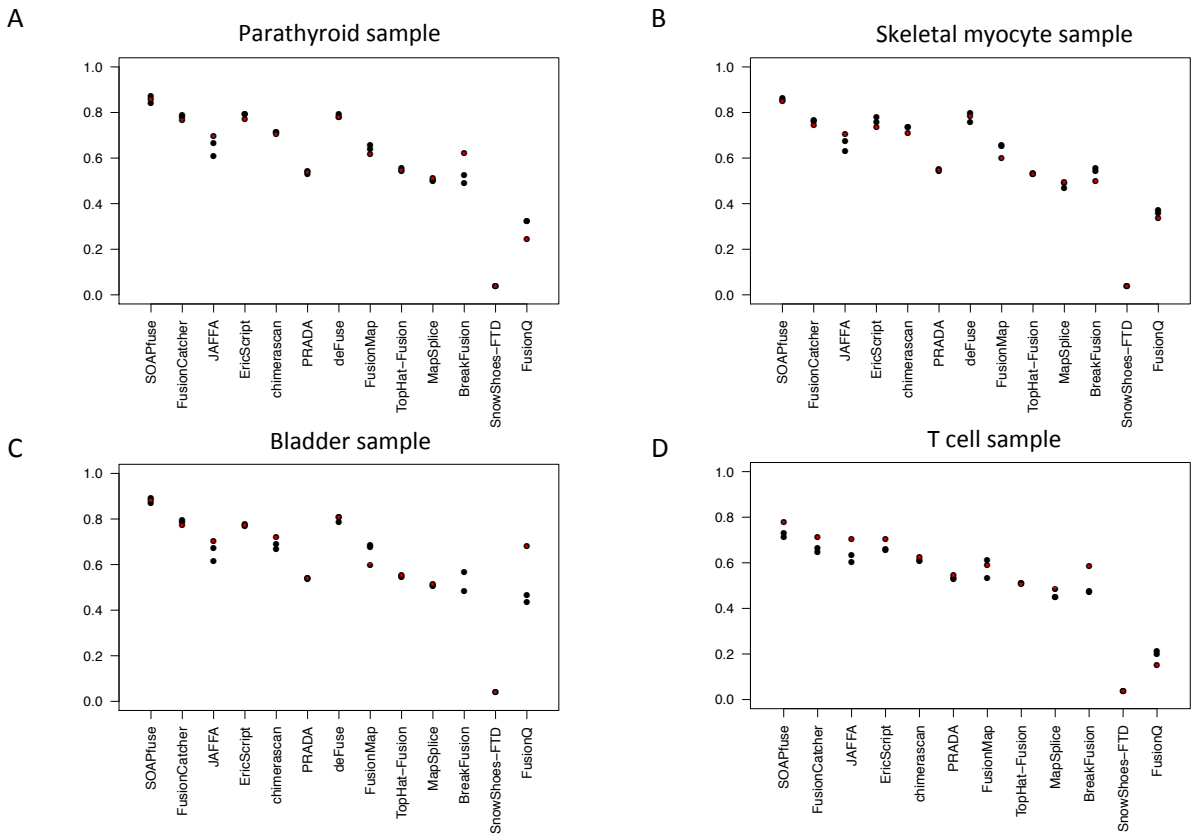


Figure A.8: F-measure for type-3B synthetic data on (A) Parathyroid sample (B) Skeletal myocyte sample (C) Bladder sample and (D) T cell sample. Results for read 50 bp are marked as red star.

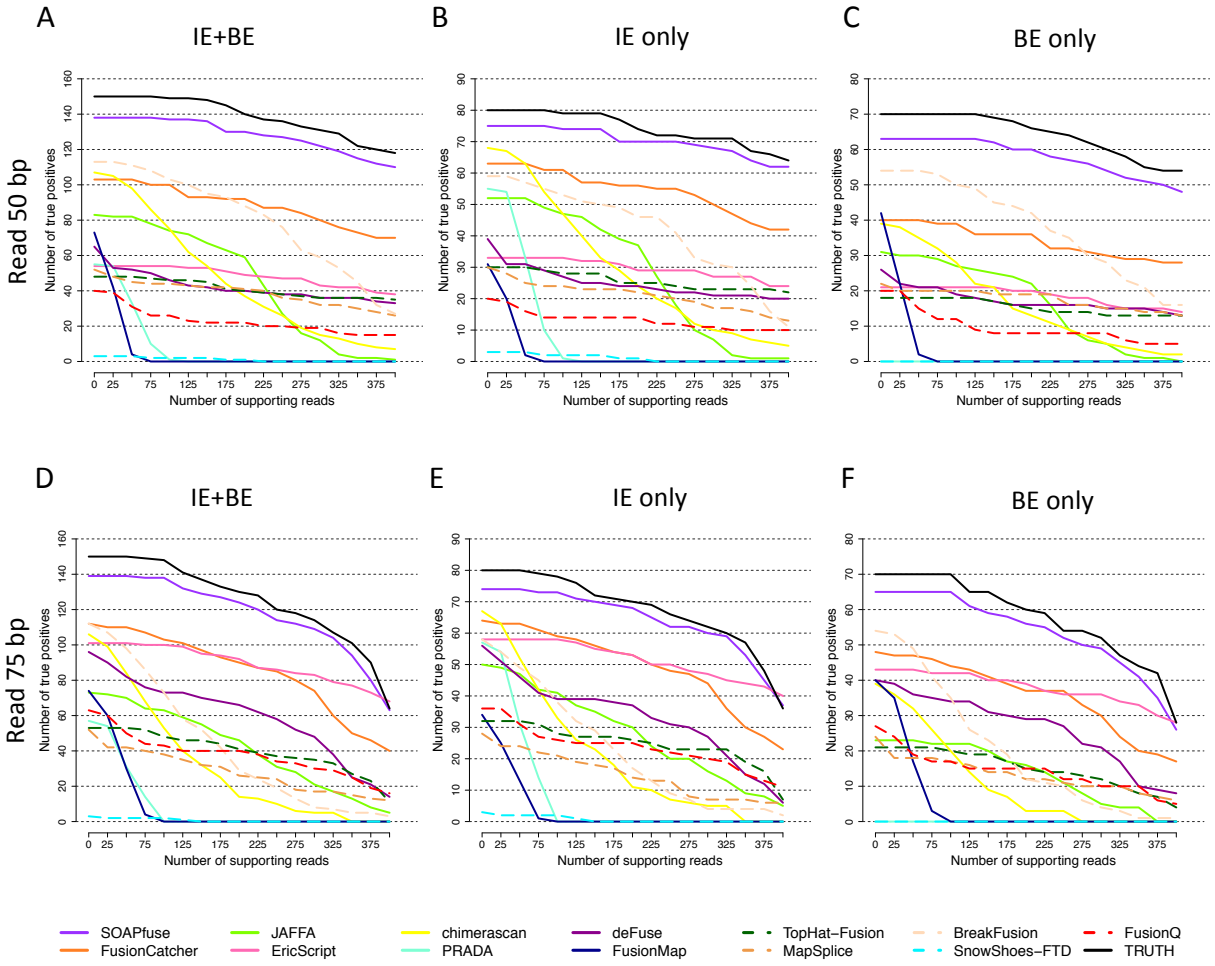


Figure A.9: Distribution plots for alignment performance and similarity across tools for type-1A synthetic data with 50 and 75 bp read length & 100X. Number of true positives (y-axis) with detected supporting reads greater than the threshold on the x-axis. (A-C) results for read 50 bp. (D-F) results for read 75 bp. (A) and (D): results for all 150 true fusion transcripts. (B) and (E): results for only IE-type fusion transcripts. (C) and (F): results for only BE-type fusion transcripts.

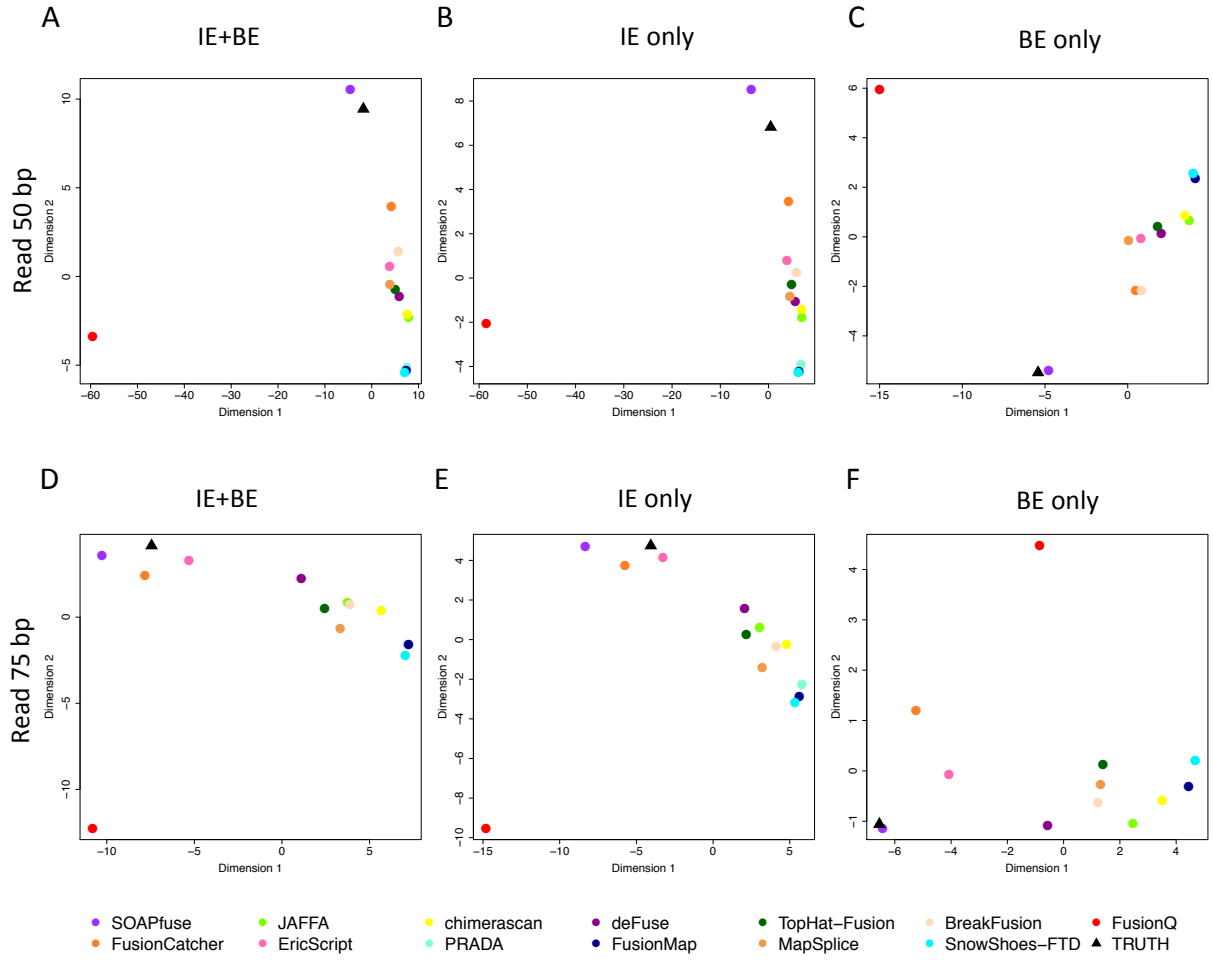


Figure A.10: Multi-dimensional scaling (MDS) plots to demonstrate pairwise similarity of detection results from 14 tools and the underlying truth. (A-C) results for read 50 bp. (D-F) results for read 75 bp. (A) and (D): results for all 150 true fusion transcripts. (B) and (E): results for only IE-type fusion transcripts. (C) and (F): results for only BE-type fusion transcripts.

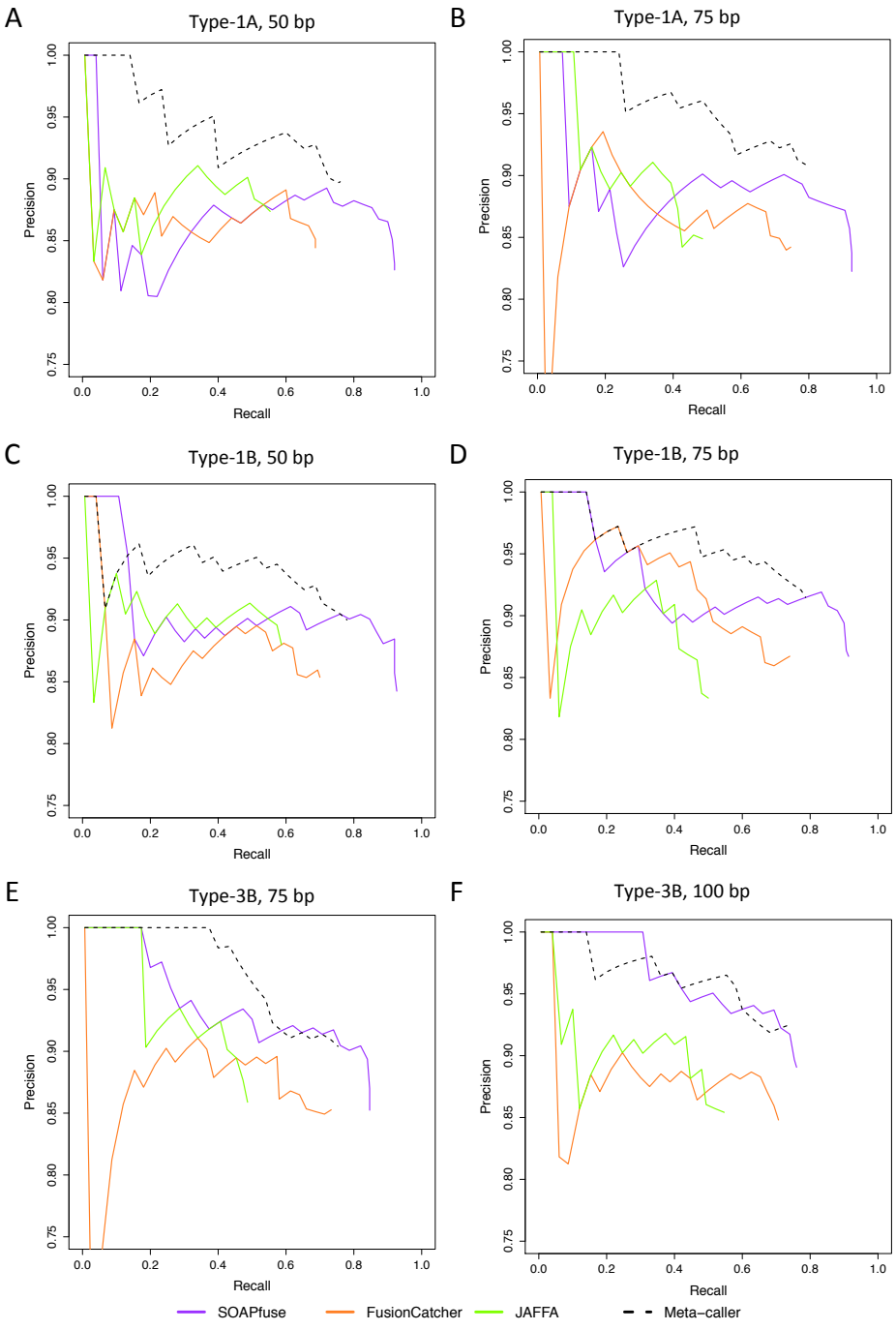


Figure A.11: Precision-recall curves of top 3 performing tools and meta-caller. (A)-(B) Type-1A synthetic data with read length 50 and 75 bp. (C)-(D) Type-1B synthetic data with read length 50 and 75 bp. (E)-(F) Type-3B synthetic data (lung sample) with read length 75 and 100 bp.

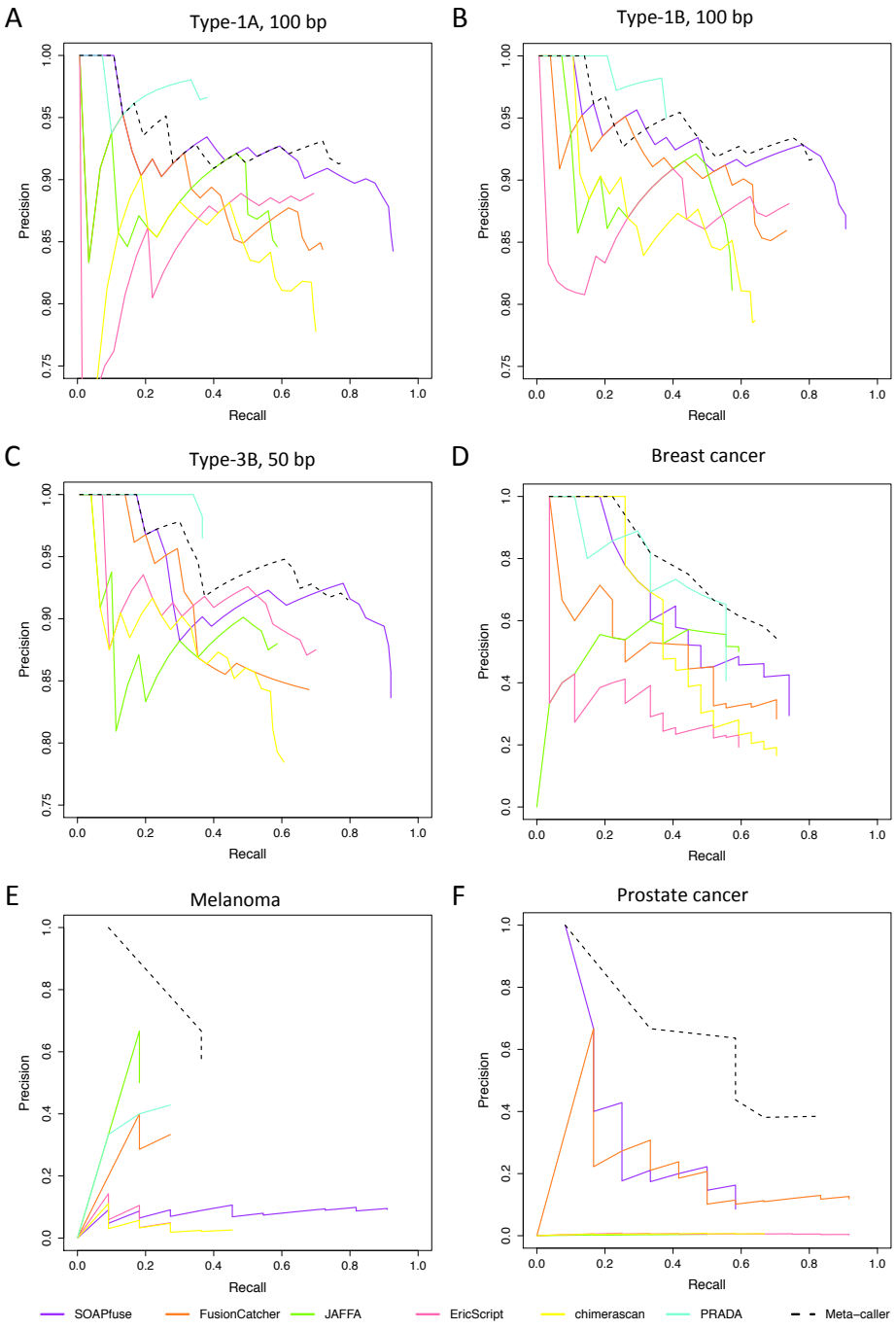


Figure A.12: Precision-recall curves of top 6 performing tools and meta-caller. (A)-(C): Type-1A, type-1B and type-3B (lung sample) synthetic data with 100X coverage and 100, 100 and 50 bp read length respectively. (D)-(F): Three real datasets: breast cancer, melanoma and prostate cancer.

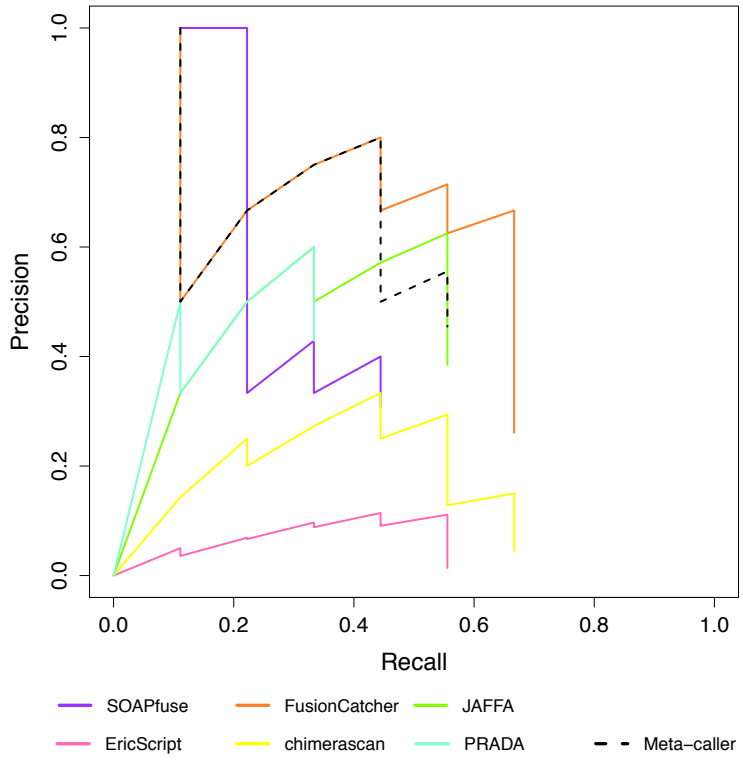


Figure A.13: Precision-recall curves of top-6 performing tools and meta-caller (with majority vote=3) on validation data.

APPENDIX B

SUPPLEMENTARY MATERIAL FOR AIM 2

B.1 SUPPLEMENTARY TABLES

Table B.1: The 2 by 2 table for the top m triplets selected by the full analysis. c_{11}: inside the top m triplets selected by the full analysis, the number of remaining triplets after the filtering; c_{12}: inside the top m triplets selected by the full analysis, the number of filtered out triplets; c_{21}: outside the top m triplets selected by the full analysis, the number of remaining triplets after the filtering; c_{22}: outside the top m triplets selected by the full analysis, the number of filtered out triplets.

m	c_{11}	c_{12}	c_{21}	c_{22}	P-value	Odds ratio
2000	1911	89	12,140,346	910,497,294	$< 10^{-200}$	1610
4000	3781	219	12,138,476	910,497,164	$< 10^{-200}$	1295
6000	5625	375	12,136,632	910,497,008	$< 10^{-200}$	1125
8000	7480	520	12,134,777	910,496,863	$< 10^{-200}$	1079
10000	9310	690	12,132,947	910,496,693	$< 10^{-200}$	1013

Table B.2: Enriched TF binding gene sets for genes controlled by Hog1 from top 100000 triplets selected by meta MLA method.

TF gene	P-value	q-value	Count	Size
Hot1	7.67E-06	4.62E-04	10	63
YPR015C	1.057E-05	4.62E-04	13	111
Hog1p	1.44E-05	4.62E-04	17	187
YGR067C	1.50E-03	3.61E-02	12	159

Table B.3: Enriched GO terms for all the genes from top 500 triplets selected by meta MLA.

ID	P-value	q-value	Odds ratio	Count	Size	Category	Term
GO:0055114	2.08E-17	2.90E-14	3.33	95	327	BP	oxidation-reduction process
GO:0016491	3.28E-12	2.29E-09	2.90	77	285	MF	oxidoreductase activity
GO:0005739	1.69E-08	7.87E-06	1.69	198	1161	CC	mitochondrion
GO:0006412	2.81E-07	9.82E-05	2.29	62	273	BP	translation
GO:0005758	1.95E-06	0.000531	4.34	21	58	CC	mitochondrial intermembrane space
GO:0009060	2.28E-06	0.000531	5.06	18	49	BP	aerobic respiration
GO:0045454	3.15E-06	0.000562	6.16	15	33	BP	cell redox homeostasis
GO:0034599	3.22E-06	0.000562	3.44	26	78	BP	cellular response to oxidative stress
GO:0005737	4.20E-06	0.000568	1.44	338	2271	CC	cytoplasm
GO:0003824	4.40E-06	0.000568	1.98	69	362	MF	catalytic activity
GO:0005783	4.81E-06	0.000568	1.80	92	447	CC	endoplasmic reticulum
GO:0009055	5.15E-06	0.000568	8.20	12	33	MF	electron carrier activity
GO:1902600	5.28E-06	0.000568	9.65	11	28	BP	hydrogen ion transmembrane transport
GO:0009277	1.23E-05	0.001224	3.31	24	85	CC	fungal-type cell wall
GO:0002181	1.62E-05	0.001508	2.50	36	168	BP	cytoplasmic translation
GO:0070469	1.74E-05	0.001519	16.32	8	16	CC	respiratory chain
GO:0006730	2.25E-05	0.001852	11.03	9	15	BP	one-carbon metabolic process
GO:0000324	2.96E-05	0.002200	2.66	30	106	CC	fungal-type vacuole
GO:0005751	3.10E-05	0.002200	21.40	7	12	CC	mitochondrial respiratory chain complex IV
GO:0005576	3.15E-05	0.002200	2.98	25	105	CC	extracellular region
GO:0071555	3.49E-05	0.002326	2.88	26	93	BP	cell wall organization
GO:0005618	4.21E-05	0.002672	3.43	20	68	CC	cell wall
GO:0006164	4.57E-05	0.002692	7.66	10	18	BP	purine nucleotide biosynthetic process
GO:0008152	4.62E-05	0.002692	1.74	79	463	BP	metabolic process
GO:0008121	4.89E-05	0.002735	36.64	6	9	MF	ubiquinol-cytochrome-c reductase activity
GO:0030170	7.22E-05	0.003881	4.20	15	41	MF	pyridoxal phosphate binding
GO:0031505	8.12E-05	0.004203	2.76	25	88	BP	fungal-type cell wall organization
GO:0003735	8.95E-05	0.004471	2.03	46	231	MF	structural constituent of ribosome
GO:0006696	0.000114	0.005485	5.62	11	24	BP	ergosterol biosynthetic process
GO:0005840	0.000136	0.006357	1.82	59	300	CC	ribosome
GO:0005198	0.000147	0.006631	3.84	15	44	MF	structural molecule activity
GO:0005750	0.000172	0.007077	18.32	6	10	CC	mitochondrial respiratory chain complex III
GO:0006122	0.000172	0.007077	18.32	6	11	BP	mitochondrial electron transport, ubiquinol to cytochrome c
GO:0006123	0.000172	0.007077	18.32	6	12	BP	mitochondrial electron transport, cytochrome c to oxygen
GO:0020037	0.000183	0.007316	6.89	9	24	MF	heme binding
GO:0004129	0.000219	0.008490	10.69	7	19	MF	cytochrome-c oxidase activity
GO:0000221	0.000299	0.011287	30.49	5	8	CC	vacuolar proton-transporting V-type ATPase, V1 domain
GO:0006457	0.000341	0.012537	2.51	24	91	BP	protein folding
GO:0005789	0.000416	0.014927	1.67	67	343	CC	endoplasmic reticulum membrane
GO:0005199	0.000461	0.015710	8.55	7	16	MF	structural constituent of cell wall
GO:0051015	0.000461	0.015710	8.55	7	12	MF	actin filament binding
GO:0030479	0.000585	0.019460	2.84	18	59	CC	actin cortical patch
GO:0005886	0.000627	0.020376	1.58	78	446	CC	plasma membrane
GO:0016874	0.000701	0.021772	2.07	32	137	MF	ligase activity
GO:0015035	0.000701	0.021772	6.11	8	17	MF	protein disulfide oxidoreductase activity
GO:0006749	0.000877	0.026082	7.13	7	13	BP	glutathione metabolic process
GO:0015991	0.000877	0.026082	7.13	7	16	BP	ATP hydrolysis coupled proton transport
GO:0016021	0.000913	0.026604	1.34	193	1324	CC	integral component of membrane
GO:0019752	0.001000	0.028535	9.15	6	11	BP	carboxylic acid metabolic process
GO:0046961	0.001161	0.032460	5.43	8	19	MF	proton-transporting ATPase activity, rotational mechanism
GO:0006888	0.001247	0.033519	2.40	21	80	BP	ER to Golgi vesicle-mediated transport
GO:0022625	0.001247	0.033519	2.40	21	99	CC	cytosolic large ribosomal subunit
GO:0030134	0.001301	0.034330	4.59	9	22	CC	ER to Golgi transport vesicle
GO:0016829	0.001493	0.038657	2.30	22	98	MF	lyase activity
GO:0006418	0.001638	0.040880	3.34	12	35	BP	tRNA aminoacylation for protein translation
GO:0006950	0.001638	0.040880	3.34	12	43	BP	response to stress
GO:0008652	0.001745	0.041051	2.22	23	99	BP	cellular amino acid biosynthetic process
GO:0006555	0.001775	0.041051	24.36	4	7	BP	methionine metabolic process
GO:0008177	0.001775	0.041051	24.36	4	6	MF	succinate dehydrogenase (ubiquinone) activity
GO:0009088	0.001775	0.041051	24.36	4	6	BP	threonine biosynthetic process
GO:0006520	0.001833	0.041051	4.89	8	24	BP	cellular amino acid metabolic process
GO:0006099	0.001897	0.041051	3.83	10	29	BP	tricarboxylic acid cycle
GO:0003779	0.001927	0.041051	3.07	13	41	MF	actin binding
GO:0005506	0.001930	0.041051	4.23	9	30	MF	iron ion binding
GO:0004601	0.001938	0.041051	7.32	6	11	MF	peroxidase activity
GO:0006090	0.001938	0.041051	7.32	6	11	BP	pyruvate metabolic process
GO:0051082	0.002154	0.044721	2.32	20	81	MF	unfolded protein binding
GO:0005385	0.002175	0.044721	10.16	5	8	MF	zinc ion transmembrane transporter activity

Table B.4: Pearson correlations of some important genes in each single study.

Gene A	Gene B	Causton	Gasch	Rosetta	GSE60613	GSE11452
IDH2	CDC19	0.22	-0.09	0.04	0.08	-0.32
IDH2	BDH2	0.41	0.41	-0.07	-0.31	0.10
IDH2	ENO2	-0.02	0.30	-0.04	0.21	-0.36
IDH2	PDC1	-0.19	-0.05	0.10	-0.07	-0.27
IDH2	PDC5	0.62	-0.08	0.16	-0.12	0.10
BDH2	CDC19	0.14	-0.45	-0.29	0.41	-0.15
BDH2	ENO2	-0.06	-0.17	-0.31	0.33	-0.24
BDH2	PDC1	-0.05	-0.47	-0.42	0.58	-0.07
BDH2	PDC5	0.44	-0.52	-0.45	0.33	0.36

B.2 SUPPLEMENTARY FIGURES

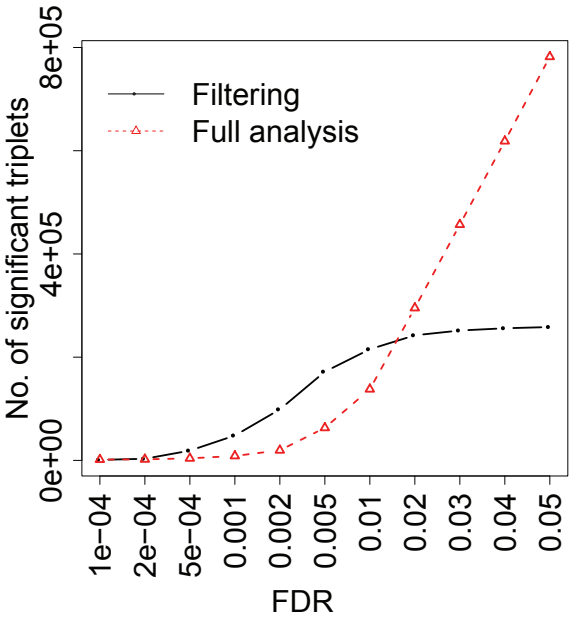


Figure B.1: Controlled by a certain FDR, the detected number of significant triplets by both filtering and full analysis pipelines.

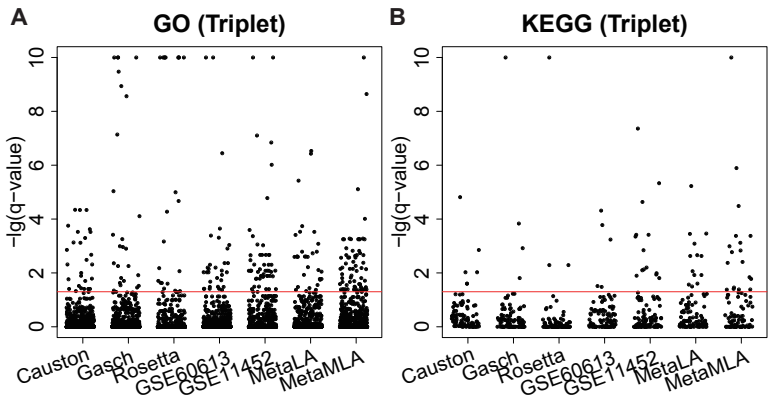


Figure B.2: Jitter plot of the q-values of the enriched gene sets for all the genes from top 500 triplets using the minus log 10 scale. **(A)** is for GO terms and **(B)** is for KEGG pathways. The values larger than 10 are cut off to be 10. The horizontal line is $y = -\lg(0.05)$.

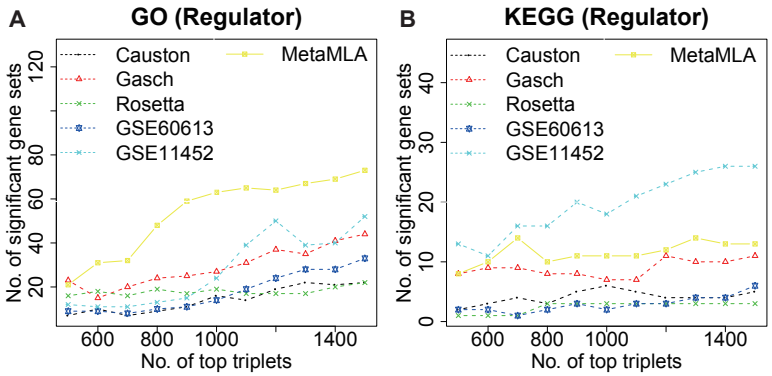


Figure B.3: The number of enriched gene sets for Z genes from different numbers of top triplets detected by meta and single analysis. **(A)** is for GO terms and **(B)** is for KEGG pathways.

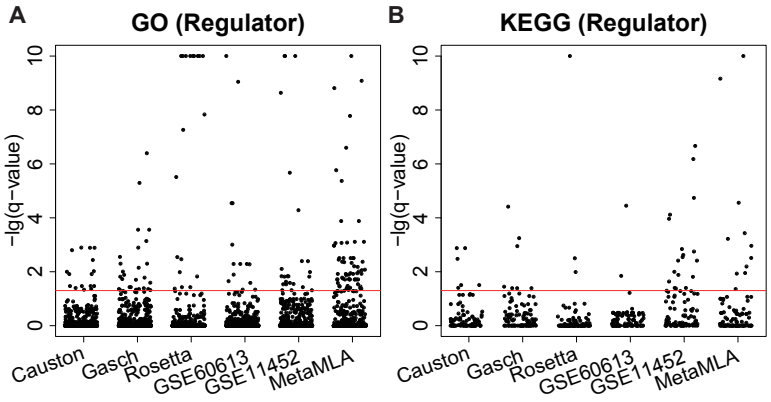


Figure B.4: Jitter plot of the q-values of the enriched gene sets for the Z genes from top 1000 triplets using the minus log 10 scale. **(A)** is for GO terms and **(B)** is for KEGG pathways. The values larger than 10 are cut off to be 10. The horizontal line is $y = -\lg(0.05)$.

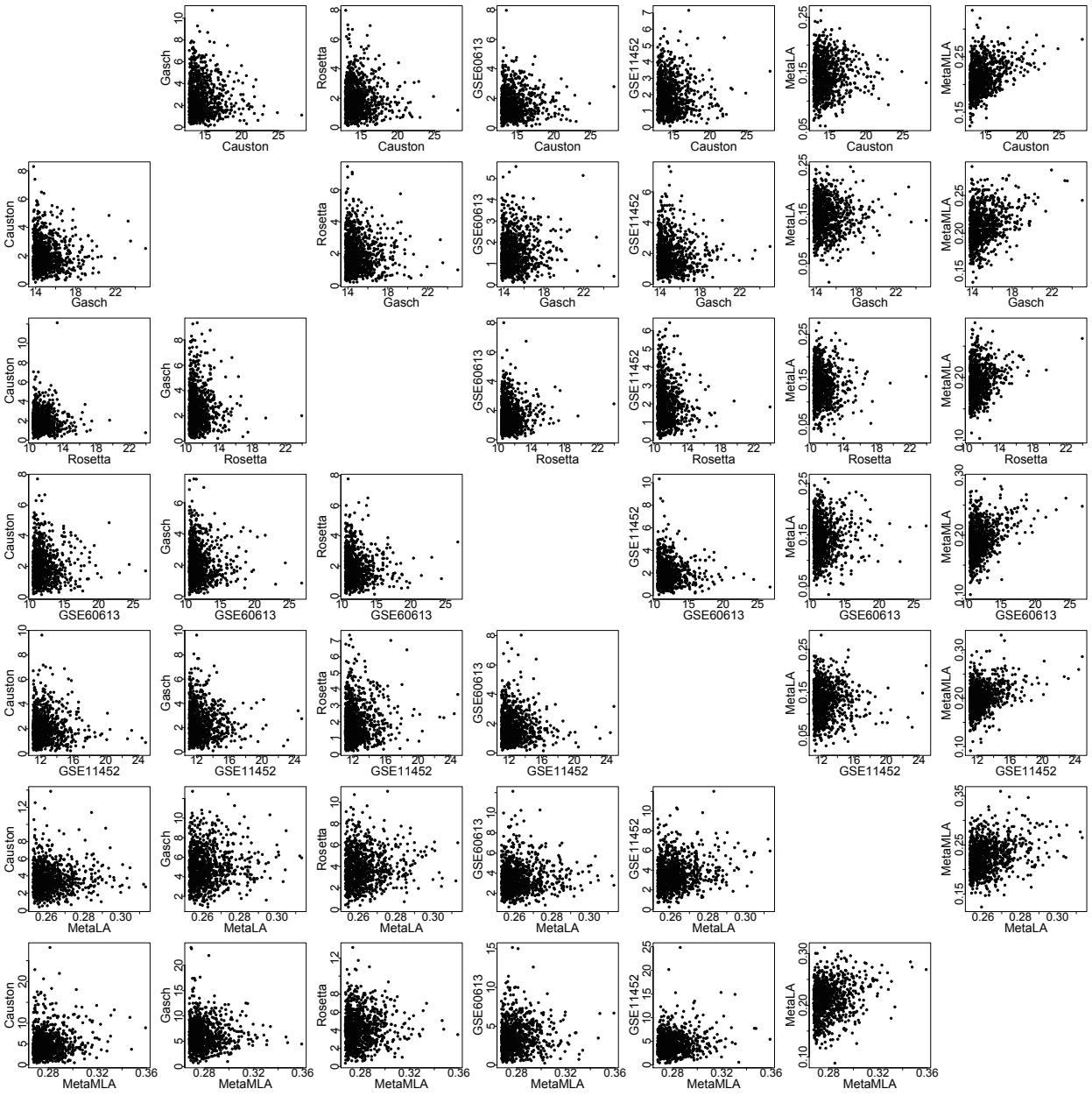


Figure B.5: Top 1000 triplets' test statistics correlation between pairwise single studies and meta-analysis. Each dot represents a triplet selected by study in the x-axes and its test statistics in both x and y axes.

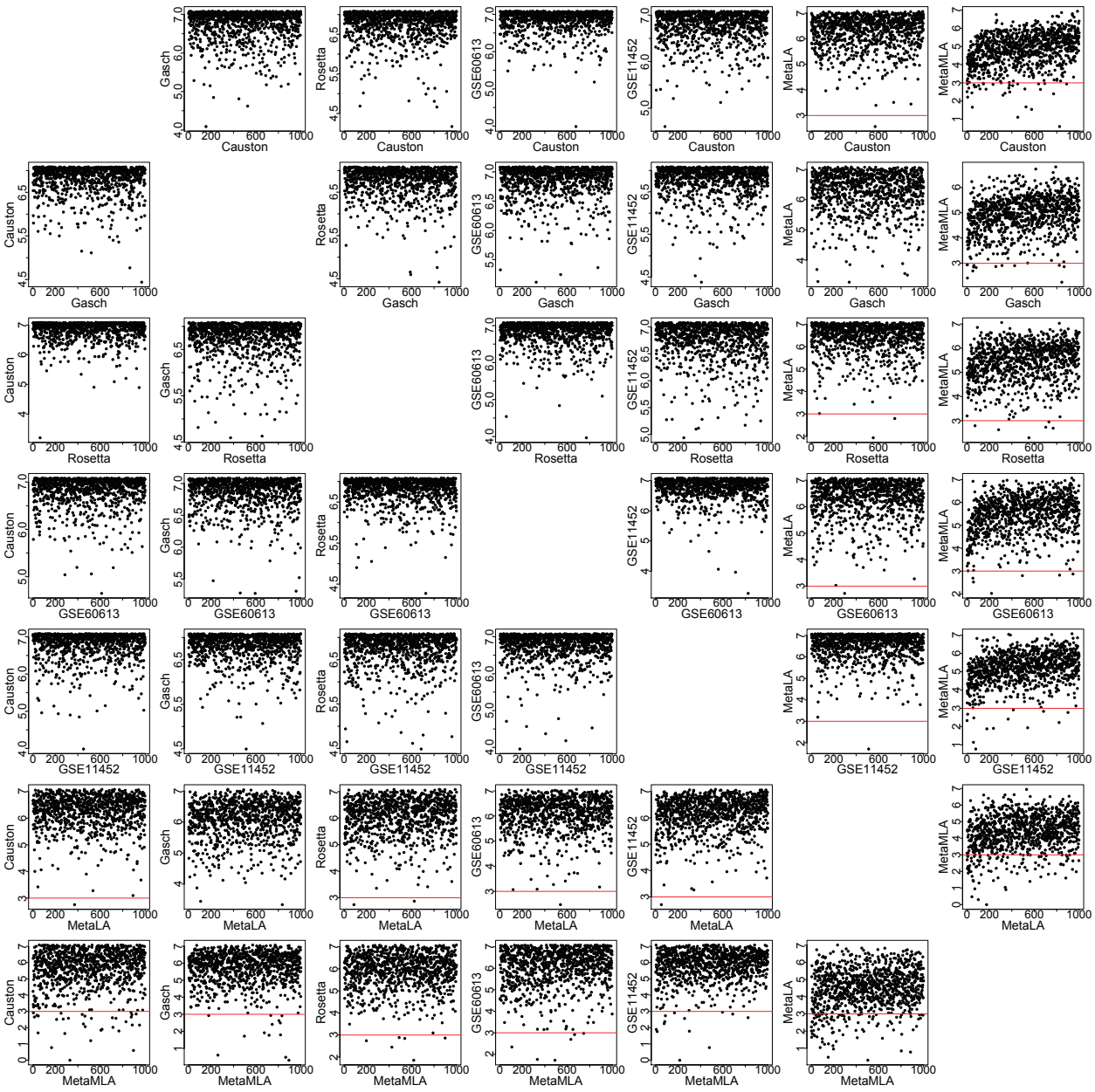


Figure B.6: Top 1000 triplets' rank correlation between pairwise single studies and meta-analysis. Each dot represents a triplet selected by study in the x-axes and its test statistics in both x and y axes. Dots under the red line represent the common triplets selected by both studies.

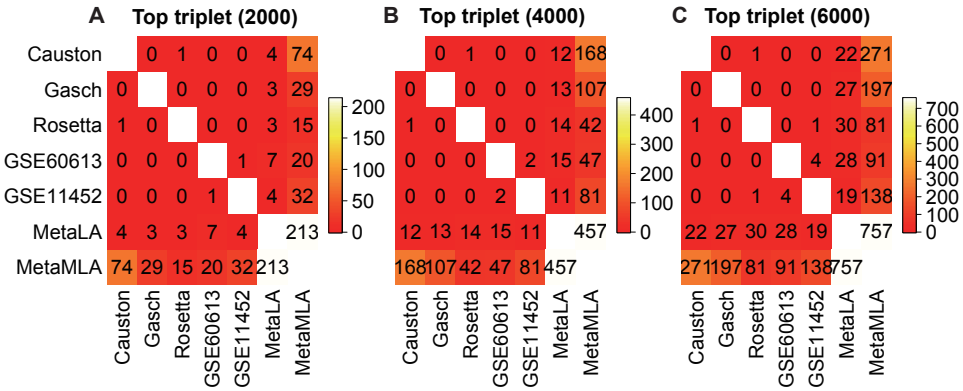


Figure B.7: Number of overlapped triplets among meta and single analysis for different top number of significant triplets. (A) is for top 2000 significant triplets; (B) is for top 4000 significant triplets; (C) is for top 6000 significant triplets.

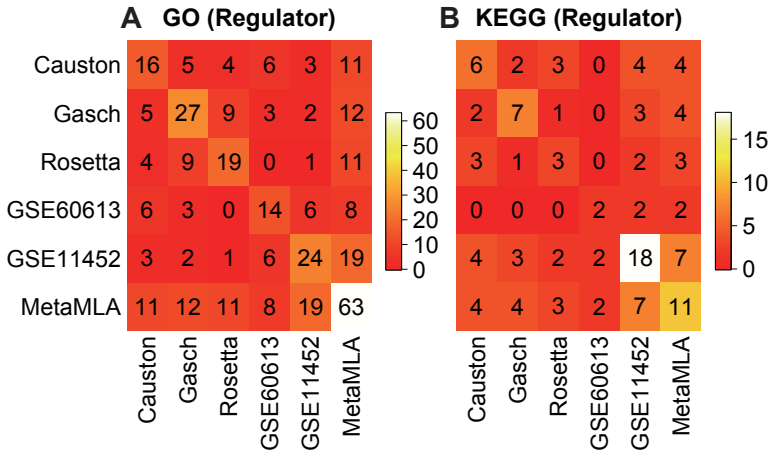


Figure B.8: Overlap of meta and single analysis. **(A)** is for the number of overlapped enriched GO terms using Z genes from top 1000 triplets are for gene set enrichment analysis; **(B)** is for the number of overlapped enriched KEGG pathways using Z genes from top 1000 triplets for gene set enrichment analysis.

APPENDIX C

SUPPLEMENTARY MATERIAL FOR AIM 3

C.1 SUPPLEMENTARY TABLES

Table C.1: Parameter setting for synthetic data.

Symbol	Setting	Note
S	4	Number of studies
K	1	Number of bicluster layers
p	100, 100, 80 and 70	Number of samples for each study
n	500	Number of genes
μ_0	0, 0, 0 and 0	Background mean value for each study
σ_0	1, 1, 1 and 1	Background SD value for each study, adjusted by simulation
μ	1.5, 1.5, 1.5 and 1.5	Bicluster mean value effect for each study
α_{min}	0.2, 0.2, 0.2 and 0.2	Bicluster row effect min value for each study
α_{max}	0.5, 0.5, 0.5 and 0.5	Bicluster row effect max value for each study
β_{min}	0.2, 0.2, 0.2 and 0.2	Bicluster column effect min value for each study
β_{max}	0.5, 0.5, 0.5 and 0.5	Bicluster column effect max value for each study
gn_1	50	Bicluster consistent gene size, 30 if gn_2 or gn_3 is non-zero
gn_2	0 or 10	Bicluster prevalent gene size
gn_3	0 or 10	Bicluster study-specific gene size
sn	40, 30, 25 and 25	Bicluster sample size for each study

Table C.2: Number of genes and samples of the biclusters detected from five breast cancer cohorts.

Bicluster	Gene size	GSE2034 sample size	GSE7390 sample size	GSE11121 sample size	TCGA sample size	METABRIC sample size
1	291	260	164	161	533	1981
2	291	260	164	161	533	1981
3	291	260	164	161	533	1981
4	291	260	164	161	533	1981
5	291	260	164	161	533	1981
6	291	260	164	161	533	1981
7	291	260	164	161	533	1981
8	291	260	164	161	533	1981
9	291	104	65	161	112	567
10	291	72	48	54	487	1979
11	291	187	36	32	528	1780
12	291	47	41	69	105	423
13	291	98	114	69	425	784
14	291	175	109	59	215	900
15	291	108	94	52	407	959

Table C.3: Association p-values between bicluster sample splitting and clinical information.

Bicluster	METABRIC ER +/- p-value	METABRIC survival p-value
9	$<$ 2.2E-16 (ER - enriched)	1.14E-14
10	0.05596 (ER + enriched)	0.371
11	$<$ 2.2E-16 (ER + enriched)	1.36E-3
12	$<$ 2.2E-16 (ER - enriched)	7.13E-11
13	3.02E-15 (ER + enriched)	8.81E-05
14	2.01E-04 (ER + enriched)	3.55E-04
15	$<$ 2.2E-16 (ER - enriched)	7.77E-16

Table C.4: Association between breast cancer subtypes and 12th bicluster sample splitting for METABRIC cohort.

Breast cancer subtype	# samples outside the bicluster	# samples inside the biclsuter
Luminal A	713	6
Luminal B	456	34
Her2-enriched	138	100
Basal-like	57	271
Normal	188	12
NC	6	0

Table C.5: Number of significantly enriched pathways for each breast cancer biclusters (FDR=5%).

Bicluster	Biocarta	KEGG	Reactome	Chemical and genetic perturbations	GO BP	GO CC	GO MF
9	0	0	0	37	0	0	0
10	1	24	32	279	285	30	22
11	0	0	0	29	0	0	0
12	1	2	17	239	27	6	3
13	0	4	10	299	210	20	22
14	0	3	9	307	90	28	16
15	0	4	61	222	80	18	1

C.2 SUPPLEMENTARY FIGURES

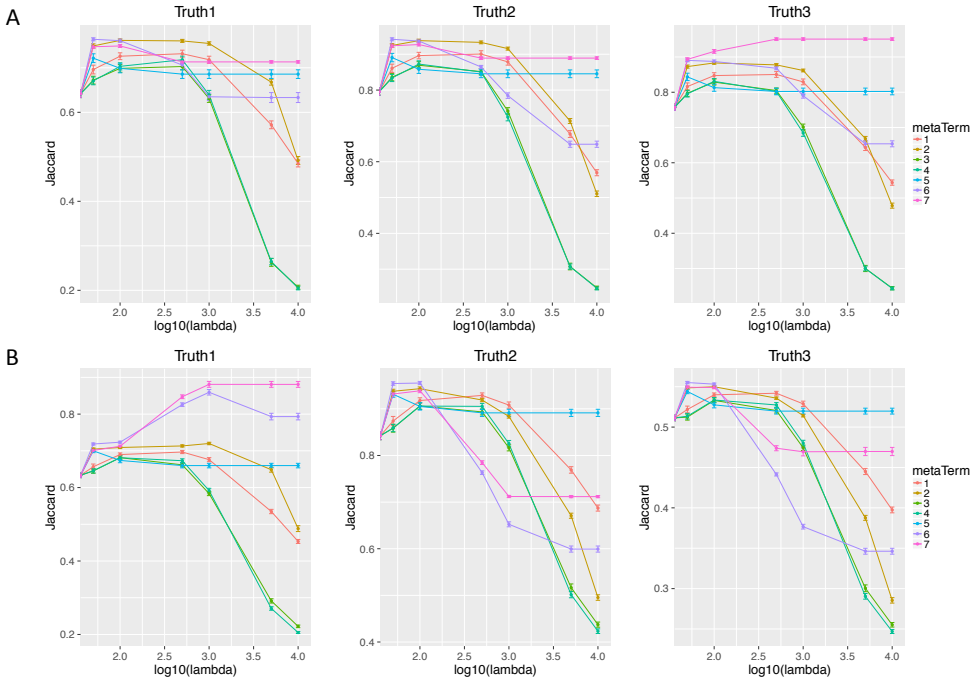


Figure C.1: Comparison of seven meta terms performance on (A) consistent + prevalent gene simulation; (B) consistent + study-specific gene simulation without pruning steps. For consistent + prevalent gene simulation, truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and prevalent genes inside each study, and truth 3 means using consistent + prevalent genes as the truth for all studies. For meta analysis, truth 3 is preferred that both consistent and prevalent genes are preferred to be detected. For consistent + study-specific gene simulation, truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and study-specific genes inside each study, and truth 3 means using consistent + study-specific genes as the truth for all studies. For meta analysis, truth 1 is preferred that only common genes are preferred but not the study-specific genes.

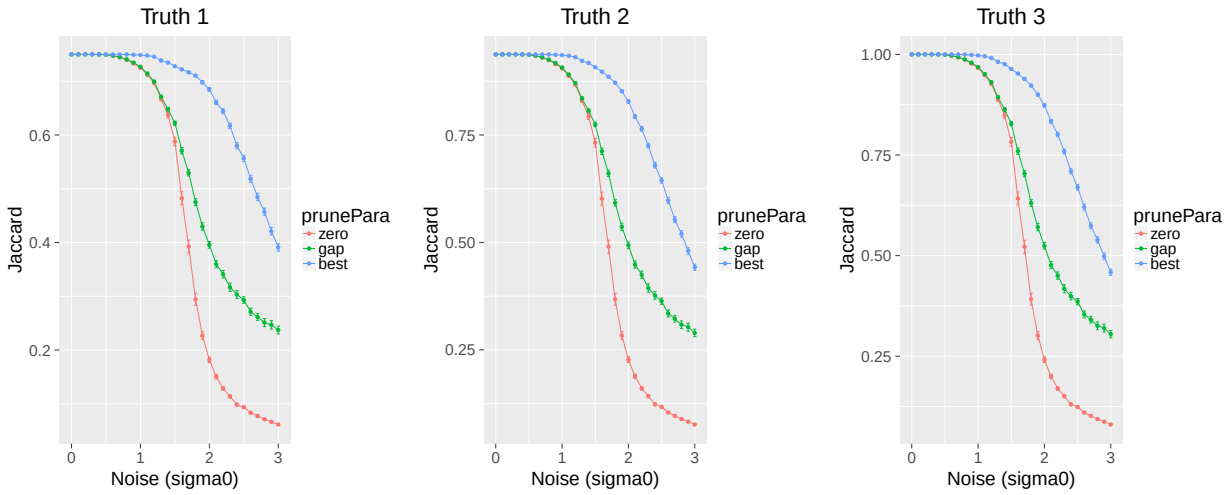


Figure C.2: Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by gap statistic (green line), and the best performance within the searching space (blue line) in consistent + prevalent gene simulation. Truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and prevalent genes inside each study, and truth 3 means using consistent + prevalent genes as the truth for all studies. For meta analysis, truth 3 is preferred that both consistent and prevalent genes are preferred to be detected.

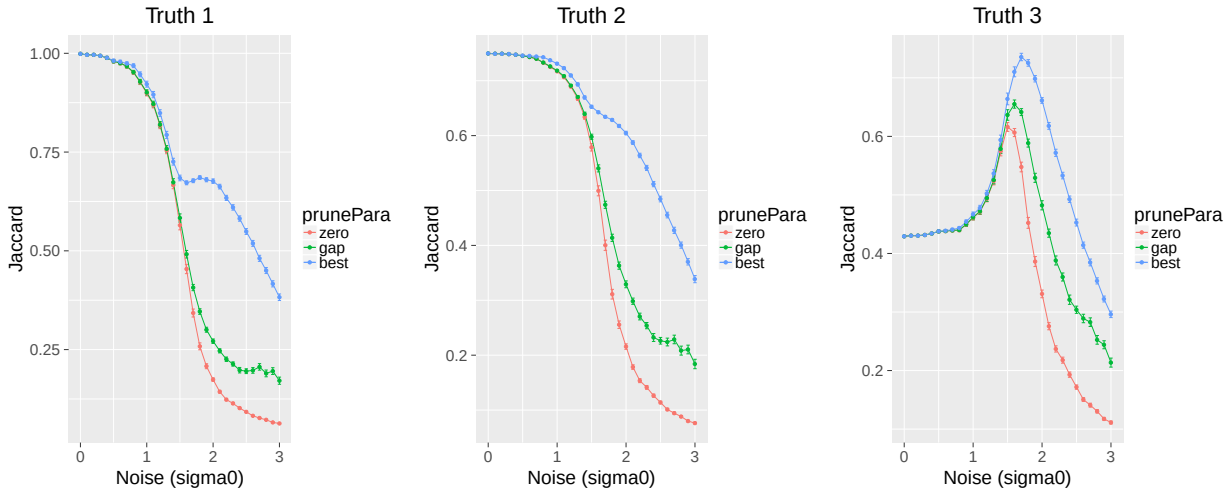


Figure C.3: Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by gap statistic (green line), and the best performance within the searching space (blue line) in consistent + study-specific gene simulation. Truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and study-specific genes inside each study, and truth 3 means using consistent + study-specific genes as the truth for all studies. For meta analysis, truth 1 is preferred that only common genes are preferred but not the study-specific genes.

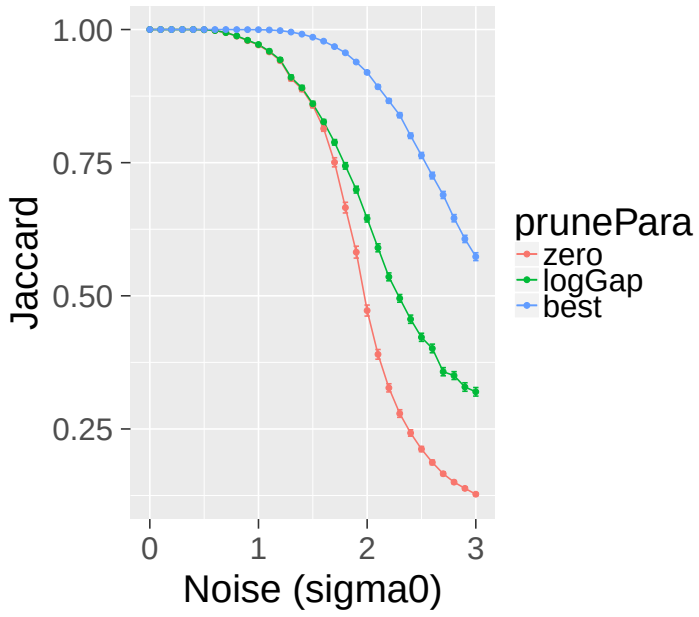


Figure C.4: Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by log of gap statistic (green line), and the best performance within the searching space (blue line).

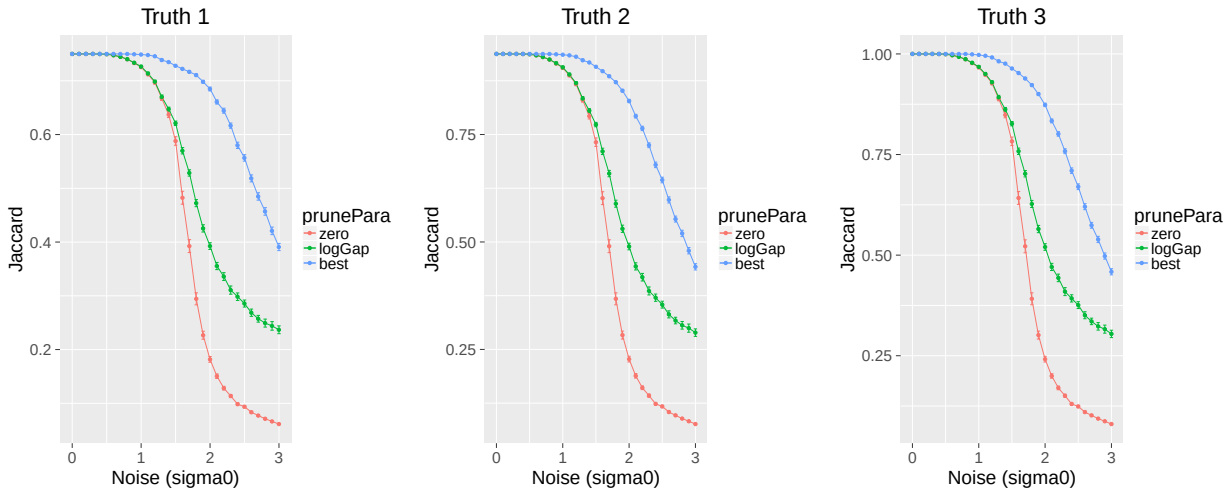


Figure C.5: Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by log of gap statistic (green line), and the best performance within the searching space (blue line) in consistent + prevalent gene simulation. Truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and prevalent genes inside each study, and truth 3 means using consistent + prevalent genes as the truth for all studies. For meta analysis, truth 3 is preferred that both consistent and prevalent genes are preferred to be detected.

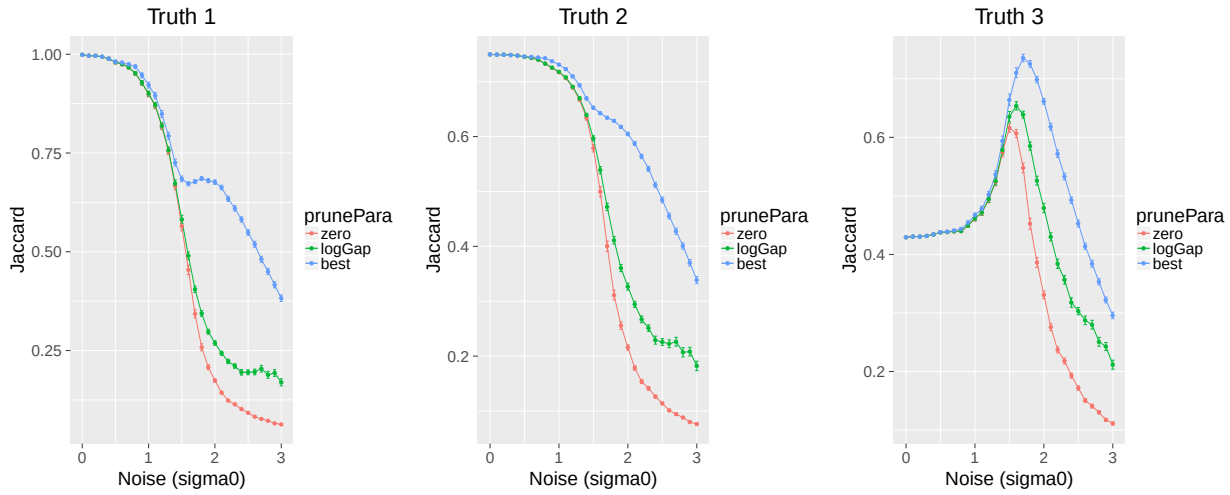


Figure C.6: Performance of bicluster detection without pruning step (red line), with pruning step where parameters are selected by log of gap statistic (green line), and the best performance within the searching space (blue line) in consistent + study-specific gene simulation. Truth 1 means only using consistent genes as the underlying truth, truth 2 means using exactly the consistent genes and study-specific genes inside each study, and truth 3 means using consistent + study-specific genes as the truth for all studies. For meta analysis, truth 1 is preferred that only common genes are preferred but not the study-specific genes.

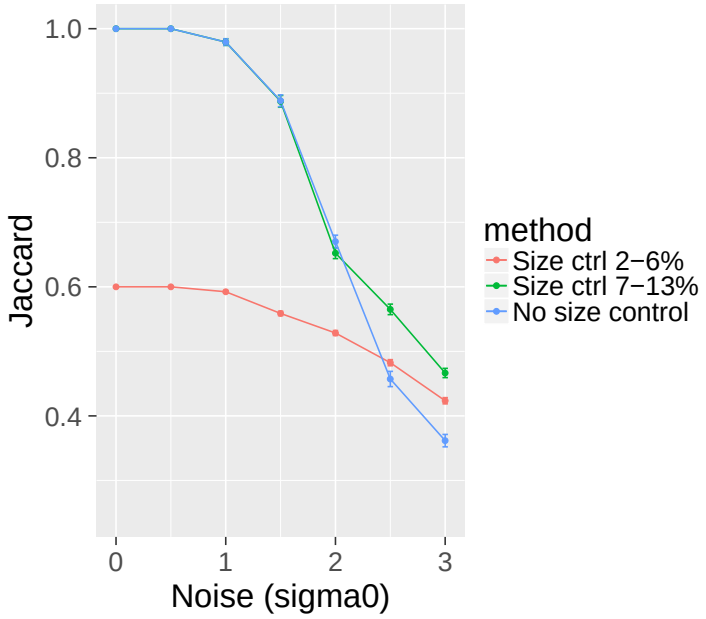


Figure C.7: Performance of bicluster detection with incorrect (2% to 6%, red line), correct (7% to 13%, green line), and without gene size control (blue line). The underlying truth of gene selection rate is 10%

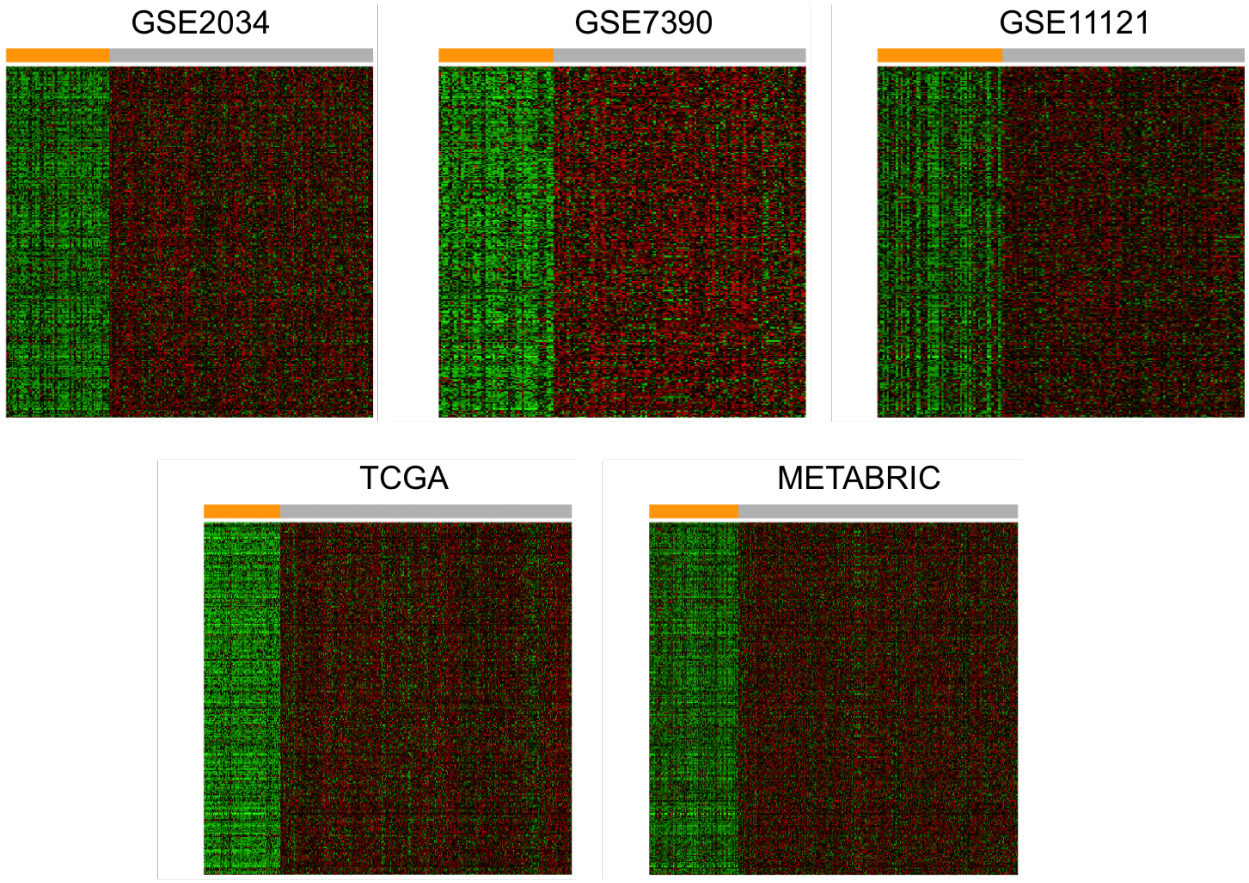


Figure C.8: Meta-bicluster detect from 5 breast cancer cohorts. Orange bar: samples inside the bicluster; grey bar: samples outside the bicluster.

BIBLIOGRAPHY

Francesco Abate, Andrea Acquaviva, Giulia Paciello, Carmelo Foti, Elisa Ficarra, Alberto Ferrarini, Massimo Delledonne, Ilaria Iacobucci, Simona Soverini, Giovanni Martinelli, et al. Bellerophontes: an rna-seq data analysis framework for chimeric transcripts discovery based on accurate fusion model. *Bioinformatics*, 28(16):2114–2121, 2012.

Paula M Alepuz, Eulàlia de Nadal, Meritxell Zapater, Gustav Ammerer, and Francesc Posas. Osmostress-induced transcription by hot1 depends on a hog1-mediated recruitment of the rna pol ii. *The EMBO journal*, 22(10):2433–2442, 2003.

Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.*, 46(3):175–185, 1992.

Dhammika Amaratunga and Javier Cabrera. Analysis of data from viral dna microchips. *J. Am. Stat. Assoc.*, 96(456):1161–1170, 2001.

Yan W. Asmann, Asif Hossain, Brian M. Necela, Sumit Middha, Krishna R. Kalari, Zhifu Sun, High-Seng Chai, David W. Williamson, Derek Radisky, Gary P. Schroth, Jean-Pierre A. Kocher, Edith A. Perez, and E. Aubrey Thompson. A novel bioinformatics pipeline for identification and characterization of fusion transcripts in breast cancer and normal cell lines. *Nucleic Acids Research*, 39(15):e100, 2011.

Monya Baker. De novo genome assembly: what every biologist should know. *Nature methods*, 9(4):333, 2012.

David J. Barnes and Junia V. Melo. Cytogenetic and molecular genetic aspects of chronic myeloid leukaemia. *Acta Haematol*, 108:180–202, 2002.

Marco Beccuti, Matteo Carrara, Francesca Cordero, Susanna Donatelli, and Raffaele A Calogero. The structure of state-of-art gene fusion-finder algorithms. *Genome Bioinformatics*, 1(2), 2013.

Marco Beccuti, Matteo Carrara, Francesca Cordero, Fulvio Lazzarato, Susanna Donatelli, Francesca Nadalin, Alberto Policriti, and Raffaele A. Calogero. Chimera: a Bioconductor package for secondary analysis of fusion products. *Bioinformatics*, 30:3556–3557, 2014. doi: 10.1093/bioinformatics/btu662.

Jon-Matthew Belton, Rachel Patton McCord, Johan Harmen Gibcus, Natalia Naumova, Ye Zhan, and Job Dekker. Hi–c: a comprehensive technique to capture the conformation of genomes. *Methods*, 58(3):268–276, 2012.

Matteo Benelli, Chiara Pescucci, Giuseppina Marseglia, Marco Severgnini, Francesca Torricelli, and Alberto Magi. Discovering chimeric transcripts in paired-end RNA-seq data by using EricScript. *Bioinformatics*, 28(24):3232–3239, 2012.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B*, pages 289–300, 1995.

Michael F. Berger, Joshua Z. Levin, Krishna Vijayendran, Andrey Sivachenko, Xian Adiconis, Jared Maguire, Laura A. Johnson, James Robinson, Roel G. Verhaak, Carrie Sougnez, Robert C. Onofrio, Liuda Ziaugra, Kristian Cibulskis, Elisabeth Laine, Jordi Barretina, Wendy Winckler, David E. Fisher, Gad Getz, Matthew Meyerson, David B. Jaffe, Stacey B. Gabriel, Eric S. Lander, Reinhard Dummer, Andreas Gnirke, Chad Nusbaum, and Levi A. Garraway. Integrative analysis of the melanoma transcriptome. *Genome Research*, 20(4): 413–427, 2010.

Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, page btu170, 2014.

Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences*, 107(21):9546–9551, 2010.

Doruk Bozdağ, Jeffrey D Parvin, and Umit V Catalyurek. A biclustering method to discover co-regulated genes using diverse gene expression datasets. In *Bioinformatics and Computational Biology*, pages 151–163. Springer, 2009.

Doruk Bozdağ, Ashwin S Kumar, and Umit V Catalyurek. Comparative analysis of biclustering algorithms. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 265–274. ACM, 2010.

Atul J Butte and Isaac S Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Pac. Symp. Biocomput.*, volume 5, pages 418–429, 2000.

Kevin P Byrne and Kenneth H Wolfe. The yeast gene order browser: combining curated homology and syntenic context reveals gene fate in polyploid species. *Genome research*, 15(10):1456–1461, 2005.

José Caldas and Samuel Kaski. Bayesian biclustering with the plaid model. In *Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on*, pages 291–296. IEEE, 2008.

Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421, 2009.

Yonghao Cao, Brittany A. Goods, Khadir Raddassi, Gerald T. Nepom, William W. Kwok, J. Christopher Love, and David A. Hafler. Functional inflammatory profiles distinguish myelin-reactive T cells from patients with multiple sclerosis. *Science Translational Medicine*, 17(287):287ra74, 2015.

Matteo Carrara, Marco Beccuti, Fulvio Lazzarato, Federica Cavallo, Francesca Cordero, Susanna Donatelli, and Raffaele A. Calogero. State-of-the-art fusion-finder algorithms sensitivity and specificity. *BioMed Research International*, 2013:340620, 2013.

Helen C Causton, Bing Ren, Sang Seok Koh, Christopher T Harbison, Elenita Kanin, Ezra G Jennings, Tong Ihn Lee, Heather L True, Eric S Lander, and Richard A Young. Remodeling of yeast genome expression in response to environmental changes. *Mol. Biol. Cell*, 12(2): 323–337, 2001.

Lun-Ching Chang, Hui-Min Lin, Etienne Sibille, and George C Tseng. Meta-analysis methods for combining multiple expression profiles: comparisons, statistical characterization and an application guideline. *BMC bioinformatics*, 14(1):368, 2013.

Deborah Chasman, Yi-Hsuan Ho, David B Berry, Corey M Nemec, Matthew E MacGilvray, James Hose, Anna E Merrill, M Violet Lee, Jessica L Will, Joshua J Coon, et al. Pathway connectivity and signaling coordination in the yeast stress-activated signaling network. *Molecular systems biology*, 10(11):759, 2014.

Ken Chen, John W. Wallis, Cyriac Kandoth, Joelle M. Kalicki-Veizer, Karen L. Mungall, Andrew J. Mungall, Steven J. Jones, Marco A. Marra, Timothy J. Ley, Elaine R. Mardis, Richard K. Wilson, John N. Weinstein, and Li Ding. BreakFusion: targeted assembly-based identification of gene fusions in whole transcriptome paired-end sequencing data. 28 (14):1923–1924, 2012.

Ken Chen, Nicholas Navin, Yong Wang, Heather Schmidt, John Wallis, Beifang Niu, Xian Fan, Hao Zhao, Michael McLellan, Katherine Hoadley, Elaine Mardis, Timothy Ley, Charles Perou, Richard Wilson, and Li Ding. BreakTrans: uncovering the genomic architecture of gene fusions. *Genome Biology*, 14(8):R87, 2013.

Zhang-Hui Chen, P Yu Yan, Junyan Tao, Silvia Liu, George Tseng, Michael Nalesnik, Ronald Hamilton, Rohit Bhargava, Joel B Nelson, Arjun Pennathur, et al. Man2a1–fer fusion gene is expressed by human liver and other tumor types and has oncogenic activity in mice. *Gastroenterology*, 2017.

Cheng and Church. Biclustering of expression data. *Proc. ISMB AAAI Press*, pages 93–103, 2000.

J Michael Cherry, Eurie L Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T Chan, Karen R Christie, Maria C Costanzo, Selina S Dwight, Stacia R Engel, et al. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic acids research*, page gkr1029, 2011.

Murim Choi, Ute I Scholl, Weizhen Ji, Tiewen Liu, Irina R Tikhonova, Paul Zumbo, Ahmet Nayir, Aysin Bakkaloğlu, Seza Özen, Sami Sanjad, et al. Genetic diagnosis by whole exome capture and massively parallel dna sequencing. *Proceedings of the National Academy of Sciences*, 106(45):19096–19101, 2009.

Jan Cools, Daniel J DeAngelo, Jason Gotlib, Elizabeth H Stover, Robert D Legare, Jorges Cortes, Jeffrey Kutok, Jennifer Clark, Ilene Galinsky, James D Griffin, et al. A tyrosine kinase created by fusion of the pdgfra and fip1l1 genes as a therapeutic target of imatinib in idiopathic hypereosinophilic syndrome. *New England Journal of Medicine*, 348(13): 1201–1214, 2003.

Francis Crick et al. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

Francis HC Crick. On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8, 1958.

Christina Curtis, Sohrab P Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M Rueda, Mark J Dunning, Doug Speed, Andy G Lynch, Shamith Samarajiwa, Yinyin Yuan, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

Susmita Datta and Somnath Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4):459–466, 2003.

Nadia Davidson, Ian Majewski, and Oshlack Alicia. JAFFA: High sensitivity transcriptome-focused fusion gene detection. *Genome Medicine*, 7:43, 2015.

Hidde De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103, 2002.

Marcilio CP de Souto, Ivan G Costa, Daniel SA de Araujo, Teresa B Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC bioinformatics*, 9(1):497, 2008.

Christine Desmedt, Fanny Piette, Sherene Loi, Yixin Wang, Françoise Lallemand, Benjamin Haibe-Kains, Giuseppe Viale, Mauro Delorenzi, Yi Zhang, Mahasti Saghatchian d'Assignies, et al. Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series. *Clinical cancer research*, 13(11):3207–3214, 2007.

Patrik D'haeseleer. How does gene expression clustering work? *Nature biotechnology*, 23 (12):1499, 2005.

Patrik D'haeseleer, Shoudan Liang, and Roland Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

J Richard Dickinson, L Eshantha J Salgado, and Michael JE Hewlins. The catabolism of amino acids to long chain and complex alcohols in saccharomyces cerevisiae. *Journal of Biological Chemistry*, 278(10):8028–8034, 2003.

Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: ultrafast universal RNA-seq aligner. 29(1):15–21, 2013.

Henrik Edgren, Astrid Murumagi, Sara Kangaspeska, Daniel Nicorici, Vesa Hongisto, Kristine Kleivi, Inga Rye, Sandra Nyberg, Maija Wolf, Anne-Lise Borresen-Dale, and Olli Kallioniemi. Identification of fusion genes in breast cancer by paired-end RNA-sequencing. *Genome Biology*, 12(1):R6, 2011.

Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Stat. sci.*, pages 54–75, 1986.

Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

Kemal Eren, Mehmet Deveci, Onur Küçüktunç, and Ümit V Çatalyürek. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, 14(3):279–292, 2013.

Simon A. Forbes, David Beare, Prasad Gunasekaran, Kenric Leung, Nidhi Bindal, Harry Boutselakis, Minjie Ding, Sally Bamford, Charlotte Cole, Sari Ward, Chai Yin Kok, Mingming Jia, Tisham De, Jon W. Teague, Michael R. Stratton, Ultan McDermott, and Peter J. Campbell. COSMIC: exploring the world's knowledge of somatic mutations in human cancer. *Nucleic Acids Research*, 43(D1):D805–811, 2015. doi: 10.1093/nar/gku1075.

Richard W Francis, Katherine Thompson-Wicking, Kim W Carter, Denise Anderson, Ursula R Kees, and Alex H Beesley. Fusionfinder: a software tool to identify expressed gene fusion candidates from rna-seq data. *PloS one*, 7(6):e39987, 2012.

Adelaide Freitas, Wassim Ayadi, Mourad Elloumi, LJ Oliveira, and Jin-Kao Hao. Survey on biclustering of gene expression data. *Biological Knowl. Disc. Handbook*, pages 591–608, 2012.

Milana Frenkel-Morgenstern, Alessandro Gorohovski, Vincent Lacroix, Mark Rogers, Kristina Ibanez, Cesar Boullosa, Eduardo Andres Leon, Asa Ben-Hur, and Alfonso Valencia. ChiTaRS: a database of human, mouse and fruit fly chimeric transcripts and RNA-sequencing data. *Nucleic Acids Research*, 41(D1):D142–D151, 2013. doi: 10.1093/nar/gks1041.

Milana Frenkel-Morgenstern, Alessandro Gorohovski, Dunja Vucenovic, Lorena Maestre, and Alfonso Valencia. ChiTaRS 2.1: an improved database of the chimeric transcripts and RNA-seq data with novel sense-antisense chimeric RNA transcripts. *Nucleic Acids Research*, 43:D68–75, 2015. doi: 10.1093/nar/gku1199.

Audrey P Gasch, Paul T Spellman, Camilla M Kao, Orna Carmel-Harel, Michael B Eisen, Gisela Storz, David Botstein, and Patrick O Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11(12):4241–4257, 2000.

Anne-Claude Gavin, Patrick Aloy, Paola Grandi, Roland Krause, Markus Boesche, Martina Marzioch, Christina Rau, Lars Juhl Jensen, Sonja Bastuck, Birgit Dümpelfeld, et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.

H. Ge, K. Liu, T. Juan, F. Fang, M. Newman, and W. Hoeck. FusionMap: detecting fusion genes from next-generation sequencing data at base-pair resolution. *Bioinformatics*, 27 (14):1922–1928, 2011.

Thomas R. Gingeras. Implications of chimaeric non-co-linear transcripts. *Nature*, 461(7261): 206–211, 2009.

Jiajun Gu and Jun S Liu. Bayesian biclustering of gene expression data. *BMC genomics*, 9 (1):S4, 2008.

Tina Gunderson and Yen-Yi Ho. An efficient algorithm to explore liquid association on a genome-wide scale. *BMC Bioinformatics*, 15(1):371, 2014.

Felix Haglund, Ran Ma, Mikael Huss, Luqman Sulaiman, Ming Lu, Inga-Lena Nilsson, Anders Höög, C. Christofer Juhlin, Johan Hartman, and Catharina Larsson. Evidence of a functional estrogen receptor in parathyroid adenomas. *The Journal of Clinical Endocrinology & Metabolism*, 97(12):4631–4639, 2012.

John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129, 1972.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

Yen-Yi Ho, Giovanni Parmigiani, Thomas A Louis, and Leslie M Cope. Modeling liquid association. *Biometrics*, 67(1):133–141, 2011.

Sepp Hochreiter, Ulrich Bodenhofer, Martin Heusel, Andreas Mayr, Andreas Mitterecker, Adetayo Kasim, Tatsiana Khamiakova, Suzy Van Sanden, Dan Lin, Willem Talloen, et al. Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010.

Danilo Horta and Ricardo JGB Campello. Similarity measures for comparing biclusterings. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(5): 942–954, 2014.

Timothy R Hughes, Matthew J Marton, Allan R Jones, Christopher J Roberts, Roland Stoughton, Christopher D Armour, Holly A Bennett, Ernest Coffey, Hongyue Dai, Yudong D He, et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

Zhiguang Huo, Ying Ding, Silvia Liu, Steffi Oesterreich, and George Tseng. Meta-analytic framework for sparse k-means to identify disease subtypes in multiple transcriptomic studies. *Journal of the American Statistical Association*, 111(513):27–42, 2016.

Matthew K. Iyer, Arul M. Chinnaiyan, and Christopher A. Maher. ChimeraScan: a tool for identifying chimeric transcription in sequencing data. 27(20):2903–2904, 2011.

Wenlong Jia, Kunlong Qiu, Minghui He, Pengfei Song, Quan Zhou, Feng Zhou, Yuan Yu, Dandan Zhu, Michael Nickerson, Shengqing Wan, Xiangke Liao, Xiaoqian Zhu, Shaoliang Peng, Yingrui Li, Jun Wang, and Guangwu Guo. SOAPfuse: an algorithm for identifying fusion transcripts from paired-end RNA-Seq data. *Genome Biology*, 14(2):R12, 2013.

Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. Kegg as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, 44(D1): D457–D462, 2016.

Dongwan D Kang, Etienne Sibille, Naftali Kaminski, and George C Tseng. Metaqc: objective quality control and inclusion/exclusion criteria for genomic meta-analysis. *Nucleic acids research*, 40(2):e15–e15, 2012.

Sara Kangaspeska, Susanne Hultsch, Henrik Edgren, Daniel Nicorici, Astrid Murumägi, and Olli Kallioniemi. Reanalysis of RNA-Sequencing data reveals several additional fusion genes with multiple isoforms. *PLoS ONE*, 7:e48745, 2012.

Yarden Katz, Eric T. Wang, Edoardo M. Airoldi, and Christopher B. Burge. Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nat Meth*, 7(12): 1009–1015, 2010.

Frederic J. Kaye. Mutation-associated fusion cancer genes in solid tumors. *Mol Cancer Ther*, 8(6):1399–1408, 2009.

W. James Kent. BLAT -the BLAST-like alignment tool. *Genome Research*, 12(4):656–664, 2012.

Daehwan Kim and Steven Salzberg. TopHat-Fusion: an algorithm for discovery of novel fusion transcripts. *Genome Biology*, 12(8):R72, 2011.

Daehwan Kim, Geo Pertea, Cole Trapnell, Harold Pimentel, Ryan Kelley, and Steven L Salzberg. Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):R36, 2013.

Marcus Kinsella, Olivier Harismendy, Masakazu Nakano, Kelly A. Frazer, and Vineet Bafna. Sensitive gene fusion detection using ambiguously mapping RNA-Seq read pairs. 27(8): 1068–1075, 2011.

Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

Theo A Knijnenburg, Jean-Marc G Daran, Marcel A van den Broek, Pascale AS Daran-Lapujade, Johannes H de Winde, Jack T Pronk, Marcel JT Reinders, and Lodewyk FA Wessels. Combinatorial effects of environmental parameters on transcriptional regulation in saccharomyces cerevisiae: a quantitative analysis of a compendium of chemostat-based transcriptome data. *BMC genomics*, 10(1):1, 2009.

Felix Krueger, Benjamin Kreck, Andre Franke, and Simon R Andrews. DNA methylome analysis using short bisulfite sequencing data. *Nature methods*, 9(2):145–151, 2012.

Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nat Meth*, 9(4):357–359, 2012.

Ben Langmead, Cole Trapnell, Mihai Pop, and Steven Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3): R25, 2009.

Laura Lazzeroni and Art Owen. Plaid models for gene expression data. *Statistica sinica*, pages 61–86, 2002.

Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. 25(14):1754–1760, 2009.

Heng Li and Nils Homer. A survey of sequence alignment algorithms for next-generation sequencing. 11(5):473–483, 2010.

Jia Li, George C Tseng, et al. An adaptively weighted statistic for detecting differential gene expression when combining multiple transcriptomic studies. *The Annals of Applied Statistics*, 5(2A):994–1019, 2011a.

Ker-Chau Li. Genome-wide coexpression dynamics: theory and application. *Proc. Natl. Acad. Sci. USA*, 99(26):16875–16880, 2002.

Ker-Chau Li, Ching-Ti Liu, Wei Sun, Shinsheng Yuan, and Tianwei Yu. A system for enhancing genome-wide coexpression dynamics study. *Proc. Natl. Acad. Sci. USA*, 101 (44):15561–15566, 2004.

Ker-Chau Li, Aarno Palotie, Shinsheng Yuan, Denis Bronnikov, Daniel Chen, Xuelian Wei, Oi-Wa Choi, Janna Saarela, and Leena Peltonen. Finding disease candidate genes by liquid association. *Genome biology*, 8(10):R205, 2007.

Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. SOAP2: an improved ultrafast tool for short read alignment. 25(15):1966–1967, 2009.

Yang Li, Jeremy Chien, David I. Smith, and Jian Ma. FusionHunter: identifying fusion transcripts in cancer using paired-end RNA-seq. 27(12):1708–1710, 2011b.

Alan Wee-Chung Liew, Ngai-Fong Law, and Hong Yan. Missing value imputation for gene expression data: computational techniques to recover missing data from available information. *Briefings in bioinformatics*, 12(5):498–513, 2011.

Chenglin Liu, Jinwen Ma, ChungChe Chang, and Xiaobo Zhou. FusionQ: a novel approach for gene fusion detection and quantification from paired-end RNA-Seq. *BMC Bioinformatics*, 14(1):193, 2013.

Silvia Liu, Wei-Hsiang Tsai, Ying Ding, Rui Chen, Zhou Fang, Zhiguang Huo, SungHwan Kim, Tianzhou Ma, Ting-Yu Chang, Nolan Michael Priedigkeit, et al. Comprehensive evaluation of fusion transcript detection algorithms and a meta-caller to combine top performing methods in paired-end rna-seq data. *Nucleic acids research*, 44(5):e47–e47, 2016.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

Jian-Hua Luo, Silvia Liu, Ze-Hua Zuo, Rui Chen, George C Tseng, and P Yu Yan. Discovery and classification of fusion transcripts in prostate cancer and normal prostate tissue. *The American journal of pathology*, 185(7):1834–1845, 2015.

Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

Christopher A. Maher, Nallasivam Palanisamy, John C. Brenner, Xuhong Cao, Shanker Kalyana-Sundaram, Shujun Luo, Irina Khrebtukova, Terrence R. Barrette, Catherine Grasso, Jindan Yu, Robert J. Lonigro, Gary Schroth, Chandan Kumar-Sinha, and Arul M. Chinnaiyan. Chimeric transcript discovery by paired-end transcriptome sequencing. *Proceedings of the National Academy of Sciences*, 106(30):12353–12358, 2009. doi: 10.1073/pnas.0904720106.

Ranjan Maitra and Ivan P Ramler. Clustering in the presence of scatter. *Biometrics*, 65(2): 341–352, 2009.

Nathan Mantel. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer chemotherapy reports. Part 1*, 50(3):163–170, 1966.

Martella and Vermunt. Model-based approaches to synthesize microarray data: a unifying review using mixture of sems. *Statistical methods in medical research*, 22:567–582, 2013.

Francesca Martella, Marco Alfo, and Maurizio Vichi. Biclustering of gene expression data by an extension of mixtures of factor analyzers. *International Journal of Biostatistics*, 4 (1):1078–1078, 2008.

Lee McAlister and Michael J Holland. Targeted deletion of a yeast enolase structural gene. identification and isolation of yeast enolase isozymes. *Journal of Biological Chemistry*, 257 (12):7181–7188, 1982.

Geoffrey J. McLachlan, RW Bean, and David Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

Andrew McPherson, Fereydoun Hormozdiari, Abdalnasser Zayed, Ryan Giuliany, Gavin Ha, Mark G. F. Sun, Malachi Griffith, Alireza Heravi Moussavi, Janine Senz, Nataliya Melnyk, Marina Pacheco, Marco A. Marra, Martin Hirst, Torsten O. Nielsen, S. Cenk Sahinalp, David Huntsman, and Sohrab P. Shah. deFuse: An algorithm for gene fusion discovery in tumor RNA-Seq data. *PLoS Comput Biol*, 7(5):e1001138, 2011a.

Andrew McPherson, Chunxiao Wu, Iman Hajirasouliha, Fereydoun Hormozdiari, Faraz Hach, Anna Lapuk, Stanislav Volik, Sohrab Shah, Colin Collins, and S Cenk Sahinalp. Comrad: detection of expressed rearrangements by integrated analysis of rna-seq and low coverage genome sequence data. *Bioinformatics*, 27(11):1481–1488, 2011b.

Mario Medvedovic, Ka Yee Yeung, and Roger Eugene Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232, 2004.

Ryan E. Mills, Klaudia Walter, Chip Stewart, Robert E. Handsaker, Ken Chen, Can Alkan, Alexej Abyzov, Seungtai Chris Yoon, Kai Ye, R. Keira Cheetham, Asif Chinwalla, Donald F. Conrad, Yutao Fu, Fabian Grubert, Iman Hajirasouliha, Fereydoun Hormozdiari, Lilia M. Iakoucheva, Zamin Iqbal, Shuli Kang, Jeffrey M. Kidd, Miriam K. Konkel, Joshua Korn, Ekta Khurana, Deniz Kural, Hugo Y. K. Lam, Jing Leng, Ruiqiang Li, Yingrui Li, Chang-Yun Lin, Ruibang Luo, Xinmeng Jasmine Mu, James Nemesh, Heather E. Peckham, Tobias Rausch, Aylwyn Scally, Xinghua Shi, Michael P. Stromberg, Adrian M. Stutz, Alexander Eckehart Urban, Jerilyn A. Walker, Jiantao Wu, Yujun Zhang, Zhengdong D. Zhang, Mark A. Batzer, Li Ding, Gabor T. Marth, Gil McVean, Jonathan Sebat, Michael Snyder, Jun Wang, Kenny Ye, Evan E. Eichler, Mark B. Gerstein, Matthew E. Hurles, Charles Lee, Steven A. McCarroll, and Jan O. Korbel. Mapping copy number variation by population-scale genome sequencing. *Nature*, 470(7332):59–65, 2011.

F Mitelman, B Johansson, and F Mertens. Mitelman database of chromosome aberrations and gene fusions in cancer, 2015. URL http://cgap.nci.nih.gov/Chromosomes/Mitelman.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1):91–118, 2003.

Kohbalan Moorthy, Mohd Saberi Mohamad, and Safaai Deris. A review on missing value imputation algorithms for microarray gene expression data. *Current Bioinformatics*, 9(1): 18–22, 2014.

TM Murali and Simon Kasif. Extracting conserved gene expression motifs from gene expression data. In *Pacific symposium on biocomputing*, volume 8, pages 77–88, 2003.

Cancer Genome Atlas Network et al. Comprehensive molecular portraits of human breast tumors. *Nature*, 490(7418):61, 2012.

Pauline C Ng and Ewen F Kirkness. Whole genome sequencing. In *Genetic variation*, pages 215–226. Springer, 2010.

Daniel Nicorici, Mihaela Satalan, Henrik Edgren, Sara Kangaspeska, Astrid Murumagi, Olli Kallioniemi, Sami Virtanen, and Olavi Kilkku. FusionCatcher - a tool for finding somatic fusion genes in paired-end RNA-sequencing data. page 011650, 2014.

Francisco Novo, Inigo de Mendibil, and Jose Vizmanos. TICdb: a collection of gene-mapped translocation breakpoints in cancer. *BMC Genomics*, 8(1):33, 2007.

Peter J Park. Chip–seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10):669–680, 2009.

Joel S Parker, Michael Mullins, Maggie CU Cheang, Samuel Leung, David Voduc, Tammi Vickery, Sherri Davies, Christiane Fauron, Xiaping He, Zhiyuan Hu, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, 27(8): 1160–1167, 2009.

Rocco Piazza, Alessandra Pirola, Roberta Spinelli, Simona Valletta, Sara Redaelli, Vera Magistroni, and Carlo Gambacorti-Passerini. Fusionanalyser: a new graphical, event-driven tool for fusion rearrangements discovery. *Nucleic acids research*, page gks394, 2012.

Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

John Quackenbush. Microarray data normalization and transformation. *Nature genetics*, 32: 496–501, 2002.

Jörg Reinders, Karina Wagner, Rene P Zahedi, Diana Stojanovski, Beate Eyrich, Martin Van der Laan, Peter Rehling, Albert Sickmann, Nikolaus Pfanner, and Chris Meisinger. Profiling phosphoproteins of yeast mitochondria reveals a role of phosphorylation in assembly of the atp synthase. *Molecular & Cellular Proteomics*, 6(11):1896–1906, 2007.

Sylvia Richardson, George C Tseng, and Wei Sun. Statistical methods in integrative genomics. *Annual Review of Statistics and Its Application*, 3:181–209, 2016.

Gordon Robertson, Jacqueline Schein, Readman Chiu, Richard Corbett, Matthew Field, Shaun D Jackman, Karen Mungall, Sam Lee, Hisanaga Mark Okada, Jenny Q Qian, et al. De novo assembly and analysis of rna-seq data. *Nature methods*, 7(11):909–912, 2010.

Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1): 139–140, 2010.

Matthew Ruffalo, Thomas LaFramboise, and Mehmet Koyutürk. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics*, 27(20):2790–2796, 2011.

Onur Sakarya, Heinz Breu, Milan Radovich, Yongzhi Chen, Yulei N. Wang, Catalin Barbacioru, Sowmi Utiramerur, Penn P. Whitley, Joel P. Brockman, Paolo Vatta, Zheng Zhang, Liviu Popescu, Matthew W. Muller, Vidya Kudlingar, Nriti Garg, Chieh-Yuan Li, Benjamin S. Kong, John P. Bodeau, Robert C. Nutter, Jian Gu, Kelli S. Bramlett, Jeffrey K. Ichikawa, Fiona C. Hyland, and Asim S. Siddiqui. RNA-Seq mapping and detection of gene fusions with a suffix array algorithm. *PLoS Comput Biol*, 8(4):e1002464, 2012.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

Andrea Sboner, Lukas Habegger, Dorothee Pflueger, Stephane Terry, David Z Chen, Joel S Rozowsky, Ashutosh K Tewari, Naoki Kitabayashi, Benjamin J Moss, Mark S Chee, et al. Fusionseq: a modular framework for finding gene fusions by analyzing paired-end rna-sequencing data. *Genome biology*, 11(10):R104, 2010.

Marcus Schmidt, Daniel Böhm, Christian von Törne, Eric Steiner, Alexander Puhl, Henryk Pilch, Hans-Anton Lehr, Jan G Hengstler, Heinz Kölbl, and Mathias Gehrmann. The humoral immune system has a key prognostic impact in node-negative breast cancer. *Cancer research*, 68(13):5405–5413, 2008.

Kui Shen and George C Tseng. Meta-analysis for pathway enrichment analysis when combining multiple genomic studies. *Bioinformatics*, 26(10):1316–1323, 2010.

Manabu Soda, Young Lim Choi, Munehiro Enomoto, Shuji Takada, Yoshihiro Yamashita, Shunpei Ishikawa, Shin-ichiro Fujiwara, Hideki Watanabe, Kentaro Kurashina, Hisashi Hatanaka, Masashi Bando, Shoji Ohno, Yuichi Ishikawa, Hiroyuki Aburatani, Toshiro Niki, Yasunori Sohara, Yukihiko Sugiyama, and Hiroyuki Mano. Identification of the transforming EML4-ALK fusion gene in non-small-cell lung cancer. *Nature*, 448(7153): 561–566, 2007.

Lin Song, Peter Langfelder, and Steve Horvath. Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC Bioinformatics*, 13(1): 328, 2012.

Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S Lander, and Todd R Golub. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912, 1999.

Miguel Cacho Teixeira, Pedro Tiago Monteiro, Joana Fernandes Guerreiro, Joana Pinho Gonçalves, Nuno Pereira Mira, Sandra Costa dos Santos, Tânia Rodrigues Cabrito, Margarida Palma, Catarina Costa, Alexandre Paulo Francisco, et al. The yeastract database: an upgraded information system for the analysis of gene and genomic transcription regulation in saccharomyces cerevisiae. *Nucleic acids research*, page gkt1015, 2013.

Waibhav D Tembe, Stephanie JK Pond, Christophe Legendre, Han-Yu Chuang, Winnie S Liang, Nancy E Kim, Valerie Montel, Shukmei Wong, Timothy K McDaniel, David W Craig, and John D Carpten. Open-access synthetic spike-in mRNA-seq data for cancer gene fusions. *BMC Genomics*, 15:824, 2014.

Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng, and George C Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 2006.

Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

Scott A. Tomlins, Daniel R. Rhodes, Sven Perner, Saravana M. Dhanasekaran, Rohit Mehra, Xiao-Wei Sun, Sooryanarayana Varambally, Xuhong Cao, Joelle Tchinda, Rainer Kuefer, Charles Lee, James E. Montie, Rajal B. Shah, Kenneth J. Pienta, Mark A. Rubin, and Arul M. Chinnaiyan. Recurrent fusion of TMPRSS2 and ETS transcription factor genes in prostate cancer. *Science*, 310(5748):644–648, 2005. doi: 10.1126/science.1117679.

Wandaliz Torres-García, Siyuan Zheng, Andrey Sivachenko, Rahulsimham Vegesna, Qianghu Wang, Rong Yao, Michael F Berger, John N Weinstein, Gad Getz, and Roel GW Verhaak. PRADA: Pipeline for RNA sequencing data analysis. *Bioinformatics*, 30(15):2224–2226, 2014.

Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J Van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

George C Tseng. Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, 23(17):2247–2255, 2007.

George C Tseng and Wing H Wong. Tight clustering: A resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, 61(1):10–16, 2005.

George C Tseng, Debashis Ghosh, and Eleanor Feingold. Comprehensive literature review and statistical considerations for microarray meta-analysis. *Nucleic acids research*, page gkr1265, 2012.

Heather Turner, Trevor Bailey, and Wojtek Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational statistics & data analysis*, 48(2):235–254, 2005a.

Heather L Turner, Trevor C Bailey, Wojtek J Krzanowski, and Cheryl A Hemingway. Biclustering models for structured microarray data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2(4):316–329, 2005b.

Graham JG Upton. Fisher's exact test. *J. Roy. Stat. Soc. A Sta.*, pages 395–402, 1992.

Mark A Urich, Joseph R Nery, Ryan Lister, Robert J Schmitz, and Joseph R Ecker. Methylc-seq library preparation for base-resolution whole-genome bisulfite sequencing. *Nature protocols*, 10(3):475–483, 2015.

Maarten van Iterson, Judith M Boer, and Renée X Menezes. Filtering, fdr and power. *BMC bioinformatics*, 11(1):450, 2010.

Leif Väremo, Camilla Scheele, Christa Broholm, Adil Mardinoglu, Caroline Kampf, Anna Asplund, Intawat Nookaew, Mathias Uhlén, Bente Klarlund Pedersen, and Jens Nielsen. Proteome- and transcriptome-driven reconstruction of the human myocyte metabolic network and its use for identification of markers for diabetes. *Cell Reports*, 11(6):921–933, 2015.

Kai Wang, Darshan Singh, Zheng Zeng, Stephen J. Coleman, Yan Huang, Gleb L. Savich, Xiaping He, Piotr Mieczkowski, Sara A. Grimm, Charles M. Perou, James N. MacLeod, Derek Y. Chiang, Jan F. Prins, and Jinze Liu. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. 38(18):e178, 2010.

Liguo Wang, Shengqin Wang, and Wei Li. Rseqc: quality control of rna-seq experiments. *Bioinformatics*, 28(16):2184–2185, 2012a.

Lin Wang, Silvia Liu, Ying Ding, SS Yuan, Yen-Yi Ho, and George C Tseng. Meta-analytic framework for liquid association. *Bioinformatics (Oxford, England)*, 2017.

Qingguo Wang, Junfeng Xia, Peilin Jia, William Pao, and Zhongming Zhao. Application of next generation sequencing to human gene fusion detection: computational tools, features and perspectives. *Briefings in Bioinformatics*, 14(4):506–519, 2013. doi: 10.1093/bib/bbs044.

Xingbin Wang, Yan Lin, Chi Song, Etienne Sibille, and George C Tseng. Detecting disease-associated genes with confounding variable adjustment and the impact on genomic meta-analysis: with application to major depressive disorder. *BMC bioinformatics*, 13(1):52, 2012b.

Yixin Wang, Jan GM Klijn, Yi Zhang, Anieta M Sieuwerts, Maxime P Look, Fei Yang, Dmitri Talantov, Mieke Timmermans, Marion E Meijer-van Gelder, Jack Yu, et al. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, 365(9460):671–679, 2005.

Cecily J Wolfe, Isaac S Kohane, and Atul J Butte. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. *BMC Bioinformatics*, 6(1):1, 2005.

Jikun Wu, Wenqian Zhang, Songbo Huang, Zengquan He, Yanbing Cheng, Jun Wang, Tak-Wah Lam, Zhiyu Peng, and Siu-Ming Yiu. Soapfusion: a robust and effective computational fusion discovery tool for rna-seq reads. *Bioinformatics*, page btt522, 2013.

Yi-Fan Xu, Xin Zhao, David S Glass, Farnaz Absalan, David H Perlman, James R Broach, and Joshua D Rabinowitz. Regulation of yeast pyruvate kinase by ultrasensitive allostery independent of phosphorylation. *Molecular cell*, 48(1):52–62, 2012.

Mingjin Yan and Keying Ye. Determining the number of clusters using the weighted gap statistic. *Biometrics*, 63(4):1031–1037, 2007.

P Yu Yan, Silvia Liu, Zhiguang Huo, Amantha Martin, Joel B Nelson, George C Tseng, and Jian-Hua Luo. Genomic copy number variations in the genomes of leukocytes predict prostate cancer clinical outcomes. *PloS one*, 10(8):e0135982, 2015.

Yan P. Yu, Ying Ding, Zhanghui Chen, Silvia Liu, Amantha Michalopoulos, Rui Chen, Zulfiqar G. Gulzar, Bing Yang, Kathleen M. Cieply, Alyssa Luvison, Bao-Guo Ren, James D. Brooks, David Jarrard, Joel B. Nelson, George K. Michalopoulos, George C. Tseng, and Jian-Hua Luo. Novel fusion transcripts associate with progressive prostate cancer. *The American Journal of Pathology*, 184(10):2840 – 2849, 2014.

Daniel R. Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18:821–829, 2008.

Bin Zhang, Steve Horvath, et al. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1128, 2005.

Jian Zhang. A bayesian model for biclustering with applications. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 59(4):635–656, 2010.

Jiexin Zhang, Yuan Ji, and Li Zhang. Extracting three-way gene interactions from microarray data. *Bioinformatics*, 23(21):2903–2909, 2007.

Li Qin Zhang, Dilyara Cheranova, Margaret Gibson, Shinghua Ding, Daniel P. Heruth, Deyu Fang, and Shui Qing Ye. RNA-seq reveals novel transcriptome of genes and their isoforms in human pulmonary microvascular endothelial cells treated with thrombin. *PLoS ONE*, 7(2):e31229, 2012.