

STATISTICAL METHODS FOR GENE SET ANALYSIS

by

Zhishi Wang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2015

Date of final oral examination: 08/28/2015

The dissertation is approved by the following members of the Final Oral Committee:

Michael A. Newton, Professor, Statistics

C. Kendzierski Newton, Professor, Biostatistics & Medical Informatics

Chunming Zhang, Professor, Statistics

Kam-Wah Tsui, Professor, Statistics

Sijian Wang, Professor, Statistics

© Copyright by Zhishi Wang 2015

All Rights Reserved

To my family

Acknowledgments

I would like to express my deepest appreciation to my thesis advisor, Professor Michael A. Newton, for his invaluable guidance and persistent help in my study. I feel very fortunate to be a PhD student of Michael, only under whom I could receive that many opportunities to work with experts from interdisciplinary fields to broaden my knowledge. Michael is also a good advisor beyond research as he convincingly delivers a spirit of adventure for being a successful man. He has always been a role model and an instrumental piece to my path in becoming a successful statistician in the future. I am also grateful to my funding source during my PhD study, Morgridge Institute for Research at UW-Madison.

I want to thank Professor Kam-Wah Tsui, Professor Christina Kendzioriski, Professor Chunming Zhang and Professor Sijian Wang for kindly serving as my committee members. They provided me with insightful comments and suggestions for my dissertation. They generously shared their ideas and advice, and encouraged me to complete my thesis research.

I am indebted to all my dear peer students and friends, without whose help and support I would not have been here today. Last but not least I want to thank my

parents and my family for their unconditional love and belief in me. I would like to express my greatest gratitude by dedicating this thesis to them.

Contents

Contents iv

List of Tables vi

List of Figures x

Abstract xv

1 Multiset Statistics for Gene Set Analysis 1

 1.1 *Forward* 1

 1.2 *Introduction to gene set analysis* 2

 1.3 *Gene set repositories* 6

 1.4 *Uni-set statistics* 10

 1.5 *Multi-set statistics* 14

 1.6 *Summary* 25

2 A Multi-Functional Analyzer using Parameter Constraints 28

 2.1 *Forward* 28

2.2	<i>Role model</i>	30
2.3	<i>Maximum a posterior (MAP) estimate via integer linear programming (ILP)</i>	41
2.4	<i>Posterior sampling via Monte Carlo Markov Chain</i>	46
2.5	<i>Simulations</i>	48
2.6	<i>Case studies</i>	53
2.7	<i>Estimating false-positive and true-positive rates</i>	66
3	Implementations and Extensions of Multi-Functional Analyzer	69
3.1	<i>Forward</i>	69
3.2	<i>Rolemodel: an R package to implement MFA analysis</i>	70
3.3	<i>Model diagnostics and other issues</i>	85
3.4	<i>A meta-analysis of four genome-wide studies of influenza virus</i>	90
3.5	<i>The application of MFA to the peptide-protein inference problem</i>	95
3.6	<i>An approximate algorithm for MFA-ILP</i>	100

List of Tables

1.1	NPC example: 44 GO terms are inferred to be active by the MFA MAP estimate, and ordered by marginal posterior activation probability. Basic statistics on these terms are provided (# NPC-associated genes/set size). The right-most columns give the MCMC-computed marginal posterior activation probabilities for these terms, using MGSA and MFA.	24
1.2	NPC example: 25 Gene Ontology (GO) terms inferred to be active by model-based gene set analysis (MGSA).	26
2.1	Simulation comparison of three gene-set analysis methods, a case of low overlap among gene sets: tabulated are mean values from 100 simulated data sets. On average 7.3 truly activated sets occur.	50
2.2	Simulation comparison of three gene-set analysis methods, a case of higher overlap among gene sets: tabulated are mean values from 100 simulated data sets. On average there are 10.1 truly activated sets in this case. . .	50

2.3	MFA results in Type-2 Diabetes (T2D) example: 11 GO terms are inferred to be active using the ILP algorithm to compute the MAP estimate (rows). Basic statistics on these terms are provided (# T2D-associated genes/set size). The next two columns give the MCMC-computed marginal posterior activation probabilities for these terms, both using MGSA and MFA, the constrained alternative. The final column holds the Benjamini-Hochberg adjusted Fisher-test p-value. All calculations start with 6037 GO terms (those annotating between 5 and 50 human genes) that together annotate 10626 human genes. Of the 77 total T2D genes, 58 have at least one annotation to these 6037 GO terms. The inferred gene sets cover 26 of these 58 T2D genes.	54
2.4	T2D: MFA-ILP results in Type-2 Diabetes example. Essentially the same as Table 2.3, but with GO ID's included; coverage 26.	56
2.5	T2D: MGSA results in Type-2 Diabetes example. Similar to Table 2.3, but the five gene sets shown are those inferred to be activated according to MGSA ($P(\text{MGSA}) \geq 0.5$); coverage 13.	56
2.6	T2D: Like Tables 2.3 and 2.4, but selected for small adjusted Fisher p-value; coverage 22.	57
2.7	T2D: GO terms with MFA marginal posterior activation probability exceeding 0.5. All of these sets are in the MAP estimate (Table 2.3). . . .	58
2.8	Influenza RNAi; GO terms found by MFA-ILP; coverage 245.	61
2.9	Influenza RNAi; GO terms found by MGSA; coverage 226.	62
2.10	Influenza RNAi; GO terms by Fisher's method; coverage 97.	63

2.11	Influenza RNAi; GO terms with MFA marginal posterior activation probability exceeding 0.5 . Three sets (GO:0046330, GO:0005719, GO:0051233) are not in the MAP estimate (Table 2.8).	64
2.12	Coverage, miscoverage, and other statistics for three methods in two case studies. Recall y_g indicates whether or not gene g is on the observed gene list; A_g is whether or not the gene is in a set that is inferred to be active. Entries (except last column) are numbers of genes satisfying the condition. In each case and gene-set system, the first two numerical columns add to a constant, as do the next two columns. The final column indicates the number of sets inferred to be active, after trimming. MFA-ILP delivers relatively high coverage and low mis-coverage in all cases.	65

- 3.1 Version effects: Tabulated are similarity scores comparing set structure and inferred active sets over time-adjacent versions of GO[5:50]. The collections of sets annotating between 5 and 50 human genes fluctuate over recent versions of GO. Respectively, in the four most recent fall versions of **Bioconductor**, the collections contain 4830, 5546, 6037 and 6488 gene sets. The first row shows the Jaccard index (size of intersection over size of union) comparing subsequent versions of these gene-set collections. In addition to the collections changing, the annotations recording which genes are in which sets also change over time. The second row measures similarity of the sets of annotations. Subsequent rows show similarity of reported lists of gene sets in the two main examples. In this comparison, the set is reported if it is in the MAP estimate by MFA-ILP and if its marginal posterior probability exceeds a threshold (50% or 80%). Inferred active sets depend to some extent on the GO version in view; setting stronger marginal posterior thresholds reduces the false-discovery rate and reduces the version effect. 89
- 3.2 A series of parameters are implemented in the role model calculations. Coverage is the number of identified peptides explained by the identified proteins, and mis-coverage is the number of non-identified peptides contained by the identified proteins. There are totally 3,433 identified peptides. 97

List of Figures

- 1.1 543 Gene Ontology (GO) terms (*red*) representing all molecular functions (MFs) holding between 5 and 10 human genes and the associated 2,613 genes (*green*), presented as a bipartite graph. The edges (*blue*) connect genes to the sets that contain them. This graph has 134 components , including one large component with over 1,700 nodes. (Based on data in `org.Hs.eg.db` version 2.8.0.) 8
- 1.2 One component from Figure 1.1 involving 3 Gene Ontology (GO) terms and 17 genes (*left*). The right panel shows one realization of latent activity states, with an activated term (*black*) inducing activity on all the contained genes (*gray*). Both model-based gene-set analysis (MGSA) and multifunctional analysis (MFA) use the set-to-gene activity mapping expressed in the right panel. MFA constrains latent activities to guarantee a unique gene-to-set mapping; obtaining such a mapping becomes an issue in more complex domains of the graph. Gene-level data are Bernoulli trials having a low rate (*white*) or a higher rate (*gray*). 18

2.1	Expected number of sets that violate the activation hypothesis (AH) for three recent versions of Gene Ontology (GO), considering sets holding between 5 and 20 genes, taken on the <i>i.i.d.</i> Bernoulli prior. Calculations are done at $\pi = 1/100$. Respectively, these systems contain 3591, 4096, 4449 and 4772 gene sets, and correspond to versions of <code>org.Hs.eg.db</code>	40
2.2	Operating characteristics of MFA-MCMC, MGSA, and Fisher's test based on simulating the role model in the <i>D. melanogaster</i> genome.	52
2.3	GO terms inferred by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) to be activated to explain the type-2 diabetes associated genes (58 of the 77 are annotated to the 6037 GO terms used). Panels are linked by the location of sets; three sets (green) were identified by multiple methods. GO ID is on the outer rim, and sets are ordered by size as in Tables 2.3, 2.4, 2.5 and 2.6, from largest (noon) and decreasing clockwise; numbers inside circles indicate the number of T2D genes annotated to that set; a line connects two sets if their intersection contains at least one T2D-associated gene. The total number of T2D genes explained by the identified sets (coverage), lower right, as a subset of the 58 starting genes. Similarly the mis-coverage (mcov) counts the number of genes within the inferred active sets that are not on the observed gene list. GO:0046323 (glucose import) and its relations are highlighted (red) in panel A.	55

2.4	Sets (GO terms) identified by three methods (A, Fisher's test; B, MGSA; C, MFA) as activated to explain the influenza-involved human genes. Layout as in Figure 2.3. GO:0032434 and its relations are highlighted (red) in panel A.	60
3.1	Display the active gene sets. Top row is the gene set having the highest overlap with the input gene list; then this gene set and genes in this set are removed from the system. The second row is the gene set having the highest overlap with the remainder, and so on.	84
3.2	KEGG pathways identified by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) as activated to explain the type 2 diabetes associated genes. Layout as in Figures 2.3 and 2.4. There is very close agreement between MGSA and MFA-ILP.	86
3.3	KEGG pathways identified by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) as activated to explain the influenza RNAi genes. Layout as in Figures 2.3 and 2.4. There is very close agreement between MGSA and MFA-ILP.	87

- 3.4 Studies agree substantially at the level of functional categories. When each study's confirmed gene list is processed for enriched functional categories from GO (Gene Ontology), substantially more categories are identified in multiple studies than one would expect by chance, suggesting a positive association in functional signals underlying the study-specific gene lists. Shown is the distribution of an agreement statistic (mean overlap fraction) generated by gene-permutation, where the multi-functional analyzer (MFA) approach is applied to every gene list to produce a list of over-represented GO terms. The actual agreement between studies (red) is strikingly higher than expected by chance (blue) and also much higher than the 6.7% gene-list agreement (purple). 93
- 3.5 Dominant Gene Ontology terms identified in analysis of combined gene list. The multi-functional analyzer (MFA) applied to the combined list of 614 influenza-involved genes produced 91 GO terms in the maximum a posteriori (MAP) estimate of activated terms. The dominant terms are shown here, with color bars to indicate the study responsible for identifying the gene. The MFA-identified category having highest overlap with the 614-gene list is GO:0022627 (top row), the overlap being 18 genes (horizontal axis). These genes and this category are removed from the system, and then the category having the highest overlap with the remainder is identified (GO:0005852), and so on until the overlap is less than four genes. Shading in the sub-diagonal indicates genes also annotated to previously named categories. 94

- 3.6 338 proteins (red) holding between 10 and 15 peptides found only by role model, and the associated 3649 peptides (green or black; black are identified peptides), presented as a bipartite graph. The edges (blue) connect peptides to their containing proteins. 98
- 3.7 444 proteins (red) holding between 10 and 20 peptides found only by role model, and the associated 5068 peptides (green or black; black are identified peptides), presented as a bipartite graph. The edges (blue) connect peptides to their containing proteins. 99

Abstract

Gene set analysis has become the first choice for gaining insight into the underlying biology of differentially expressed genes and proteins, as it reduces complexity and has increased explanatory power. This dissertation mainly focuses on three aspects: 1) reviewing the background of gene set analysis, 2) developing a new model-based gene set analysis method, called multi-functional analyzer (MFA), and 3) discussing the implementations and extensions of multi-functional analyzer.

Chapter 1 surveys a wide variety of statistical methodologies and computational tools developed for gene set analysis. I divide them into two distinct categories: uni-set analysis and multi-set analysis, according to the number of gene sets considered at a time. I also investigate the limitations that are specific to each methodology, and how they are addressed by successive methods. The hypotheses on which different methods are built are carefully examined, which most of the review articles about gene set analysis often ignore.

Motivated by the role modeling (Bauer *et al.*, 2010), Chapter 2 investigates a novel approach by considering the parameter constraints to overcome the oversimplification of the original model. The new model also addresses the overlap and redundancy

Under the guidance of Professor Michael A. Newton.

issue between gene sets and it is proved to be more statistically efficient through either simulations or real data examples. It has two computational advances for posterior inference within the constrained parameter space: one using integer linear programming (ILP) for optimization and the other using a penalized Monte Carlo Markov Chain (MCMC) by Metropolis-Hastings algorithm. The complexity of functional records causes a computational challenge on the posterior inference, and then these two contributions help to advance the state of the art.

Chapter 3 presents future elaborations of the multi-functional analyzer described in Chapter 2. The first part is a vignette describing the R computing package `Rolemodel`. Other parts are a series of experiments and extensions that further examine the use of *role model* statistics, which include model diagnostics, approximation and application of MFA to peptide-protein inference problem.

Chapter 1

Multiset Statistics for Gene Set Analysis

1.1 Forward

Gene set analysis makes use of larger and more biologically inclusive entities than individual genes, and can be statistically more powerful than traditional gene-level analysis as evidence is accumulated from many genes. Gene sets represent information that is exogenous to experiment data, from the perspective of a scientist who generates gene-level measurement. As exogenous information, the accuracy and completeness of gene set sources are of great importance to the gene set analysis. This chapter introduces several commonly used gene set data sources. It also discusses one key problem in the gene set analysis: what is the null hypothesis in use and how to calibrate it. The current gene set analysis methods are classified into two

major groups: univariate (uni-set) analysis and model-based multivariate (multi-set) analysis. I introduce some representative methods in each category, and discuss their advantages and limitations. In particular, I introduce three novel model-based multi-set approaches proposed recently to overcome the redundancy in the reported gene sets: GenGO, MGSA and MFA. An example from cancer genomics is given to compare several different methods. The main elements of this chapter appeared in Newton and Wang (2015).

1.2 Introduction to gene set analysis

So complex are biological systems that even our data-rich research environment fails at many predictions, such as on the fate of a tissue (e.g., will it become cancerous?) or on the consequence of an intervention (e.g., what is a drug's effect on viral infection?). At the disposal of biomedical researchers who address such challenges are assays that yield genome-wide gene-level measurements, such as on the expression of each gene's RNA or protein products, on structural properties of the chromatin harboring each gene, on other factors affecting the regulation of each gene, on associations that variants of each gene have with various disease states, and so on. After great expense and effort the researcher has obtained her genome-wide gene-level data, D say, and is challenged to make sense of it through various forms of data analysis. One central analytical challenge is how to relate the local, *endogenous* D to all the other *exogenous* knowledge, K say, that has so far been compiled on the same genes. Great efforts are underway to encode this exogenous knowledge in ways that facilitate data analysis,

and the use of the resulting knowledge resources is becoming an essential component of biological research. One encoding of K is via collections of *gene sets*, each one of which is an unordered set of genes identified by some other evidence as having some specific biological property. Gene set analysis refers to a host of strategies and procedures for integrating the observed D with available gene set information in the pursuit of further knowledge. It addresses several important challenges in genomic data analysis:

1. Gene sets enable data reduction in order to organize, simplify, and explain a high-dimensional signal. This is especially useful when gene-level signals are relatively strong (e.g., many genes differentially expressed between two cellular states) and when a concise description of their functional content can be derived from the gene sets.
2. Gene sets improve sensitivity compared to gene-level analysis in cases where there are consistent but weak signals among genes in the same set.
3. Gene sets structure gene-level data in a way that may improve the prediction of other phenotypes (e.g., regression, biomarker development)

The content of a gene set analysis depends on the structures of data D and knowledge K , the bioinformatic and statistical tools being applied, and the context of the driving problem. In Section 2 we review some of the major efforts at encoding K in publicly available data resources. Further, a large number of bioinformatic and statistical tools are available for integrating D with K . As a reference point we recognize useful reviews of gene set methodology. Khatri *et al.* (2012) present

a comprehensive review from the perspective of computational biology, describing three major classes of analysis techniques, from the earliest enrichment methods (over-representation analysis), through more quantitative functional class scoring methods, and on to methods that recognize internal relationships within each set. Goemann and Bühlmann (2007), Barry *et al.* (2008), Ackermann and Strimmer (2009), and Maciejewski (2013) survey statistical considerations, especially for Khatri’s first two classes. These papers shed light on hypothesis testing issues and multivariate analysis issues, including on the various ways that permutations and bootstraps may be used to assess statistical significance. Virtually all the reviewed methods focus on gene sets considered one at a time, and in this sense they are *uni-set* methods. Of course individual gene sets contain multiple genes, and so uni-set methods are multivariate from the perspective of gene-level data. Analysts recognize that gene sets are organized in larger assemblages, but this fact does not inform uni-set inference computations except possibly through post-hoc adjustment for false discovery rate control. We review uni-set statistics and calibration issues in Section 3.

More recently, methods have emerged that aim to improve overall inference by simultaneously processing all the sets in a collection. Simultaneous analysis is known to have benefits in other domains of high-dimensional inference (e.g., empirical Bayes estimation), and this *multi-set* approach has the potential to deal with persistent inference challenges of gene set analysis. One challenge is that *set size* affects testing power, and so even if procedures are calibrated on a null hypothesis, there remains a power imbalance on the alternative hypothesis (Newton *et al.* 2007). This imbalance affects how we prioritize or rank order the most interesting gene sets in a given

analysis. For instance we might rank by uni-set p-values, looking at the sets having the smallest p-values as the ones worth reporting. An artifact emerges because large sets more easily yield small p-values, all else being equal.

A more substantial challenge is the *set overlap problem*. In addition to statistical dependencies between sets sharing genes, there may be spurious associations and problems with redundancy. For example, many methods would consider a gene set S to be non-null (and thus worthy of reporting) if any of its contained genes exhibits a difference in mean expression between two cellular states. The cause of the differential expression of gene $g \in S$ might be the altered activity of a molecular pathway that is encoded by the set S . Now any other set S' which happens to contain g is non-null by association, even if the function represented by S' is unaltered. Thus pleiotropy of a gene can lead to spurious gene set associations (Bauer *et al.* 2010; Newton *et al.* 2012).

While the multi-functionality of genes is one expression of the set overlap problem, extensive overlap also happens because biological properties can be expressed at different levels of resolution and from different perspectives (e.g., Gillis and Pavlidis, 2013). The upshot for a uni-set statistical analysis is that reported findings may exhibit substantial redundancy, and one fails to achieve a concise functional description of the gene-level data D . Some bioinformatic tools address this problem by working with trimmed collections of gene sets (e.g., GO SLIMS), by invoking post-hoc heuristics to simplify reporting (e.g., Supek *et al.* 2011), or by adjusting a uni-set p-value based on closely related sets (e.g., Alexa *et al.* 2006; Grossman *et al.* 2007). None of these solutions is entirely satisfactory. One loses the full information content and

fine granularity of knowledge K when using a trimmed collection; we risk masking functional signals when using the rule-based clustering schemes available for post-processing uni-set output; and available p-value adjustment schemes are cumbersome and use only a fraction of the actual set overlap information.

In Section 1.5 we discuss some multi-set statistical approaches aimed at detecting functional signals via genome-wide modeling the gene-level data. The methods adhere to the following basic premise: *in addition to using gene-level data on genes within a given gene set, inference regarding this set should incorporate knowledge about the other sets containing these same genes*. This incorporation occurs explicitly through a parsimonious probability model describing how gene-level data are produced as a consequence of activity patterns of biological functions represented by the gene sets. As with any model-based approach, there is a cost in terms of model simplifications which may be at odds with the reality of data generation. On the other hand, the model assumptions are explicit, and so may be evaluated; the inference summaries may be relatively insensitive to them; and the joint modeling approach may be the only way to enable probabilistic reasoning for multi-set analysis. We illustrate two of the model-based multi-set calculations in a gene expression study of nasopharyngeal carcinoma.

1.3 Gene set repositories

When evidence links genes by virtue of some shared biological property, these genes comprise an archivable gene set. Large-scale collaborative research efforts are underway

to encode such biological knowledge. Other efforts, notably **Bioconductor** (Gentleman *et al.*, 2004) provide open software systems for accessing these data.

Gene Ontology (GO): From the Gene Ontology Consortium (2000), the GO project aims to provide a structured and controlled vocabulary for describing genes and their products across different species. The GO project has the following three main goals: first, to compile and maintain the structured and controlled ontologies by virtue of shared model organism databases; second, to describe the roles of genes and their products using these ontologies; third, to develop tools for querying and manipulating the ontologies (e.g., the GO Consortium has developed AMIGO for searching and browsing GO). Each term (i.e., set) in GO is assigned to one of the three root ontologies: molecular function (the function of a gene product at the biochemical level), cellular component (the place in the cell where a gene product is active) and biological process (a biological objective to which the gene product contributes). These three domains are organized into three directed acyclic graphs, wherein each node is a functional category (GO term) and directed edges convey proper-subset information. The terms in each ontology are linked to one another by two types of relationships: “is-a”, which represents a simple class-subclass relationship and “part-of”, which refers to a component relationship. GO is not static. The initial GO project only involved three model organism databases: SGD, the *Saccharomyces* Genome Database; FlyBase, the *Drosophila* Genome Database; and MGD, the Mouse Genome Database. After that, more and more model organism databases were incorporated into the project. Now GO has become a standard knowledge base for integration and interpretation of large-scale molecular datasets. At writing, GO

contains over 34400 terms reflecting diverse biological function in a large number of organisms (Bioconductor, GO.db, version 2.10.1). Figure 1.1 renders a small piece of GO: it presents the molecular functions containing 5-10 genes, together with their association genes, in a bipartite graphical representation.

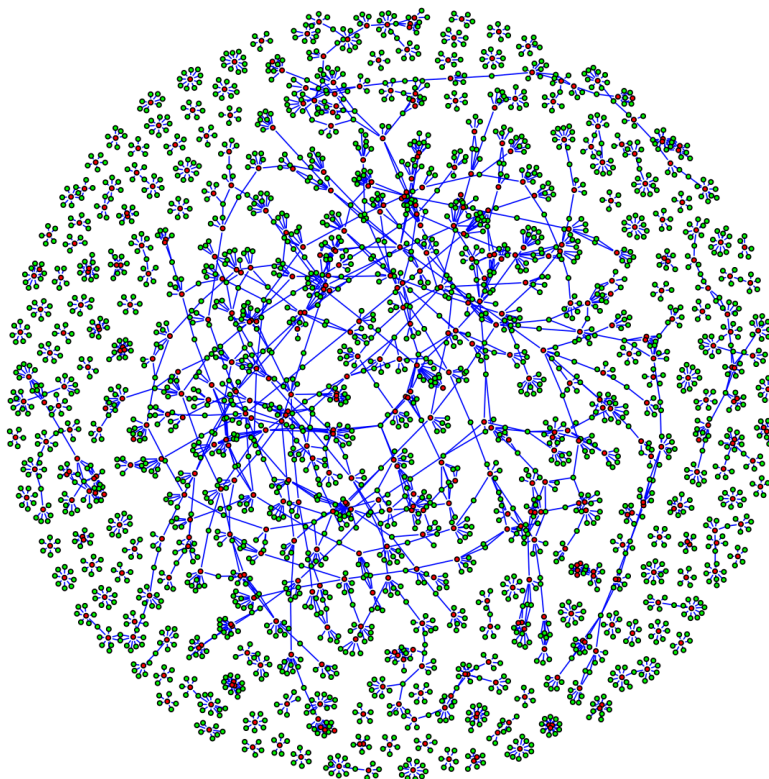


Figure 1.1: 543 Gene Ontology (GO) terms (*red*) representing all molecular functions (MFs) holding between 5 and 10 human genes and the associated 2,613 genes (*green*), presented as a bipartite graph. The edges (*blue*) connect genes to the sets that contain them. This graph has 134 components, including one large component with over 1,700 nodes. (Based on data in `org.Hs.eg.db` version 2.8.0.)

Kyoto Encyclopedia of Genes and Genomes: KEGG (Kanehisa and Goto, 2000) is an effort to link the genomic and molecular information with high-level

functional information (e.g., cell, organism and ecosystem), and thus to produce a computer representation of the biological system. KEGG was initiated by the Japanese human genome program in 1995, and since then it has been developed and extended; now it maintains sixteen main databases. These databases are related to different respects of the biological system, which include systems information, genomic information, chemical information and health information. KEGG has been widely used to infer the functional significance within large-scale genetic studies.

Reactome: REACTOME (Joshi-Tope et al., 2005) is a curated, peer-reviewed knowledge base of human biological processes. REACTOME aims to provide an integrated view of biological processes, which captures what is already known about the interactions between genes, proteins and molecules by using a data model that is accessible to computation. The REACTOME project also provides tools for visualizing and interpreting the large-scale experimental datasets by using REACTOME pathways. The core unit in the REACTOME data model is a reaction in which input physical entities are converted to output entities; then reactions linked together by shared physical entities form a biological pathway. These reactions are gathered by experts in the field, peer-reviewed, and edited by REACTOME team members prior to being published in the database. According to Croft et al. (2013), the latest version of REACTOME includes 7088 human proteins, which participated in 6744 reactions based on data extracted from 15107 research publications with PubMed links.

Molecular Signature Database (MSigDB): MSigDB (Liberzon et al., 2011), which mainly focuses on human gene sets, is a curated database created particularly

for the GSEA method (Subramanian et al., 2005). Compared to other gene set sources, MSigDB includes more diverse and more types of gene sets, which include not only biological pathways collected from original publications but also gene sets derived directly from other known data sources including GO, KEGG, and REACTOME. MSigDB version 4.0 includes 10925 gene sets and a richer set of annotations.

1.4 Uni-set statistics

The structure of genome-wide gene-level data $D = \{D_g\}$ has considerable bearing on what statistics we compute on a set of genes. At one extreme, gene g 's data D_g records raw data from a multi-sample experiment, such as expression levels from a number of microarrays on cells under various experimental conditions. At another extreme, D_g may only record whether or not gene g was output on a list of genes deemed relevant in the study; for example D_g may report the decision from a gene-level hypothesis test. Efforts at data integration support the first extreme, whereby as much available raw data as possible are incorporated into the set-level analyses. On the other hand, practical considerations may limit access to such raw data or may present substantial complexities in case we attempt to model them, and these considerations then support analyses using simple gene lists. Both extremes as well as various intermediate cases actually occur in practice.

Regardless of the structure of data D_g , two quite different modes of comparison are possible when evaluating a gene set S . As articulated by Goemann and Bühlmann (2007), we can (1) compare what is in S with what is outside of S , or (2) we can

compare what is in S with what we might have expected to be in S if some null hypothesis had held on the distribution of D_g for $g \in S$. The former is called *competitive*; the latter *self-contained*. Tian *et al.* (2005) make a similar distinction, which is relevant not only to what test statistic we use to score a set S , but also how we calibrate the statistical significance of that score.

As to the construction of set-level test statistics, many methods follow a two-stage format (e.g., Barry *et al.* 2008; Ackermann and Strimmer, 2009). Data on gene g are reduced to a local real-valued statistic, say T_g ; then local statistics are combined into a real-valued set-level statistic, say U_S , either by averaging or some rank-based combination. The SAFE score (Barry *et al.* (2005) and the GSEA score (Subramanian *et al.*, 2005) are prominent examples; the max-mean statistic is another (Efron and Tibshirani, 2007). Recent improvements include the self-contained ROAST score (Wu *et al.* 2010) and the competitive CAMERA score (Wu *et al.*, 2012). In a curious reversal, the LRPATH score by Sartor *et al.* (2009) treats the set as a binary response and the gene level scores T_g as predictor values in a logistic regression model. Other methods are more explicitly multivariate: they focus on the vectors holding gene-level scores in each biological sample, and address possible affects on the joint distribution of this vector caused by various treatment factors. These have not proven more useful than the two-stage scores above, owing to challenges with the estimation of covariance (Ackermann and Strimmer, 2009). Both the two-stage and multivariate methods are examples of functional class scoring statistics, according to the classification by Khatri *et al.* (2012). In case gene-level scores T_g are simple indicators that gene g is reported on a short list of interesting genes, we are in a

somewhat simpler domain; the only useful second-stage score is a count, such as $U_S = \sum_{g \in S} T_g$. When U_S is unusually large we say S is enriched for interesting genes or, equivalently, the list of interesting genes is enriched for type S genes. Methods based on assessing over-representation remain popular, in part due to their simplicity (Khatra *et al.* 2005, 2012).

Having constructed a set-level score $U_S(D)$ the next question is how to calibrate it; answering this tells something about how interesting are the data in S , either in isolation or in comparison with those in other sets. Considerable attention has been paid to the calibration problem. Possible solutions include permuting samples, permuting genes, bootstrapping, or invoking some other randomization aiming to respect whatever null hypothesis is in view.

1. **Sample permutation:** This is applicable when the data structure involves multiple samples, labeled by some kind of phenotypic response, such as when multiple expression profiles are labeled by the cell type or treatment applied to the cells under investigation. Then the local T_g is some test statistic measuring association between the gene level data and the label. Sample permutation amounts to shuffling the sample labels and recomputing the T_g 's to develop a reference distribution for $U_S(D)$. Here, the set S is fixed as are all the gene level data D_g *except* the sample labels. A benefit of sample permutation is that any dependencies among genes within S are respected. A weakness is in the stringency of the underlying null hypothesis, which asserts that all genes $g \in S$ present data that is independent of the phenotypic label. Problems with pleiotropic effects and power imbalance complicate the interpretation of such

self-contained analyses when applied to multiple sets.

2. **Gene permutation:** This is applicable over a wider range of data structures, as it requires only gene level scores $\{T_g\}$. Gene permutation is not intended to work on the same stringent null hypothesis of sample permutation; indeed it acknowledges that some fraction of genes g in set S may have an association between genomic data and sample label. At issue, rather, is whether the extent of this association is unusually large, compared in some way to the rest of the genome outside of S . A weakness of gene permutation is that it does not respect any particular internal structure of data D_g (e.g. dependence among genes within each set), and thus type-I error rates can be inflated. Yet it remains a dominant competitive approach, possibly because of its flexibility with data structures or possibly owing to the exploratory manner in which it is deployed. With gene-level data fixed, gene permutation simply asks how the observed $U_S(D)$ compares to the same statistic computed on random sets S^* of the same size as S .
3. **Other calibrations:** On the self-contained side, the ROAST method (Wu *et al.* 2010) utilizes a form of conditional inference suitable to ANOVA-style decompositions of the gene-level data. Instead of sample permutation it relies on an interesting scheme by which simulated data have the same covariance pattern as observed data within the set. On the competitive side, the non-parametric bootstrap can accommodate dependence-respecting tests, such as of the null hypothesis that the average gene-level effect within a set S is the same as the average over the whole genome (Barry *et al.* 2008; Dudoit *et al.* 2007). The

small sample sizes associated with many genomic studies corresponds to high approximation error for bootstrap methods. Alternatively, Wu and Smyth (2012) derive competitive statistics $U_S(D)$ by estimating the variance inflation effect caused by dependence and then by referring to a suitable t -approximating null distribution.

1.5 Multi-set statistics

Role modeling

GENGO (Lu *et al.* 2008), MGSA (Bauer *et al.* 2010, 2011), and MFA (Wang *et al.* 2015) are three distinct inference tools for multi-set analysis. They are not readily described as either competitive or self-contained tools; they aim to identify interesting gene sets via modeling of gene-level data in terms of latent states of biological functions associated with the gene sets, which they infer simultaneously by some version of Bayesian inference. Their utility rests in part with the simplicity of both the model structure and the gene-level data structure, though more fully elaborated models could further the general data integration task. Owing to their multivariate stance, GENGO, MGSA, and MFA offer solutions to difficulties caused by power imbalance and gene set overlap.

MGSA and MFA rest on the same generative probability model for gene-list data. The GENGO model is very similar. The biggest differences among the methods have to do with how inference is deployed and on their operating characteristics. We describe first the basic generative model. We assume that the genome-wide gene-

level data D have been reduced to a gene list; in other words gene level indicators $D = \{D_g\}$, where

$$D_g = \begin{cases} 1 & \text{if gene } g \text{ exhibits an interesting feature in endogenous data} \\ 0 & \text{if not.} \end{cases}$$

As with any gene-list data, the D_g may be the result of a gene-level hypothesis test computed from more extensive data. The context guides the language, but we might say that D_g indicates whether or not gene g seems to be *activated* in the experimental situation, or, equivalently, if it seems to be *non-null*. It is helpful also to recognize that hypothesis tests may incur either false-positive or false-negative errors, and so the actual null/non-null or inactive/active status is recorded in unobserved indicators:

$$A_g = \begin{cases} 1 & \text{if gene } g \text{ is active (non-null)} \\ 0 & \text{if not.} \end{cases}$$

The first part of the *role model* simply recognizes gene-level hypothesis testing errors:

$$D_g|A_g \sim \text{Bernoulli} \begin{cases} \alpha & \text{if } A_g = 0 \\ \gamma & \text{if } A_g = 1 \end{cases}$$

These parameters record the false-positive rate α and the true positive rate γ ; together $\theta = (\alpha, \gamma)$. Some presentations use $\beta = 1 - \gamma$ as the false-negative rate. Typically we have $\alpha < \gamma$, and so truly active genes are reported to the list at a higher rate than truly inactive genes. MGSA, and MFA treat the estimation of the error-rate parameters

differently: MFA takes user input, recognizing the utility of external information, for example about the targeted false discovery rate of the gene-level tests. MGSA allows user input but alternatively produces estimates from fitting the gene-list data. In any case, both methods use the same Bernoulli sampling model for D_g given A_g . They further assume mutual independence within $D = \{D_g\}$ conditionally upon the full list of activations $A = \{A_g\}$; dependencies are assumed to arise only from the joint distribution of A . Future elaborations of these tools could accommodate error dependencies. In both tools, the log-likelihood from the *observation component* of the sampling model is

$$\begin{aligned}
l_{obs}(A, \theta) &= \log \Pr(D|A, \theta) \\
&= \log \prod_g \Pr(D_g|A_g, \theta) \\
&= \sum_g \{A_g [D_g \log(\gamma) + (1 - D_g) \log(1 - \gamma)] + \\
&\quad (1 - A_g) [D_g \log(\alpha) + (1 - D_g) \log(1 - \alpha)]\} \\
&= N_{1,1} \log \gamma + N_{1,0} \log(1 - \gamma) + N_{0,1} \log \alpha + N_{0,0} \log(1 - \alpha). \quad (1.1)
\end{aligned}$$

Here $N_{1,1} = \sum_g A_g D_g$ counts the truly active genes that appear active (i.e., are reported); $N_{1,0} = \sum_g A_g (1 - D_g)$ counts the truly active genes that are not reported; $N_{0,1} = \sum_g (1 - A_g) D_g$ counts the truly inactive genes that are falsely reported, and $N_{0,0} = \sum_g (1 - A_g) (1 - D_g)$ counts the truly inactive genes that are correctly not reported.

The central structure of the *role model* relates gene activities $A = \{A_g\}$ to more

primitive binary set activities $Z = \{Z_S\}$:

$$Z_S = \begin{cases} 1 & \text{if set } S \text{ is active (non-null)} \\ 0 & \text{if not.} \end{cases}$$

The notion is that $Z_S = 1$ corresponds to an active role played by genes in S within the cells on test, and the gene's activity is inherited from any sets it's part of (roles it may have):

$$A_g = \begin{cases} 1 & \text{if set } Z_S = 1 \text{ for any } S \text{ with } g \in S \\ 0 & \text{if not.} \end{cases} \quad (1.2)$$

More concisely, $A_g = \max_{S: g \in S} Z_S$. In this way, the gene activities $A = A(Z)$ arise as a mapping from the set activities. Figure 1.2 express this inheritance on a small component of the bipartite graph expressing GO molecular functions.

In model-based gene set analysis (MGSA), the unknown Z_S are modeled as independent and identically distributed Bernoulli variables, with success probability π , thus contributing a log probability mass

$$l_{hidden}(Z, \pi) = \sum_S \{Z_S \log(\pi) + (1 - Z_S) \log(1 - \pi)\}. \quad (1.3)$$

In a Bayesian analysis, the joint posterior of all unknowns has logarithm:

$$l_{posterior}(Z, \theta, \pi) = l_{obs}(A(Z), \theta) + l_{hidden}(Z, \pi) + \text{constant}. \quad (1.4)$$

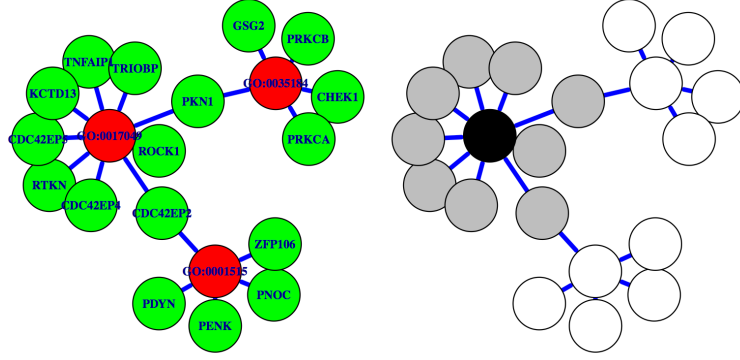


Figure 1.2: One component from Figure 1.1 involving 3 Gene Ontology (GO) terms and 17 genes (*left*). The right panel shows one realization of latent activity states, with an activated term (*black*) inducing activity on all the contained genes (*gray*). Both model-based gene-set analysis (MGSA) and multifunctional analysis (MFA) use the set-to-gene activity mapping expressed in the right panel. MFA constrains latent activities to guarantee a unique gene-to-set mapping; obtaining such a mapping becomes an issue in more complex domains of the graph. Gene-level data are Bernoulli trials having a low rate (*white*) or a higher rate (*gray*).

The MGSA tool samples the joint posterior via Markov chain Monte Carlo, and scores sets according to marginal posterior activation probability:

$$U_S(D) = P(Z_S = 1|D). \quad (1.5)$$

Empirical evidence shows that scoring sets by marginal posterior probability substantially reduces the redundancy problem and results in reported sets with minimal overlap (Bauer *et al.* 2010; Wang *et al.* 2015).

It is helpful to reflect on why the computation in (1.5) is difficult in complex systems of sets. Where overlap and size variation limit the utility of uni-set analysis, they simply complicate the computations of multi-set analysis. This is because the

apparent activation, $D_g = 1$, of some gene, supposing it's not a false positive, must be attributable to the non-null activity of some containing set. In evaluating a given set S , the calculation uses all the D_g values for $g \in S$, but it must also recognize other sets S' containing such genes g , since the activation of such sets might better explain the data. Of course data on other genes in such sets S' have a bearing on the issue of activation of S' ; we quickly find ourselves regressing away from the original set S in trying to fully evaluate the evidence supporting $Z_S = 1$; only a posterior probability computation enables this assessment. In Newton *et al.* (2012) the complexity of the posterior computation was investigated; when the intersection graph of the sets is sufficiently simple then message-passing algorithms (e.g, the junction tree algorithm) enable exact marginal posterior computation. In systems as complex as GO these exact algorithms are no longer practical.

Newton *et al.* (2012) also investigated the mapping $Z \longrightarrow A$ in (1.2), observing that in general it is not invertible, but that it becomes invertible when we refine the definition of *activity*. The *activation hypothesis* (AH) asserts that a set of genes is active (non-null) if and only if each gene in the set is active. Under AH, the inverse mapping $A \longrightarrow Z$ is:

$$Z_S = \min_{g \in S} A_g, \tag{1.6}$$

which finds that the set must be active if it contains only active genes. Considering the directed graphical structure of GO, it may be reasonable to assert AH. Take two sets: a more specific set A and a less specific set B , with $A \subset B$. The role-model assumption (1.2) transfers activity to all genes $g \in A$ in case the larger set is non-null:

$Z_B = 1$. Genes $g \in A$ inherit their inclusion in set/property/term B by virtue of B describing a less specific annotation. Without AH, we have to deal with the possibility that all genes g with property A are non-null, but somehow the set of genes with property A is null.

The non-invertibility of the $Z \rightarrow A$ mapping affects the operating characteristics of the posterior summaries. To measure the effect, Wang *et al.* (2015) formulated a slightly different prior, P_2 , as an alternative to the i.i.d. Bernoulli prior P_1 for $Z = \{Z_S\}$, encoded in (1.3). Namely

$$P_2(Z = z) = P_1(Z = z | Z \in \text{AH})$$

where $z = \{z_S\}$ is a vector of possible binary set-level activation states. By conditioning the i.i.d. prior on the constraint AH, one greatly alters the distribution of probability mass over the high-dimensional space of latent activity states. The P_1 probability of AH is small for complex collections (and getting smaller, according to the analysis in Wang *et al.* 2015). It's as if, like water leaking from a system of pipes, the prior mass distributed by P_1 has leaked into regions of the state space that ought to be avoided. The cost of fixing this leak is that posterior computations under the P_2 prior must respect the constraints imposed by AH. Wang *et al.* (2015) showed that AH is equivalent to a set of linear inequality constraints on the joint collection of activities, $\{\{A_g\}, \{Z_S\}\}$, and further that the log posterior (1.4) is a linear function of this augmented set of variables. The multi-functional analyzer (MFA) tool utilizes a novel MCMC scheme to approximate $U_S(D) = P_2(Z_S = 1 | D) = P_1(Z_S = 1 | D, \text{AH})$ and also deploys integer-linear programming (ILP) to find the joint maximum a

posteriori (MAP) estimate $\hat{Z} = \{\hat{Z}_S\}$ by maximizing a linear function in binary variables subject to linear inequality constraints. Initial numerical studies reported in Wang *et al.* (2015) suggest that MFA has improved sensitivity compared to MGSA.

Example

Sengupta *et al.* (2006) reported a genome-wide expression study of nasopharyngeal carcinoma (NPC), with particular attention paid to unusually extensive negative associations between host genes in the NPC cells and the expression of a key gene in the infecting Epstein-Barr virus (EBV). Gene-set analysis was used in that study to guide the interpretation of genome-wide findings and to inform follow up experiments. The gene-level Spearman correlations between host genes and the EBV gene are available in the `allez` package, which performs basic uni-set computations (Newton *et al.* 2007). Here we reconsider these data in order to illustrate multi-set computations.

Genes showing extreme negative Spearman correlation were entered onto the gene list; fully 438 genes (Entrez ID) constituted the 5% FDR-controlled list. We integrated this list with GO[5:50], 5994 GO terms that hold between 5 and 50 human genes. GO[5:50] itself annotates only about half of the human genes (10293), and only 232 of the genes showing NPC-EBV association. We applied MFA and MGSA to assess the functional content of this reduced NPC-EBV gene list. For both methods, we fixed the false positive rate α and the true positive rate γ using a simple mixture argument applied to available gene-level data. The mixture argument goes as follows: the marginal probability $P(D_g = 1)$ is estimated by the relative size of the gene-list; the marginal inactivation rate $P(A_g = 0)$ is estimated by the average, over all genes,

of the local FDR $P(A_g = 0 | \text{correlation}_g)$, as computed by the `locfdr` package (Efron, 2004). Restriction of this averaging to genes reported on the top list gives an estimate of $P(A_g = 0 | D_g = 1)$ and restricting to those not so reported gives an estimate of $P(A_g = 0 | D_g = 0)$. The true and false positive rates follow by applying Bayes's rule, as the R code below shows:

```
## R version 3.0.1; allez version 2.0.1

data(npc) ## NPC Spearman correlations between host expression and virus
scores <- (1/2)*sqrt(28)*log( (1-npc)/(1+npc) ) ## variance stabilized
pv <- pnorm( scores, lower.tail=FALSE ) ## nominally N(0,1) on H0
qv <- p.adjust(pv, method="BH" ) ## Benjamini Hochberg
ok <- (qv <= 0.05) ### probe sets at 5% FDR by Benjamini Hochberg

## now consider locfdr; for mixture estimation
library(locfdr)
fit <- locfdr( scores, nulltype=0 , plot=0)
u <- fit$fdr

## A = gene level activity indicator
## D = gene level call

pD1 <- mean(ok) ## about 1% of genes reported
pA0 <- mean(u) ## about 69% of genes are truly null
pA0gD1 <- mean( u[ok] ) ## P(A=0|D=1), FDR about 4% (good, since BH says 5%!)
pA0gD0 <- mean( u[!ok] ) ## P(A=0|D=0), about 70%

## now Bayes rule to get P(D|A)
```

```
pD1gA0 <- pA0gD1*pD1/pA0          ## alpha = false positive rate 0.00061
pD1gA1 <- (1-pA0gD1)*pD1/(1-pA0)  ## gamma = true positive rate 0.03297
```

The list of annotated HPV-associated genes was integrated with GO, fixing the error rates estimated above in both MGSA and MFA. Table 1.1 summarizes the gene sets inferred to be active by MFA-ILP, that is, the MAP estimate of the active states. 44 sets constitute the MAP estimate. The uni-set analysis previously reported in Sengupta *et al.* (2006) is not directly comparable to Table 1.1, as it worked with a simpler collection that included relatively larger sets. We do not attempt a detailed analysis of the identified functions, except to say that (a) some of them recapitulate in further detail the broad classes previously reported (e.g., GO:0005031 and GO:2001238 are particular aspects of the cell death function) and (b) understanding the role of these functions in NPC is part of ongoing research. Our primary objective, rather, is to use the NPC example as a platform to compare MFA and MGSA, which rely on similar statistical assumptions but yield different findings.

To control the comparison, we used the MGSA estimate of π to drive MFA calculations, both optimization and marginal posterior computation. Table 1.1 reports MGSA-computed marginal posterior probabilities assigned to sets in MAP estimate by MFA. Considering MCMC characteristics, very little of the discrepancy can be explained by Monte Carlo error, and therefore represents differences induced by the different prior assumptions. As was also reported in Wang *et al.* (2015), MFA and MGSA both produce relatively non-overlapping gene sets (very little redundancy).

Table 1.1: NPC example: 44 GO terms are inferred to be active by the MFA MAP estimate, and ordered by marginal posterior activation probability. Basic statistics on these terms are provided (# NPC-associated genes/set size). The right-most columns give the MCMC-computed marginal posterior activation probabilities for these terms, using MGSA and MFA.

GOID	Term (up to 40 chars)	Statistics	P.MFA	P.MGSA
GO:0005070	SH3/SH2 adaptor activity	4/49	1.00	0.98
GO:0030672	synaptic vesicle membrane	4/45	0.96	0.77
GO:0015377	cation:chloride symporter activity	2/8	0.93	0.16
GO:0033179	proton-transporting V-type ATPase, V0 do	2/10	0.93	0.20
GO:0005031	tumor necrosis factor-activated receptor	2/12	0.93	0.04
GO:0005154	epidermal growth factor receptor binding	3/20	0.92	0.84
GO:0015459	potassium channel regulator activity	2/27	0.91	0.62
GO:0004957	prostaglandin E receptor activity	2/5	0.89	0.01
GO:0005545	1-phosphatidylinositol binding	2/20	0.88	0.64
GO:0004708	MAP kinase kinase activity	2/15	0.85	0.18
GO:0030127	COPII vesicle coat	1/9	0.84	0.00
GO:0051642	centrosome localization	2/9	0.83	0.64
GO:0004091	carboxylesterase activity	2/29	0.81	0.61
GO:0005544	calcium-dependent phospholipid binding	3/22	0.79	0.54
GO:0031235	intrinsic to internal side of plasma mem	2/14	0.78	0.18
GO:0005245	voltage-gated calcium channel activity	2/33	0.78	0.23
GO:0012507	ER to Golgi transport vesicle membrane	6/25	0.74	0.51
GO:0001937	negative regulation of endothelial cell	2/23	0.72	0.22
GO:0005044	scavenger receptor activity	2/44	0.70	0.53
GO:0019388	galactose catabolic process	2/5	0.68	0.44
GO:0043027	cysteine-type endopeptidase inhibitor ac	2/21	0.67	0.29
GO:0001725	stress fiber	2/42	0.66	0.11
GO:0051016	barbed-end actin filament capping	2/7	0.64	0.06
GO:0030574	collagen catabolic process	2/22	0.64	0.34
GO:0035085	cilium axoneme	2/47	0.63	0.25
GO:0002020	protease binding	2/50	0.63	0.26
GO:0005801	cis-Golgi network	2/26	0.61	0.26
GO:0003950	NAD+ ADP-ribosyltransferase activity	2/22	0.58	0.45
GO:0060397	JAK-STAT cascade involved in growth horm	3/24	0.57	0.16
GO:0045648	positive regulation of erythrocyte diffe	3/15	0.55	0.15
GO:0043401	steroid hormone mediated signaling pathw	2/14	0.55	0.12
GO:0032747	positive regulation of interleukin-23 pr	2/5	0.52	0.23
GO:0050957	equilibrioception	1/7	0.46	0.01
GO:0006685	sphingomyelin catabolic process	2/5	0.44	0.01
GO:2001238	positive regulation of extrinsic apoptot	2/15	0.42	0.17
GO:0001961	positive regulation of cytokine-mediated	2/16	0.40	0.09
GO:0030175	filopodium	4/50	0.37	0.93
GO:0090184	positive regulation of kidney developmen	2/13	0.36	0.06
GO:0050885	neuromuscular process controlling balanc	4/41	0.35	0.90
GO:0032266	phosphatidylinositol-3-phosphate binding	3/17	0.34	0.93
GO:0071300	cellular response to retinoic acid	3/46	0.33	0.09
GO:2000725	regulation of cardiac muscle cell differ	2/7	0.26	0.00
GO:0001916	positive regulation of T cell mediated c	2/11	0.03	0.02

An interesting feature of MFA is the coverage of the 232 NPC-associated genes:

$$\begin{aligned}\text{coverage} &= \sum_g \hat{A}_g D_g \\ \text{miscoverage} &= \sum_g \hat{A}_g (1 - D_g)\end{aligned}$$

where $\hat{A}_g = 1$ if gene g is part of any of the inferred active sets. MFA covers 99 genes; the MGSA estimate covers 72 genes using 25 sets (14 in common with MFA; see Table 1.2). MFA miscovers more than MGSA (789 vs 673), but the coverage/miscoverage ratio favors MFA, as it has done in all the examples considered to date. Both methods cover the reported gene list much better than the standard uni-set method (Fisher’s test). For example, if we rank sets by Fisher p-value, then in accumulating enough to cover 99 NPC-associated genes we would miscover 869 genes.

1.6 Summary

The integration of experimental genome-wide gene-level data with pre-existing data on the functional properties of the same genes remains a central challenge in genomic data analysis. International efforts, including **GO** and **Bioconductor**, continue to develop computational knowledge systems for the organization and analysis of genomic data. These systems rely on statistical methodology to enable the design of effective data analysis tools. From the extensive literature on such methodology we discussed both uni-set and multi-set statistics in this brief chapter: uni-set statistics process gene-level data within each set to derive set-level scores; these are calibrated according

Table 1.2: NPC example: 25 Gene Ontology (GO) terms inferred to be active by model-based gene set analysis (MGSA).

GOID	Term (up to 40 chars)	Statistics	P.MGSA	P.MFA
GO:0005070	SH3/SH2 adaptor activity	4/49	0.98	1.00
GO:0030175	filopodium	4/50	0.93	0.37
GO:0032266	phosphatidylinositol-3-phosphate binding	3/17	0.93	0.34
GO:0031228	intrinsic to Golgi membrane	3/48	0.91	0.00
GO:0050885	neuromuscular process controlling balanc	4/41	0.90	0.35
GO:0033209	tumor necrosis factor-mediated signaling	3/36	0.89	0.00
GO:0005154	epidermal growth factor receptor binding	3/20	0.84	0.92
GO:0030672	synaptic vesicle membrane	4/45	0.77	0.96
GO:0015296	anion:cation symporter activity	3/31	0.75	0.00
GO:0015932	nucleobase-containing compound transmemb	3/19	0.71	0.00
GO:0008484	sulfuric ester hydrolase activity	2/16	0.67	0.02
GO:0034109	homotypic cell-cell adhesion	3/37	0.66	0.00
GO:0005545	1-phosphatidylinositol binding	2/20	0.64	0.88
GO:0051642	centrosome localization	2/9	0.64	0.83
GO:0021954	central nervous system neuron developmen	3/43	0.62	0.00
GO:0015459	potassium channel regulator activity	2/27	0.62	0.91
GO:0004091	carboxylesterase activity	2/29	0.61	0.81
GO:0006595	polyamine metabolic process	2/20	0.61	0.00
GO:0005100	Rho GTPase activator activity	4/35	0.58	0.06
GO:0017022	myosin binding	3/28	0.58	0.03
GO:0043392	negative regulation of DNA binding	2/32	0.56	0.07
GO:0005544	calcium-dependent phospholipid binding	3/22	0.54	0.79
GO:0005044	scavenger receptor activity	2/44	0.53	0.70
GO:0004629	phospholipase C activity	2/36	0.51	0.02
GO:0012507	ER to Golgi transport vesicle membrane	6/25	0.51	0.74

to some reference distribution and possibly corrected for multiple-testing as a last step. Multi-set statistics are multivariate in the sets themselves; model-based multi-set statistics entail jointly modeling genome-wide gene-level data. On the one hand these present an opportunity to overcome persistent difficulties caused by gene-set overlap and set-size variation; on the other hand the current formulations remain extremely simplistic in their modeling of data integration. Further research is needed to better understand the effects of: (1) prior structure, (2) error dependencies, (3) gene-level data reduction, and (4) annotation uncertainty, among other issues. On (1), for

example, MGSA's i.i.d. Bernoulli prior for set activities induces a non-i.i.d. prior on gene activities; we know that constraining the i.i.d. prior has a substantial effect, and so the uniformity scale (set vs gene) may also have an effect that requires examination. As with so much model-based analysis, the scope of potential computations and diagnostics is limited only by the imagination; the multi-set modeling approach will gain traction through successful application in genomic examples. Examples are still too few at present; many opportunities present themselves, and many more will do so given the continual elaboration of the genomic knowledge base. We hope that through the development of open software tools (R packages cited in Bauer *et al.* 2011; Wang *et al.* 2015), a more extensive empirical investigation will yield further insights into this challenging data integration problem.

Chapter 2

A Multi-Functional Analyzer using Parameter Constraints

2.1 Forward

Integrating experimental genomic data with exogenous functional information is important in statistical genomics for the purposes of effective data reduction and boosting weak gene-level signals. Chapter 1 classifies the majority of gene set analysis methods into uni-set and multi-set analysis. The uni-set analysis methods consider one gene set at a time which ignores the complexity of the functional record. Also, the uni-set analysis fails to incorporate the overlapping structure of categories and the output is usually hard to interpret. In contrast, the multi-set approaches consider all the gene sets simultaneously and aim to find the best combination of gene sets to explain the observed biological response. In particular, we are most interested in

answering the following three questions:

- (a) Can we propose a model that incorporates the overlapping and hierarchical structure of functional categories and generates inferences that respect this structure?
- (b) Compared to existing model-based methods, does our model perform better in detecting subtle signals?
- (c) Can we develop an efficient algorithm to apply the proposed method to large-scale gene-set analysis problems?

This chapter is organized to provide evidence in response to these questions. The main findings, which were published in Wang et al. (2015), are recapitulated here. The novel MCMC scheme adopted and studied here was conceived by Dr. Bret Larget. Our method is called multi-functional analyzer, or MFA for short. Briefly, MFA provides a model-based analysis of a gene list. A gene’s presence on the list is explained by latent binary activities of GO terms which annotate it; through model-based computations MFA determines the posterior activity probabilities over all terms in the collection. It refines model-based gene set analysis (MGSA; Bauer *et al.*, 2010) by encoding natural constraints on the function-level activities in order to improve statistical efficiency. MFA imposes an important model assumption called the activation hypothesis to assure identifiability. It addresses deficiencies of simpler gene-set methods (e.g., in handling set overlaps) by the simultaneous analysis of all sets in a collection. MFA develops two computational advances to enable the posterior inference within the constrained parameter space: one using integer linear programming for optimization, and the other using a penalized Markov chain sampler. Simulations (Section 2.5) are conducted to demonstrate the advantages of MFA over

MGSA and Fisher’s test. Two case studies (Section 2.6) are examined: one on a genome wide association study (GWAS) of type 2 diabetes and the other is on RNA interference studies of influenza.

2.2 Role model

What’s role model?

Following the description in Newton *et al.* (2012), we have a finite number of *parts* p and a finite number of *wholes* w , where each whole is an unordered set of parts. The incidence matrix $I = (I_{p,w})$ is assumed to be known, where $I_{p,w} = 1$ if and only if $p \in w$. The intended correspondence is that genes are parts and gene sets (*i.e.*, functional categories) are wholes. The matrix I encodes a full collection of gene sets. We will have measured data on the parts and aim to make inference on properties of the wholes.

The experimentally-derived gene list may be viewed as a vector of Bernoulli trials $X = (X_p)$, with $X_p = 1$ if and only if part (gene) p is on the list. First proposed in Bauer *et al.* (2010), the role model describes the joint distribution of X in terms of latent binary (0/1) activity variables $Z = (Z_w)$ and by part-level activities induced by them: $A_p = 1$ if $Z_w = 1$ for any w with $p \in w$, or equivalently,

$$A_p = \max_{w:p \in w} Z_w. \quad (2.1)$$

This conveys the simple assumption that a part is activated if it is in any whole that

is activated. For false-positive and true-positive parameters $\alpha, \gamma \in (0, 1)$, with $\alpha < \gamma$, the model for X entails mutually independent components (conditionally on latent activities), with

$$X_p \sim \text{Bernoulli} \begin{cases} \alpha & \text{if } A_p = 0 \\ \gamma & \text{if } A_p = 1 \end{cases} \quad (2.2)$$

Simply, activated parts (*i.e.*, those with $A_p = 1$) are delivered to the list at a higher rate than inactivated parts. A key feature of the model is that a part (gene) is activated by virtue of any one of its functional *roles*; this implies that a gene may be activated and yet be part of a functional category that is inactivated, which is in contrast to most other gene-set inference methods (*e.g.*, Goeman and Bühlmann, 2007; Barry *et al.*, 2008; Sartor, Leikauf, and Medvedovic, 2009), and which provides for a fully multivariate analysis of the gene list. In Bauer *et al.* (2010) it is further assumed, for the sake of Bayesian analysis, that uncertainty in whole-level activities is represented with a single rate parameter $\pi \in (0, 1)$:

$$Z_w \sim_{i.i.d.} \text{Bernoulli}(\pi). \quad (2.3)$$

Taken together, the model (2.2) and the prior (2.3) determine a joint posterior for Z given X . The R package **MGSA** (model-based gene set analysis) reports MCMC-computed marginal posterior probabilities $P(Z_w = 1|X)$, also integrating uncertainty in the system parameters α , γ , and π , and thus provides a useful ranking of the wholes (Bauer *et al.* 2011; R Core Development Team 2011).

In addition to the system incidence matrix I , a useful data structure for computations turns out to be the bipartite graph $\mathcal{G}$, having whole nodes and part nodes, and an edge between w and p if and only if $I_{p,w} = 1$ (i.e., iff $p \in w$).

The activation hypothesis

As defined above, the role model allows that a whole can be inactive while all of its parts are active. This can happen because of overlap among the wholes. Specifically, if w is contained in the union of other wholes $\{w'\}$ then all $Z_{w'} = 1$ will force $A_p = 1$ for all $p \in w$, regardless of the value of Z_w . This rather odd situation calls into question the meaning of *active* and what we might realistically expect can be inferred from data. Indeed the issue is related to identifiability of the activity vector Z , since it shows that distinct Z vectors may produce the same part-level activity vector $A = (A_p)$. (In the case above, switching Z_w from 0 to 1 does not change A .) The mapping $Z \rightarrow A$ given by (2.1) is not necessarily invertible, depending on the system as defined in I . Lack of identifiability would not necessarily create difficulty in a Bayesian analysis, however in the present case we are specifically interested in inferring the activity states of the gene sets and prioritizing these sets, and so it stands to reason that we ought to confer a real, if still only model-based, meaning on the activities.

When *activity* is defined more fully, there is a simple solution to the problem. The *activation hypothesis* asserts that a set of parts is active if and only if all parts in the set are active. It was shown previously (Newton *et al.*, 2012):

Theorem 2.2.1. *Under the activation hypothesis (AH), the mapping $Z \rightarrow A$ defined*

by

$$A_p = \max_{w:p \in w} Z_w$$

is invertible, with inverse $A \longrightarrow Z$

$$Z_w = \min_{p:p \in w} A_p.$$

The inverse mapping is simply that a whole is inactive if and only if any of its parts is inactive. So the odd case at the beginning of the section cannot occur under AH; if all parts are active, then $Z_w = 1$ must hold. Further, with parameters α and γ fixed, the Z vector is identifiable under AH, since different Z vectors necessarily give different probability distributions to data X .

The first contribution of the present work is to show that the activation hypothesis is equivalent to a set of linear inequality constraints on the activity variables. The finding is useful for posterior inference computations. We prove that:

Theorem 2.2.2. *AH holds if and only if all of the following hold:*

1. $Z_w \leq A_p$ for all p, w with $p \in w$
2. $A_p \leq \sum_{w:p \in w} Z_w$ for all p
3. $\sum_{p:p \in w} (Z_w - 2A_p + 2) \geq 1$ for all w

Proof. Relative to all the sets and parts in the system I , we say AH holds if and only if $A_p = \max_{w:p \in w} Z_w$ for all p and $Z_w = \min_{p:p \in w} A_p$ for all w . Recall that

all A_p and Z_w are binary, in $\{0, 1\}$. The first condition $A_p = \max_{w:p \in w} Z_w$ implies $A_p \geq Z_w$ for all w with $p \in w$; that A_p achieves the max of the Z_w 's has to account for the possibility that $A_p = 1$ when all $Z_w = 0$, but this is covered by having $A_p \leq \sum_{w:p \in w} Z_w$. Thus the condition $A_p = \max_{w:p \in w} Z_w$ is equivalent to the first two constraints in Theorem 2.2.2.

To address the second condition, that $Z_w = \min_{p:p \in w} A_p$ for all w , define a new variable

$$T_w = 1 + \sum_{p:p \in w} (A_p - 1),$$

and notice that $T_w = 1$ if and only if $A_p = 1$ for all $p \in w$, otherwise $T_w \leq 0$. Observe that the second condition is equivalent to

$$\sum_{p:p \in w} (Z_w - A_p + 1) - T_w \geq 0 \tag{2.4}$$

since if all $A_p = 1$, for $p \in w$, then $T_w = 1$, and Z_w must equal 1 to satisfy (2.4); otherwise, if at least one of the A_p 's equals 0, then $T_w \leq 0$, and noting that the summation in (2.4) is positive confirms the claim. Next, replacing T_w in (2.4) with its definition, we obtain the third stated inequality

$$\sum_{p:p \in w} (Z_w - 2A_p + 2) \geq 1.$$

□

Evidently, the i.i.d. Bernoulli prior (2.3) does not respect AH in the sense that

vectors Z which violate AH have positive prior probability. In simple systems such violation may be innocuous. We provide evidence that in the complex systems such as GO, this violation creates a substantial loss of statistical efficiency. We note first that alternative prior specifications are available that respect AH. A simple one is to condition prior (2.3) on the AH event, namely

$$P(Z = z) = \left(\frac{1}{c}\right) \pi^{\sum_w z_w} (1 - \pi)^{\sum_w (1 - z_w)} \quad \text{if } z \text{ satisfies AH,} \quad (2.5)$$

otherwise $P(Z = z) = 0$, where c is the probability, in prior (2.3), that Z satisfies AH, and z is a vector of binaries representing a possible realization of Z . In other words, with subscript ‘1’ for the i.i.d. prior (2.3) and ‘2’ for prior (2.5), we have: $P_2(Z = z) = P_1(Z = z | \text{AH})$. Upon conditioning, the (Z_w) are not necessarily either mutually independent or identically distributed.

Some statistical properties

The role of the prior distribution in Bayesian analysis has surely been the subject of considerable debate. On one hand it helps by regularizing inference, especially in high dimensions. On the other hand, data need to work against it to produce inferences that trade off empirical characteristics with prior assumptions. A fact of relevance to the present problem is that gene-list data must work against either prior (2.3) or (2.5) to deliver an inferred list of activated gene sets. For two Bayesian analysts, one using prior (2.3) and the other using prior (2.5), the true state is ascribed different prior mass. The ratio of these masses, ρ , represents the extra effort needed to be done by

the data to overcome prior (2.3) compared to prior (2.5):

$$\rho = \frac{P_2(Z = z_{true})}{P_1(Z = z_{true})} = \frac{P_1(Z = z_{true}|\text{AH})}{P_1(Z = z_{true})} = \frac{1}{P_1(\text{AH})} \geq 1. \quad (2.6)$$

Here we have used the particular structure of prior (2.5) and also the assumption that z_{true} satisfies AH. If z_{true} did not satisfy AH, the target of inference it would be beyond the realm of any gene-level data set to estimate, owing to lack of identifiability. Indeed, it is difficult to see what meaning could be ascribed to z_{true} in that case. The observation to be gained from (2.6) is that the probability of AH under the i.i.d. prior affects the efficiency of inference. In systems where that probability is very small, there is reason to believe that improved inferences are possible. As to the precise effect of ignoring AH, that depends on the particular system I , the true activation state, and the system parameters α and γ . What our initial investigation finds is that a truly activated whole w may tend to have $P_1(Z_w = 1|X)$ smaller than $P_2(Z_w = 1|X)$, and if so the P_1 inference is too conservative.

Whether or not AH holds for a given state Z may be assessed by calculating the part-level activities A and then checking Theorem 2.2.2. Alternatively, we consider whole-level *violation* variables (V_w). These Bernoulli trials are defined:

$$V_w = \begin{cases} 1 & \text{if } Z_w = 0 \text{ and if for all } p \in w \text{ there exists } w' \\ & \text{with } p \in w' \text{ and } Z_{w'} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

The probability, under P_1 , that Z satisfies AH is equivalent to the probability of no

violations, that is:

$$P_1(AH) = P_1(V_w = 0, \forall w), \quad (2.8)$$

and so AH probability might be approachable by considering the violation variables. Except in stylized examples we do not expect these variables to be mutually independent; indeed they may have a complicated dependence induced by overlaps of the wholes, and hence direct calculation of $P_1(AH)$ is intractable. However, the expectations of V_w are readily computable for a given system, either by Monte Carlo or by a more sophisticated algorithm:

An algorithm to evaluate the violation probability A violation of AH occurs for whole w if (and only if) $Z_w = 0$ and yet $A_p = 1$ for all $p \in w$. Immediately, we see $P_1(V_w = 1) = (1 - \pi)P(A_p = 1 \forall p \in w | Z_w = 0)$. The second factor is difficult to compute by full enumeration since all configurations of overlapping sets must be considered, and this is exponential in the number of overlapping sets. One simple fact is that if $A_p = 1$, then at least one element in $\{Z_{w'} : w' \neq w, p \in w'\}$ should be equal to 1. Based on this observation, one idea is to find disjoint sets of $Z_{w'}$'s which will result in $\prod_{p \in w} A_p = 1$ (this probability is easy to be computed using their independence), and then consider all the cases. We derived an algorithm and coded a recursive function to implement it. In the function, we do something like a Depth First Search.

- Activate the whole containing the most parts (parts contained in this whole are thus turned on), and see which parts are left to be activated; if those remaining

parts are contained by disjoint sets of wholes, compute the probability directly and go for the next whole containing the second most parts, and so on; if not, invoke the function recursively.

- Overlook wholes already activated and check if there are duplicate wholes which contain the same parts; if true (this happens a lot in GO), retain only one of them and record the number of duplicates.
- If there exists some part left to be activated which is not contained any left whole, stop the execution of the function.

One advantage of this algorithm is that we check if there are duplicate wholes and combine them at each recursion, which makes use of the local structure of the incidence matrix to the fullest extent and can save a lot on running time. In pseudo-code the algorithm is:

```
GetValue(p, wholes, parts)

## p is the Bernoulli parameter, wholes and parts are
## two lists storing the containing information
if (there are duplicate wholes which contain the same parts)
    only retain one of them and record the number of duplicates;
endif

sort all the wholes according to the decreasing order of the number of parts
they contain;

for (i in 1:n)## length(wholes) = n
    remove wholes[0: i-1];
    turn on wholes[i] and check which parts are left to be activated;
```

```

    if (some part left to be activated is not included by any whole in wholes[i:n])
        break;
    endif

    if (sets of wholes containing these parts are disjoint)
        compute the probability, stored by prob[i];
    else (prob[i] = Getvalue(p, trimmed wholes, trimmed parts))
        ## trimmed wholes for ones containing parts left to be activated (parts in
        ## wholes[i] will be removed from the containing information) and
        ## trimmed parts for ones left to be activated
    endif
endfor
return sum(prob);

```

Considering the Chen-Stein result for Poisson approximations, we conjecture that $-\log P_1(AH)$ is approximately equal to $E_1(\sum_w V_w)$, though we have not been able to guarantee an error on this approximation (*c.f.* Arratia, Goldstein, and Gordon, 1990).

Figure 2.1 charts the expected value $E_1(\sum_w V_w)$ over four recent versions of Gene Ontology, for $\pi = 1/100$. For concreteness it focuses on GO terms holding between 5 and 20 genes (for which an exact calculation of the expectation is feasible), though the key finding is not sensitive to that restriction, as evidenced by Monte Carlo computations (not shown). As one might expect by the increasing density and complexity of GO, the expected number of AH violations increases. This may very well reflect the fact that $P_1(AH)$ is decreasing over time, which indicates to us that

ignoring AH is becoming an ever greater problem for gene-set analysis.

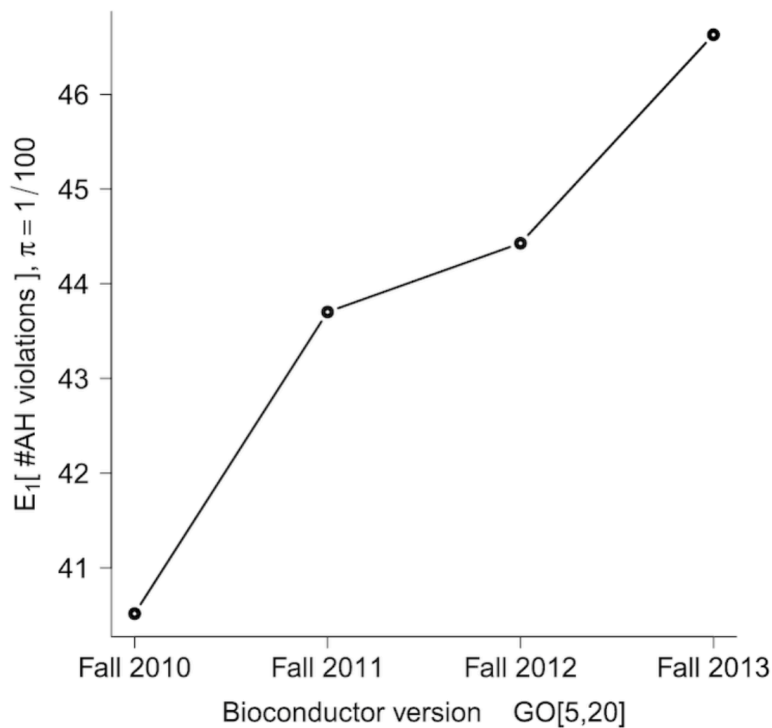


Figure 2.1: Expected number of sets that violate the activation hypothesis (AH) for three recent versions of Gene Ontology (GO), considering sets holding between 5 and 20 genes, taken on the *i.i.d.* Bernoulli prior. Calculations are done at $\pi = 1/100$. Respectively, these systems contain 3591, 4096, 4449 and 4772 gene sets, and correspond to versions of `org.Hs.eg.db`.

In terms of modeling assumptions there is no additional cost to accounting for AH in the Bayesian analysis; the cost is purely computational, since inference must now deal with the constraints imposed by AH on the space of latent activities. The next sections describe two computational advances that address the problem.

2.3 Maximum a posterior (MAP) estimate via integer linear programming (ILP)

Decoding a discrete signal is frequently accomplished by algorithms that compute the parameter state having the highest posterior mass: the maximum a posteriori (MAP) estimate. Although limited as a posterior summary when noise levels are high, the MAP estimate contains useful multivariate information (*e.g.*, Carvalho and Lawrence, 2009). Our representation of model-based gene set analysis reveals that under model (2.2) and prior (2.5), the log posterior is linear in the joint collection of whole and part activity variables. Up to an additive constant, this log posterior is:

$$\begin{aligned}
 l(Z, A) = & \sum_w \{Z_w \log(\pi) + (1 - Z_w) \log(1 - \pi)\} + \\
 & \sum_p \{A_p [x_p \log(\gamma) + (1 - x_p) \log(1 - \gamma)] + \\
 & (1 - A_p) [x_p \log(\alpha) + (1 - x_p) \log(1 - \alpha)]\},
 \end{aligned} \tag{2.9}$$

where x_p is the realized value of the gene-list indicator X_p , and α , γ , and π are system parameters, which are considered fixed in the present calculation. Considering Theorem 2.2.2, finding the MAP estimate $(\hat{Z}, \hat{A})$ amounts to maximizing a linear function in discrete variables subject to linear inequality constraints. As such, it fits naturally into the domain of *integer linear programming* (ILP), an active subfield of optimization. Our computations take advantage of ILP software available in the GNU Linear Programming Kit through its interface with R.¹ We employed a series

¹See www.gnu.org/software/glpk and cran.R-project.org/package=Rglpk

of basic code checks to assure our implementation worked in: (1) simple examples where the MAP estimate is computable by other means; and (2) limiting situations where X_p was Binomial having a high sample size, and thus where the MAP estimate must converge to the true activity state.

The reconstruction $\hat{Z}$ obtained through this optimization holds an estimate of the activated and inactivated gene sets. We refer to the overall method as the *multi-functional analyzer* (MFA), and specifically MFA-ILP to refer to the posterior mode computed by ILP. We note that by invertibility of the mapping $Z \rightarrow A$ under AH, the log-posterior l could be expressed either as a function of Z only or as a function of A only; however in neither reduced case would l be linear in the input variables. Moreover, in neither reduced case could the constraints be expressed as linear inequality constraints. By expanding the domain, we formulate the constrained optimization as an integer linear program.

Large systems strain unaided ILP computation, but the special structure of the gene-set problem allows for several refinements.

Shrinking I : Observe that the objective function in (3.1) may be expressed:

$$l(Z, A) = c_1 \sum_w Z_w + c_2 \sum_{p \in P^-} A_p + c_3 \sum_{p \in P^+} A_p$$

where

$$\begin{aligned} c_1 &= \log \pi - \log(1 - \pi) \\ c_2 &= \log(1 - \gamma) - \log(1 - \alpha) \\ c_3 &= \log \gamma - \log \alpha \end{aligned}$$

and where P^- and P^+ denote the observed inactivated and activated parts, respectively. That is, $p \in P^-$ if $x_p = 0$ and $p \in P^+$ if $x_p = 1$. By assumption (2.2), $\alpha < \gamma$ and so $c_2 < 0$ and $c_3 > 0$. If we further insist that $\pi < 1/2$, then $c_1 < 0$ also. In some cases we can know which $\hat{Z}_w$ and $\hat{A}_p$ must equal 0 in the optimal solution, and if so we can remove these variables from the system prior to implementing ILP. For each whole w denote $P_w^+ = w \cap P^+$ and similarly $P_w^- = w \cap P^-$, and define $W^* = \left\{ w : c_1 + c_3 \sum_{p \in P_w^+} 1 < 0 \right\}$. Clearly those wholes containing no reported parts are in W^* , but there may be others. We prove that if W^* is not empty then we may be able to shrink the system prior to solving the constrained optimization problem via ILP.

Theorem 2.3.1. *Suppose $\pi < 1/2$ and let w_0 denote an element of W^* . If there exists $p_0 \in w_0$ such that $\{w : p_0 \in w\} \subset W^*$, then $\hat{Z}_{w_0} = \hat{A}_{p_0} = 0$.*

Proof. Compared to $Z_{w_0} = 0$, the possible maximal value added to the objective function by letting Z_{w_0} be 1 is $c_1 + c_3 \sum_{p \in P_{w_0}^+} 1$ (considering parts in $P_{w_0}^-$ may already be activated or w_0 has no parts in P^- , the best case), however, which is negative since $w_0 \in W^*$. So $Z_{w_0} = 0$ is preferred towards maximizing the objective function. Next

we need to prove that letting $Z_{w_0} = 0$ and $A_{p_0} = 0$ will not violate the inequalities in Theorem 2.2.

Denote by W_0 and P_0 the sets of w_0 and p_0 satisfying the state in the Theorem, respectively. We claim that for each $p_0 \in P_0$, $\{w : I_{p_0,w} = 1\} \subset W_0$. If not, then there exists $w^* \in W^* \setminus W_0$ such that $I_{p_0,w^*} = 1$, so w^* will be in W_0 . This is a contradiction. Thus $A_{p_0} = \max_{\{w: I_{p_0,w}=1\}} Z_w = 0$, so the first two AH inequalities are satisfied. It is readily verified that the third inequality is also satisfied. $\square$

Letting W_0 and P_0 denote wholes and parts for which the optimal solution is known (in advance of computation), we may remove these from the incidence matrix I , effectively shrinking it. The amount of shrinkage may be dramatic, but it depends on the observed data x , the system I , and system parameters α , γ , and π . When α is small and γ is large the effects may be minimal.

A sequential approach: In the unlikely event that the system matrix I is separable into blocks of wholes that do not overlap between blocks, then ILP may be applied separately to these distinct blocks in order to identify the MAP activities. We do not expect this separability in GO or related systems, but we can take advantage of size variation of the wholes and work sequentially from small ones to larger ones. As an example, let S_{10} denote the sets containing no more than $n.up = 10$ genes. In order to obtain the optimal solution for the full problem, we start from the sub-matrix $I.10$ obtained by extracting these sets from I . Suppose Z_{10}^* is the MAP solution based on the data for $I.10$, and use notation S_{10}^* to denote the active sets in S_{10} as inferred by Z_{10}^* . We aim to find the optimal solution Z_{11}^* for $I.11$ using what has

already been computed in the smaller system. Denote the newly added sets in $I.11$ by S_{10}^{11} (*i.e.*, the sets containing exactly 11 genes). We just need to consider the sets with the possibility being active in the optimal solution on $I.11$. First of all, S_{10}^* and S_{10}^{11} should be included, in the case we have no any other prior knowledge about Z_{11}^* . Second, by the 3rd AH inequality (Theorem 2.2.2), any set in $S_{10} \setminus S_{10}^*$ which is a subset of some set in S_{10}^{11} , denoted by D , should also be included. But these sets already considered are not enough. Actually, for each set w_1 in S_{10}^{11} , we need to check whether there exist some set w_2 in $S_{10} \setminus (S_{10}^* \cup D)$ satisfying

$$c_1 + c_2 \sum_{p \in P^- \cap P_{w_1}^{w_2}} A_p + c_3 \sum_{p \in P^+ \cap P_{w_1}^{w_2}} A_p > 0, \quad (2.10)$$

where $P_{w_1}^{w_2}$ denote the set of genes contained by w_2 and not by w_1 . We do this since each set in S_{10}^{11} may be active in the optimal solution Z_{11}^* , and we need to check whether some sets in S_{10} should be activated towards maximizing the objective function. We denote the sets in $S_{10} \setminus (S_{10}^* \cup D)$ satisfying the condition (2.10) by E . Finally, by the 3rd AH inequality (Theorem 2.2.2), any set in $S_{10} \setminus (S_{10}^* \cup D \cup E)$ which is a subset of some set in E , denoted by F , should also be included. Thus, we need to run the ILP on the incidence matrix only for $S_{10}^* \cup S_{10}^{11} \cup D \cup E \cup F$, instead of $I.11$. Hence, we obtain a sequential approach to solve the full ILP problem from a sequence of smaller problems. Examples show this is feasible in GO for sub-systems holding sets of up to 50 genes, without excessive computational burden.

2.4 Posterior sampling via Monte Carlo Markov Chain

To obtain a sample from the posterior distribution defined by prior (2.5) and model (2.2) in which the whole activity variables $Z = (Z_w)$ have positive probability only when Z satisfies AH, we designed a Markov chain to run according to a penalized posterior within the entire, unconstrained space. Up to a constant, this penalized log posterior is:

$$\tilde{l}(Z) = l(Z, A) - \lambda \sum_w V_w \quad (2.11)$$

where $l(Z, A)$ is defined in (3.1), V_w is the violation indicator (2.7), and $\lambda \geq 0$ is a tuning parameter. The desired sample is obtained by discarding any sampled states that do not satisfy AH. Note that there are no violations ($\sum_w V_w = 0$) for Z that satisfy AH, so that $\tilde{l}(Z) = l(Z, A)$ in this case and the conditional log posterior distribution under $\tilde{l}(Z)$ restricted to AH is identical to the target log posterior distribution. Increasing the tuning parameter λ increases the probability of AH in the larger state space, which is essential for efficient sampling when this probability is small. As the Markov chain samples from the larger space of all possible binary vectors Z , here the part activities A are determined by Z , the mapping from Z to A need not be invertible as in Theorem 2.2.1, and we can consider $\tilde{l}$ as a function only of Z .

It is helpful to visualize the Markov chain as operating by changing colors on the

node-colored bipartite graph $\mathcal{G}$ having whole nodes and part nodes, with an edge between a whole node w and part node p if and only if $p \in w$, and where the coloring of the whole nodes $\{w\}$ and part nodes $\{p\}$ match the activities Z and A , respectively. It is useful in assessing the state of the Markov chain to associate with each node a count $n(\cdot)$ of its active connected neighbors in $\mathcal{G}$. $A_p = 1$ if and only if $n(p) > 0$ and $V_w = 1$ if and only if $Z_w = 0$ and $n(w) = \deg(w)$, the number of part nodes $p \in w$.

The Markov chain proceeds by selecting at random a whole node w and proposing a color swap (a change in the status of the activity variable, $Z_w^* = 1 - Z_w$) for this node. This proposed change can, but need not, affect the activities of parts contained in this whole. When $Z_w^* = 1$, the active neighbor counts $n(p)$ increase by 1 for each $p \in w$. If A_p changes from 0 to 1, then each node w' that contains p (including w) gains an additional active neighbor and $n(w')$ increases by 1. This increase could cause a violation if p were the only remaining inactive neighbor of an inactive w' , causing $V_{w'}$ to change from 0 to 1. If node w were in violation before this proposal, activating it would eliminate the violation. Similarly, when $Z_w^* = 0$, the active neighbor counts $n(p)$ decrease by 1 for each $p \in w$. If this decrease is from 1 to 0, then the activity A_p changes from 1 to 0 as well and all of the whole nodes w' connected to p would lose an active neighbor, $n(w')$ decreasing by 1. If whole node w' had been in violation, this change would eliminate the violation with $V_{w'}$ changing from 1 to 0.

Careful accounting of the changes to a few key counts allows for quick calculation of the change in $\tilde{l}(Z^*)$ and subsequent acceptance or rejection of the proposal by Metropolis-Hastings. The log posterior $\tilde{l}(Z^*)$ is a function of α , γ , π , the penalty λ and

the counts of the numbers of active and inactive whole nodes ($\sum_w Z_w$ and $\sum_w (1 - Z_w)$, respectively), the number of whole nodes in violation ($\sum V_w$), the numbers of active part nodes with realized values 1 and 0 ($\sum_p A_p x_p$ and $\sum_p A_p (1 - x_p)$, respectively), and the numbers of inactive part nodes with realized values 1 and 0 ($\sum_p (1 - A_p) x_p$ and $\sum_p (1 - A_p) (1 - x_p)$, respectively).

Random selection of w for the proposed change may be uniform. Some increase in MCMC mixing efficiency is obtainable by using a nonuniform proposal distribution, for example, by selecting whole nodes with smaller set size and with larger proportions of parts with realized values equal to 1 with greater probability.

We note that penalizing the log posterior within the unconstrained space leads to a rapidly mixing sampler, where our previous attempts to construct move types within the constrained space were less successful.

In subsequent calculations we deploy both MAP estimation (MFA-ILP) and MCMC sampling on each data set in order to infer wholes that are probably activated. MFA-ILP is our primary tool, as it gives a summary functional decoding of the gene list. Posterior probabilities from the MCMC computation provide a measure of confidence in the inferred sets.

2.5 Simulations

Numerical experiments

In each experiment reported below we represented a system with a part-by-whole incidence matrix I ; we fixed the false positive rate $\alpha = 1/10$ and the true positive

rate $\gamma = 9/10$. We simulated 100 gene-lists X from model (2.2), each time using a simulated activity vector Z . For method comparison, we applied: (1) the commonly used Fisher’s exact test for enrichment of each gene set in the data X , (2) MGSA (version 1.7.0), and (3) MFA-ILP. We allowed both model-based methods to know the system parameter settings. To evaluate performance we calculated specificity, sensitivity, and precision of the estimated activity vector $\hat{Z}$ for the true activity vector Z by averaging over the 100 replicates.

Experiment 1: Low overlap. Initially, I had size 300 genes (parts) by 100 gene sets (wholes). We randomly picked 5 and 10 parts for each whole in columns 1-50 and 51-100, respectively. Then we removed parts not contained by any whole, leaving a 296×100 incidence matrix. We sampled Z from prior (2.3) and then projected it onto AH by constructing $A_p = \max_{w:p \in w} Z_w$ and then updating $Z_w = \min_{p:p \in w} A_p$. All methods exhibit similar operating characteristics in this case (Table 2.1).

Experiment 2: Higher overlap and parent-child structure. Initially, I had size 300 parts by 105 wholes. From column 1 to column 20, each column has 20 parts, of which 15 parts are in common with each other and 5 parts are randomly selected from the other parts; column 21 has 10 parts which are randomly picked from the 15 common parts shared by columns 1-20. Thus, columns 1-20 have a lot of overlaps and column 21 is a child of columns 1-20. Similarly we built columns 22-42, 43-63, 64-84 and 85-105. The common 15 parts in each column combination are all different. Then parts not contained by any whole were removed, which resulted in a 265×105 incidence matrix. We activated wholes by sampling one whole from columns 1-20,

22-41, 43-62, 64-83 and 85-104 as activated, respectively, and projected onto AH as above.

Table 2.2 exhibits properties of three methods in relatively complicated system just defined. The univariate Fisher test tends to select the wholes with a high correlation (overlap) with the truly activated wholes, which results in high sensitivity but low specificity (or precision). The extra activation calls correspond to spurious associations that the multivariate, model-based approaches are able to recognize. The MGSA method often fails to discover truly activated wholes, which corresponds to a reduced sensitivity. The small *child* wholes tend to be missed by MGSA in this case. The proposed MFA-ILP method is right on target.

Table 2.1: Simulation comparison of three gene-set analysis methods, a case of low overlap among gene sets: tabulated are mean values from 100 simulated data sets. On average 7.3 truly activated sets occur.

Method	Predicted # active	Sensitivity	Specificity	Precision
MFA-ILP	7.4	0.963	0.997	0.958
Fisher (cut-off=0.05)	5.9	0.790	0.998	0.966
Fisher (cut-off=0.1)	6.8	0.873	0.996	0.948
MGSA (cut-off=0.5)	7.2	0.954	0.998	0.968

Table 2.2: Simulation comparison of three gene-set analysis methods, a case of higher overlap among gene sets: tabulated are mean values from 100 simulated data sets. On average there are 10.1 truly activated sets in this case.

Method	Predicted # active	Sensitivity	Specificity	Precision
MFA-ILP	10.2	0.975	0.997	0.993
Fisher (cut-off=0.05)	104.2	0.996	0.008	0.096
Fisher (cut-off=0.1)	104.8	0.996	0.002	0.096
MGSA (cut-off=0.5)	5.5	0.490	0.995	0.920

Operating characteristics

To assess the performance of MFA-MCMC, we simulated gene-list data according to the role model in the *Drosophila Melanogaster* genome, following the scheme presented in Bauer, Gagneur and Robinson (2010). Briefly, we used 3275 GO terms annotating between 5 and 50 fly genes, according to version 2.14.0 of Bioconductor package `org.Dm.eg.db`. In each simulation run, a number of GO terms were activated and then a gene list was constructed from independent Bernoulli trials depending on the activation states and settings of false-positive and false-negative error rates. Figure 2.2 shows receiver-operating (ROC) curves and precision-recall curves for two parameter settings, based on 100 simulated gene lists in each setting. Selection to the reported set list is based on thresholding the marginal posterior probability (MGSA, MFA-MCMC) or the p-value (Fisher). Evidently, MGSA and MFA-MCMC are accurate and show similar behavior when error rates are low, though MFA-MCMC shows improved precision and sensitivity in more difficult settings.

In subsequent calculations we deploy both MAP estimation (MFA-ILP) and MCMC sampling on each data set in order to infer wholes that are probably activated. For MCMC, we use 10^7 sweeps, burn-in of 10^6 , and $\lambda = 5$, which causes about one third of the states to satisfy AH. MFA-ILP gives a summary functional decoding of the gene list. Posterior probabilities from the MCMC computation provide a measure of confidence in the inferred sets and also highlight notable non-MAP sets. Fisher’s test is the default univariate method for gene-list data; we include it for comparison, even though the hypotheses it tests are different from the activation states assessed by MFA and MGSA.

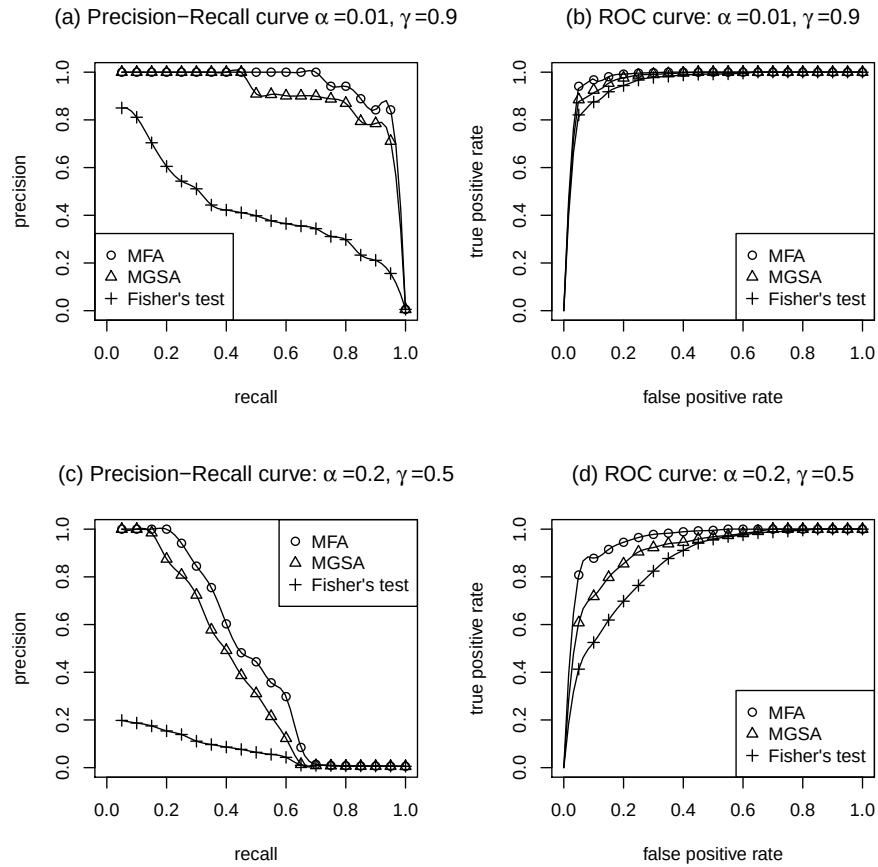


Figure 2.2: Operating characteristics of MFA-MCMC, MGSA, and Fisher's test based on simulating the role model in the *D. melanogaster* genome.

2.6 Case studies

Genes implicated in Type-2 diabetes (T2D)

From a large-scale genome-wide association study (GWAS) involving more than 34,000 cases and 114,000 control subjects, 77 human genes have been implicated as affecting T2D disease susceptibility (Morris *et al.* 2012, Supplementary Table 15, primary list). To assess the functional content of this gene list, we applied MFA, MGSA, and simple enrichment via Fisher’s exact test, all in the context of 6037 gene ontology terms, each annotating between 5 and 50 genes.² Here and in other examples we took advantage of available information on likely false positive (α) and false negative ($1 - \gamma$) error rates at the gene level. Using the fitted mixture model from Morris *et al.*, we estimated $\alpha = 0.00019$ and $\gamma = 0.02279$ for this large-scale GWAS (see Section 2.7).

Figure 2.3 summarizes the application of MFA, MGSA, and Fisher’s test to this example. Table 2.3 reports those gene sets inferred by MFA-ILP to be activated in T2D. Tables 2.4-2.7 provide further information for comparison of MFA with MGSA and Fisher’s test. The example illustrates features we see repeatedly with these methods. Sets identified by Fisher’s test tend to overlap substantially, reflecting the univariate nature of the approach; both MGSA and MFA-ILP alleviate this redundancy problem, but MGSA finds fewer sets than MFA-ILP. As expected, the 11 sets inferred by the ILP algorithm (i.e., the MAP estimate of the activated sets) each has high marginal posterior probability of activation (P.MFA). Furthermore,

²These 6037 terms annotate a total of 10,626 genes; among the 77 T2D-associated genes, 58 are in this *moderately annotated* class.

MFA-ILP is able to explain more of the gene level findings than the other methods, as indicated by the number of genes that are both in the reported gene list (T2D) and are in at least one of the gene sets inferred to be activated (*coverage*). It does this without increasing the mis-coverage, which is the number of non-T2D genes within the inferred active sets. Figure 2.3 summarizes the sets found by these methods and reports these coverages.

Table 2.3: MFA results in Type-2 Diabetes (T2D) example: 11 GO terms are inferred to be active using the ILP algorithm to compute the MAP estimate (rows). Basic statistics on these terms are provided (# T2D-associated genes/set size). The next two columns give the MCMC-computed marginal posterior activation probabilities for these terms, both using MGSA and MFA, the constrained alternative. The final column holds the Benjamini-Hochberg adjusted Fisher-test p-value. All calculations start with 6037 GO terms (those annotating between 5 and 50 human genes) that together annotate 10626 human genes. Of the 77 total T2D genes, 58 have at least one annotation to these 6037 GO terms. The inferred gene sets cover 26 of these 58 T2D genes.

Gene set (GO term)	Statistics	P.MFA	P.MGSA	FisherQ
RNA polymerase II core promoter ...	3/45	0.517	0.028	0.161
positive regulation of insulin secretion	4/41	0.964	0.372	0.016
positive regulation of peptidyl-serine ...	2/35	0.537	0.096	0.756
negative regulation of insulin secretion	4/23	0.996	0.201	0.003
ER overload response	2/9	0.398	0.159	0.102
positive regulation of insulin secretion ...	0/9	0.964	0.002	1.000
hepatocyte differentiation	2/9	0.316	0.016	0.102
endodermal cell fate specification	2/8	0.596	0.036	0.091
exocrine pancreas development	3/8	0.946	0.600	0.003
negative regulation of protein ...	2/5	0.420	0.101	0.051
lamin filament	2/5	0.790	0.400	0.051

An interesting set in this case is the GO term *glucose import* (GO:0046323), for which the proportion 5/41 of observed T2D genes is very high (tiny Fisher p-value), but there is very small posterior activation probability according to MFA. That's because the 5 genes are explained more easily as parts of three other terms in the

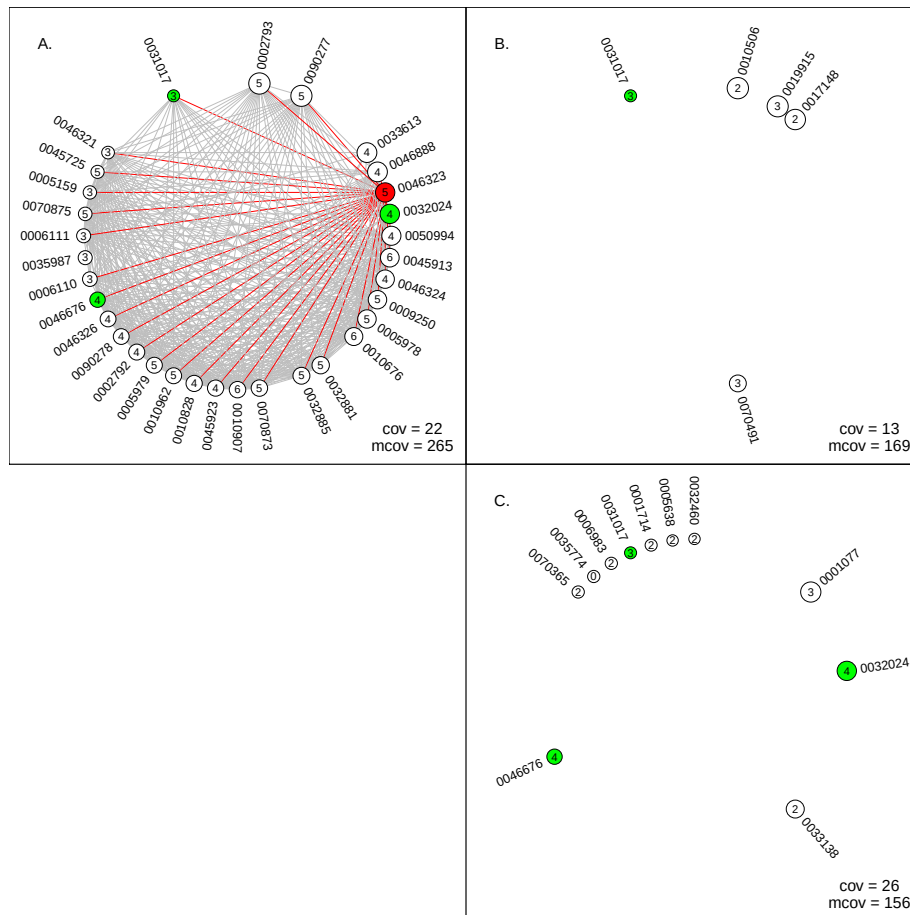


Figure 2.3: GO terms inferred by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) to be activated to explain the type-2 diabetes associated genes (58 of the 77 are annotated to the 6037 GO terms used). Panels are linked by the location of sets; three sets (green) were identified by multiple methods. GO ID is on the outer rim, and sets are ordered by size as in Tables 2.3, 2.4, 2.5 and 2.6, from largest (noon) and decreasing clockwise; numbers inside circles indicate the number of T2D genes annotated to that set; a line connects two sets if their intersection contains at least one T2D-associated gene. The total number of T2D genes explained by the identified sets (coverage), lower right, as a subset of the 58 starting genes. Similarly the mis-coverage (mcov) counts the number of genes within the inferred active sets that are not on the observed gene list. GO:0046323 (glucose import) and its relations are highlighted (red) in panel A.

Table 2.4: T2D: MFA-ILP results in Type-2 Diabetes example. Essentially the same as Table 2.3, but with GO ID's included; coverage 26.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0001077	RNA polymerase II core promoter proximal	3/45	0.517	0.028	0.161
GO:0032024	positive regulation of insulin secretion	4/41	0.964	0.372	0.016
GO:0033138	positive regulation of peptidyl-serine p	2/35	0.537	0.096	0.756
GO:0046676	negative regulation of insulin secretion	4/23	0.996	0.201	0.003
GO:0006983	ER overload response	2/9	0.398	0.159	0.102
GO:0035774	positive regulation of insulin secretion	0/9	0.964	0.002	1.000
GO:0070365	hepatocyte differentiation	2/9	0.316	0.016	0.102
GO:0001714	endodermal cell fate specification	2/8	0.596	0.036	0.091
GO:0031017	exocrine pancreas development	3/8	0.946	0.600	0.003
GO:0032460	negative regulation of protein oligomeri	2/5	0.420	0.101	0.051
GO:0005638	lamin filament	2/5	0.79	0.4	0.051

Table 2.5: T2D: MGSA results in Type-2 Diabetes example. Similar to Table 2.3, but the five gene sets shown are those inferred to be activated according to MGSA ($P.MGSA \geq 0.5$); coverage 13.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0010506	regulation of autophagy	2/50	0	0.797	0.856
GO:0019915	lipid storage	3/49	0	0.708	0.192
GO:0017148	negative regulation of translation	2/46	0	0.552	0.856
GO:0070491	repressing transcription factor binding	3/30	0.053	0.823	0.074
GO:0031017	exocrine pancreas development	3/8	0.946	0.600	0.003

MAP estimate that have yet other genes supporting their activation.

A second curious case is GO:0035774, a small term (9 genes) to do with regulation of insulin secretion. None of these genes was reported to be involved in T2D, however the set is fully contained in a parent set which is inferred to be activated by MFA-ILP. As the calculation respects AH, all subsets of activated sets are activated. We note that for the sake of further simplification of output, it is reasonable to suppress any such subsets from primary tabulations:

Algorithm for trimming activated sets When reporting an inferred list of activated sets it is reasonable to remove from primary consideration those sets that are

Table 2.6: T2D: Like Tables 2.3 and 2.4, but selected for small adjusted Fisher p-value; coverage 22.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0002793	positive regulation of peptide secretion	5/50	0	0.279	0.003
GO:0090277	positive regulation of peptide hormone s	5/49	0	0.337	0.003
GO:0046888	negative regulation of hormone secretion	4/43	0	0.149	0.018
GO:0033613	activating transcription factor binding	4/43	0.066	0.305	0.018
GO:0032024	positive regulation of insulin secretion	4/41	0.964	0.372	0.016
GO:0046323	glucose import	5/41	0	0.143	0.001
GO:0050994	regulation of lipid catabolic process	4/40	0	0.173	0.015
GO:0045913	positive regulation of carbohydrate meta	6/38	0	0.028	0
GO:0005978	glycogen biosynthetic process	5/37	0	0.016	0.001
GO:0009250	glucan biosynthetic process	5/37	0	0.02	0.001
GO:0046324	regulation of glucose import	4/37	0	0.017	0.012
GO:0010676	positive regulation of cellular carbohyd	6/35	0	0.027	0
GO:0032881	regulation of polysaccharide metabolic p	5/32	0	0.025	0
GO:0032885	regulation of polysaccharide biosynthesi	5/30	0	0.02	0
GO:0010907	positive regulation of glucose metabolic	6/28	0	0.02	0
GO:0070873	regulation of glycogen metabolic process	5/28	0	0.022	0
GO:0010828	positive regulation of glucose transport	4/27	0.002	0.019	0.004
GO:0045923	positive regulation of fatty acid metabo	4/27	0	0.013	0.004
GO:0002792	negative regulation of peptide secretion	4/26	0	0.171	0.003
GO:0005979	regulation of glycogen biosynthetic proc	5/26	0	0.022	0
GO:0010962	regulation of glucan biosynthetic proces	5/26	0	0.022	0
GO:0046326	positive regulation of glucose import	4/25	0.038	0.025	0.003
GO:0090278	negative regulation of peptide hormone s	4/25	0.004	0.208	0.003
GO:0046676	negative regulation of insulin secretion	4/23	0.996	0.201	0.003
GO:0006110	regulation of glycolysis	3/20	0	0.002	0.031
GO:0006111	regulation of gluconeogenesis	3/17	0	0.004	0.019
GO:0035987	endodermal cell differentiation	3/17	0	0.207	0.019
GO:0070875	positive regulation of glycogen metaboli	5/15	0	0.025	0
GO:0005159	insulin-like growth factor receptor bind	3/14	0.01	0.003	0.013
GO:0045725	positive regulation of glycogen biosynth	5/13	0.095	0.026	0
GO:0046321	positive regulation of fatty acid oxidat	3/11	0.001	0.014	0.006
GO:0031017	exocrine pancreas development	3/8	0.946	0.6	0.003

Table 2.7: T2D: GO terms with MFA marginal posterior activation probability exceeding 0.5. All of these sets are in the MAP estimate (Table 2.3).

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0001077	RNA polymerase II core promoter proximal	3/45	0.517	0.028	0.161
GO:0032024	positive regulation of insulin secretion	4/41	0.964	0.372	0.016
GO:0033138	positive regulation of peptidyl-serine p	2/35	0.537	0.096	0.756
GO:0046676	negative regulation of insulin secretion	4/23	0.996	0.201	0.003
GO:0035774	positive regulation of insulin secretion	0/9	0.964	0.002	1
GO:0001714	endodermal cell fate specification	2/8	0.596	0.036	0.091
GO:0031017	exocrine pancreas development	3/8	0.946	0.6	0.003
GO:0005638	lamin filament	2/5	0.79	0.4	0.051

subsets of unions of larger sets in the list. This has a minimal effect in the T2D and RNAi case studies (one set in each GO run), but it's helpful in general to see how many activated sets are effectively covering the gene list.

Pseudo-code

```

order sets inferred to be active, from largest to smallest loop over order
  if set is contained within union of previously retained sets
    then remove
  else retain
return: list of retained sets

```

R-code

```

trimsets <- function(activeILP, xx) {
## activeILP is vector of set id's
## xx is a nset x ngene incidence matrix

```

```

size <- diag(t(xx[,activeILP])%*%xx[,activeILP])
activeILPord <- names(sort(size, decreasing = T))
activeILPsimple <- activeILPord[1]
for (k in 2:length(activeILP))
  if(apply(as.matrix(xx[, activeILPsimple]), 1, max) %*%
      xx[, activeILPord[k]] < size[activeILPord[k]])
    activeILPsimple <- c(activeILPsimple, activeILPord[k])
return(activeILPsimple)
}

```

RNA interference and influenza-virus replication

In a meta-analysis of four genome-wide studies of influenza virus, Hao *et al.* (2013) reported that 984 human genes had been detected by RNA interference as possibly being associated with viral replication. As in the T2D example, we compared MFA with MGSA and Fisher's test on this gene list using 6037 GO terms annotating between 5 and 50 human genes. Among the 984 influenza-involved genes, 683 are annotated to at least one of these terms. To apply the model-based methods we took advantage of external information on the false positive rate α and the true positive rate γ (see Section 2.7). Figure 2.4 illustrates the sets found by MFA-ILP, MGSA and Fisher's test, and Tables 2.8-2.11 contain further details of the comparative analysis. Again, we find that MFA-ILP dominates the other methods in terms of gene coverage, with 245 genes explained as compared to 226 (MGSA) and 90 (Fisher) and with mis-coverages 635 (MFA-ILP), 634 (MGSA), and 206 (Fisher). Furthermore,

MFA-ILP detects more sets than MGSA (50 in the trimmed list compared to 30 by MGSA).

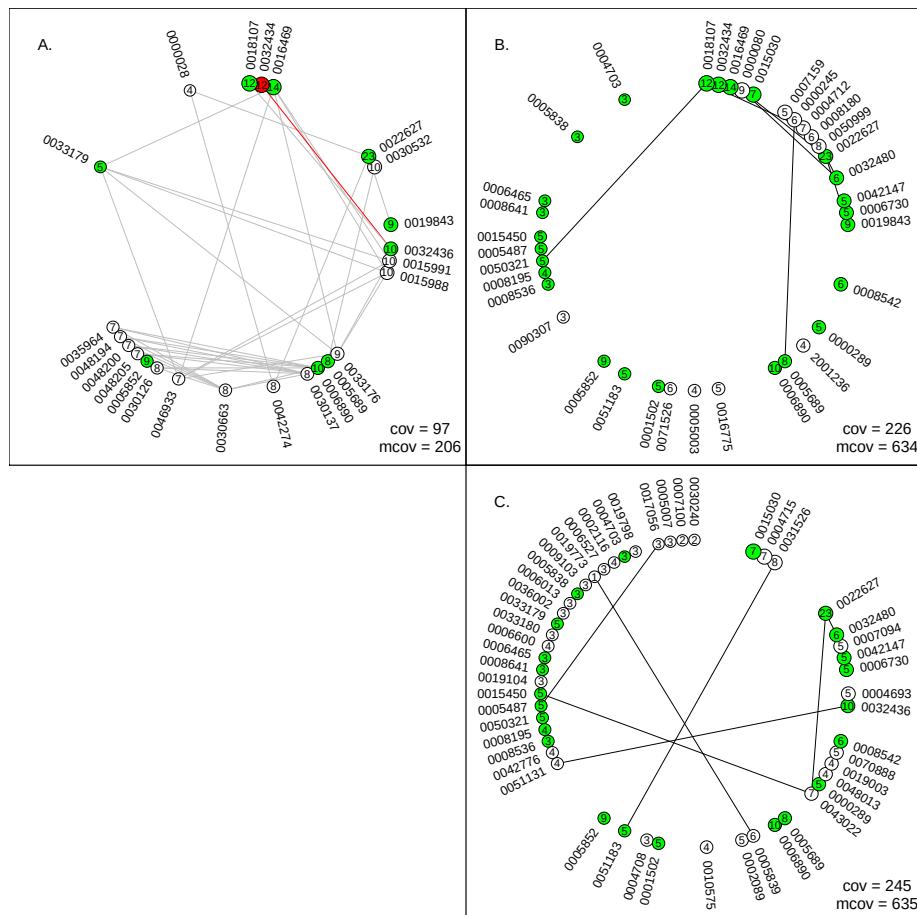


Figure 2.4: Sets (GO terms) identified by three methods (A, Fisher's test; B, MGSA; C, MFA) as activated to explain the influenza-involved human genes. Layout as in Figure 2.3. GO:0032434 and its relations are highlighted (red) in panel A.

To better understand differences between the methods, consider one gene set GO:0032434, *regulation of proteasomal ubiquitin-dependent protein catabolic process*, which annotates 48 human genes, 12 of which were in the 984-list of influenza involved

Table 2.8: Influenza RNAi; GO terms found by MFA-ILP: coverage 245.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0015030	Cajal body	7/46	0.992	0.773	0.846
GO:0004715	non-membrane spanning protein tyrosine k	7/45	0.864	0.024	0.846
GO:0031526	brush border membrane	8/39	0.957	0.484	0.301
GO:0022627	cytosolic small ribosomal subunit	23/35	1	0.994	0
GO:0007094	mitotic cell cycle spindle assembly chec	5/33	0.216	0.097	0.882
GO:0032480	negative regulation of type I interferon	6/33	0.73	0.632	0.775
GO:0006730	one-carbon metabolic process	5/32	0.81	0.722	0.882
GO:0042147	retrograde transport, endosome to Golgi	5/32	0.819	0.76	0.882
GO:0032436	positive regulation of proteasomal ubiqu	10/31	1	0.103	0.007
GO:0004693	cyclin-dependent protein kinase activity	5/31	0.62	0.058	0.882
GO:0008542	visual learning	6/29	0.643	0.704	0.519
GO:0070888	E-box binding	5/27	0.597	0.061	0.846
GO:0043022	ribosome binding	7/26	0.731	0.03	0.152
GO:0000289	nuclear-transcribed mRNA poly(A) tail sh	5/26	0.841	0.915	0.841
GO:0048013	ephrin receptor signaling pathway	4/26	0.694	0.078	0.882
GO:0019003	GDP binding	4/26	0.632	0.029	0.882
GO:0006890	retrograde vesicle-mediated transport, G	10/24	0.971	0.909	0.002
GO:0005689	U12-type spliceosomal complex	8/24	1	0.834	0.026
GO:0005839	proteasome core complex	6/20	0.972	0.391	0.169
GO:0002089	lens morphogenesis in camera-type eye	5/19	0.696	0.48	0.404
GO:0010575	positive regulation vascular endothelial	4/18	0.266	0.39	0.841
GO:0004708	MAP kinase kinase activity	3/16	0.609	0.022	0.882
GO:0001502	cartilage condensation	5/16	0.997	0.971	0.25
GO:0051183	vitamin transporter activity	5/15	0.619	0.674	0.21
GO:0005852	eukaryotic translation initiation factor	9/14	1	0.992	0
GO:0042776	mitochondrial ATP synthesis coupled prot	4/13	0.933	0.007	0.417
GO:0051131	chaperone-mediated protein complex assem	4/13	0.436	0.083	0.417
GO:0008195	phosphatidate phosphatase activity	4/12	0.985	0.954	0.365
GO:0008536	Ran GTPase binding	3/12	0.813	0.65	0.882
GO:0019104	DNA N-glycosylase activity	3/11	0.618	0.054	0.846
GO:0015450	P-P-bond-hydrolysis-driven protein trans	5/11	0.612	0.688	0.065
GO:0005487	nucleocytoplasmic transporter activity	5/11	0.993	0.976	0.065
GO:0050321	tau-protein kinase activity	5/11	0.994	0.901	0.065
GO:0033179	proton-transporting V-type ATPase, V0 do	5/10	0.999	0.005	0.046
GO:0033180	proton-transporting V-type ATPase, V1 do	3/10	0.865	0.004	0.775
GO:0006600	creatine metabolic process	4/10	0.995	0.221	0.239
GO:0006465	signal peptide processing	3/10	0.856	0.569	0.775
GO:0008641	small protein activating enzyme activity	3/10	0.866	0.605	0.775
GO:0005838	proteasome regulatory particle	3/9	0.934	0.636	0.662
GO:0006013	mannose metabolic process	3/9	0.847	0.437	0.662
GO:0036002	pre-mRNA binding	3/9	0.832	0.369	0.662
GO:0006527	arginine catabolic process	3/8	0.514	0.074	0.519
GO:0019773	proteasome core complex, alpha-subunit c	1/8	0.976	0.004	0.882
GO:0009103	lipopolysaccharide biosynthetic process	3/8	0.909	0.135	0.519
GO:0019798	procollagen-proline dioxygenase activity	3/7	0.885	0.403	0.41
GO:0004703	G-protein coupled receptor kinase activi	3/7	0.943	0.892	0.41
GO:0002116	semaphorin receptor complex	4/7	0.545	0.144	0.079
GO:0030240	skeletal muscle thin filament assembly	2/5	0.486	0.265	0.846
GO:0007100	mitotic centrosome separation	2/5	0.509	0.013	0.846
GO:0005007	fibroblast growth factor-activated recep	3/5	0.866	0.42	0.22
GO:0017056	structural constituent of nuclear pore	3/5	0.461	0.381	0.22

Table 2.9: Influenza RNAi; GO terms found by MGSA; coverage 226.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0018107	peptidyl-threonine phosphorylation	12/49	0	0.886	0.019
GO:0032434	regulation of proteasomalubiquitin-depe	12/48	0	0.897	0.017
GO:0000080	G1 phase of mitotic cell cycle	9/47	0.042	0.592	0.301
GO:0016469	proton-transporting two-sector ATPase co	14/47	0	0.963	0.002
GO:0015030	Cajal body	7/46	0.992	0.773	0.846
GO:0000245	spliceosomal complex assembly	6/37	0	0.539	0.846
GO:0007159	leukocyte cell-cell adhesion	5/37	0	0.504	0.882
GO:0050999	regulation of nitric-oxide synthase acti	8/35	0.009	0.836	0.197
GO:0008180	signalosome	6/35	0	0.947	0.841
GO:0022627	cytosolic small ribosomal subunit	23/35	1	0.994	0
GO:0004712	protein serine/threonine/tyrosine kinase	7/35	0.147	0.934	0.413
GO:0032480	negative regulation of type I interferon	6/33	0.73	0.632	0.775
GO:0006730	one-carbon metabolic process 5/32	0.81	0.722	0.882	
GO:0042147	retrograde transport, endosome to Golgi	5/32	0.819	0.76	0.882
GO:0019843	rRNA binding	9/31	0.009	0.549	0.032
GO:0008542	visual learning	6/29	0.643	0.704	0.519
GO:0000289	nuclear-transcribed mRNA poly(A) tail sh	5/26	0.841	0.915	0.841
GO:2001236	regulation of extrinsic apoptotic signal	4/25	0	0.532	0.882
GO:0006890	retrograde vesicle-mediated transport, G	10/24	0.971	0.909	0.002
GO:0005689	U12-type spliceosomal complex	8/24	1	0.834	0.026
GO:0016775	phosphotransferase activity, nitrogenous	5/18	0	0.57	0.344
GO:0005003	ephrin receptor activity	4/17	0	0.835	0.775
GO:0001502	cartilage condensation	5/16	0.997	0.971	0.25
GO:0071526	semaphorin-plexin signaling pathway	6/16	0	0.599	0.065
GO:0051183	vitamin transporter activity	5/15	0.619	0.674	0.21
GO:0005852	eukaryotic translation initiation factor	9/14	1	0.992	0
GO:0090307	spindle assembly involved in mitosis	3/13	0.063	0.509	0.882
GO:0008195	phosphatidate phosphatase activity	4/12	0.985	0.954	0.365
GO:0008536	Ran GTPase binding	3/12	0.813	0.65	0.882
GO:0005487	nucleocytoplasmic transporter activity	5/11	0.993	0.976	0.065
GO:0015450	P-P-bond-hydrolysis-driven protein trans	5/11	0.612	0.688	0.065
GO:0050321	tau-protein kinase activity	5/11	0.994	0.901	0.065
GO:0006465	signal peptide processing	3/10	0.856	0.569	0.775
GO:0008641	small protein activating enzyme activity	3/10	0.866	0.605	0.775
GO:0005838	proteasome regulatory particle	3/9	0.934	0.636	0.662
GO:0004703	G-protein coupled receptor kinase activi	3/7	0.943	0.892	0.41

Table 2.10: Influenza RNAi; GO terms by Fisher’s method; coverage 97.

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0018107	peptidyl-threonine phosphorylation	12/49	0	0.886	0.019
GO:0032434	regulation of proteasomal ubiquitin-depe	12/48	0	0.897	0.017
GO:0016469	proton-transporting two-sector ATPase co	14/47	0	0.963	0.002
GO:0022627	cytosolic small ribosomal subunit	23/35	1	0.994	0
GO:0030532	small nuclear ribonucleoprotein complex	10/33	0	0.274	0.011
GO:0032436	positive regulation of proteasomal ubiqu	10/31	1	0.103	0.007
GO:0019843	rRNA binding	9/31	0.009	0.549	0.032
GO:0015988	energy coupled proton transport, against	10/30	0	0.015	0.006
GO:0015991	ATP hydrolysis coupled proton transport	10/30	0	0.005	0.006
GO:0006890	retrograde vesicle-mediated transport, G	10/24	0.971	0.909	0.002
GO:0005689	U12-type spliceosomal complex	8/24	1	0.834	0.026
GO:0033176	proton-transporting V-type ATPase comple	9/24	0	0.018	0.006
GO:0030137	COPI-coated vesicle	8/20	0	0.013	0.007
GO:0042274	ribosomal small subunit biogenesis	8/18	0	0.003	0.004
GO:0030663	COPI coated vesicle membrane	8/16	0	0.022	0.003
GO:0046933	hydrogen ion transporting ATP synthase a	7/15	0.003	0.007	0.007
GO:0005852	eukaryotic translation initiation factor	9/14	1	0.992	0
GO:0030126	COPI vesicle coat	8/14	0.03	0.043	0.001
GO:0035964	COPI-coated vesicle budding	7/13	0	0.006	0.003
GO:0048194	Golgi vesicle budding	7/13	0	0.003	0.003
GO:0048200	Golgi transport vesicle coating	7/13	0	0.006	0.003
GO:0048205	COPI coating of Golgi vesicle	7/13	0	0.006	0.003
GO:0033179	proton-transporting V-type ATPase, V0 do	5/10	0.999	0.005	0.046
GO:0000028	ribosomal small subunit assembly	4/6	0.014	0.004	0.046

genes (highlighted in red Figure 2.4). The Benjamini-Hochberg corrected Fisher p-value is 0.017, and so the term would be considered enriched in the standard analysis; it is also inferred to be active by MGSA (posterior probability 0.897). But it is not in the MAP estimate by MFA-ILP and its posterior activation probability is 0. Now 10 of these 12 influenza-involved, GO:0032434 genes are part of the *child* term GO:0032436, *positive regulation of proteasomal ubiquitin-dependent protein catabolic process*, a set of size 31 genes. Both terms are found by the univariate Fisher procedure, exemplifying the redundancy issue; the more specific term GO:0032436 is identified by MFA-ILP and has high posterior activation probability. The MFA

Table 2.11: Influenza RNAi; GO terms with MFA marginal posterior activation probability exceeding 0.5 . Three sets (GO:0046330, GO:0005719, GO:0051233) are not in the MAP estimate (Table 2.8).

GOID	Term (up to 40 characters)	Statistics	P.MFA	P.MGSA	FisherQ
GO:0046330	positive regulation of JNK cascade	9/46	0.714	0.191	0.27
GO:0015030	Cajal body	7/46	0.992	0.773	0.846
GO:0004715	non-membrane spanning protein tyrosine k	7/45	0.864	0.024	0.846
GO:0031526	brush border membrane	8/39	0.957	0.484	0.301
GO:0022627	cytosolic small ribosomal subunit	23/35	1	0.994	0
GO:0032480	negative regulation of type I interferon	6/33	0.73	0.632	0.775
GO:0006730	one-carbon metabolic process	5/32	0.81	0.722	0.882
GO:0042147	retrograde transport, endosome to Golgi	5/32	0.819	0.76	0.882
GO:0032436	positive regulation of proteasomal ubiqu	10/31	1	0.103	0.007
GO:0004693	cyclin-dependent protein kinase activity	5/31	0.62	0.058	0.882
GO:0008542	visual learning	6/29	0.643	0.704	0.519
GO:0070888	E-box binding	5/27	0.597	0.061	0.846
GO:0000289	nuclear-transcribed mRNA poly(A) tail sh	5/26	0.841	0.915	0.841
GO:0048013	ephrin receptor signaling pathway	4/26	0.694	0.078	0.882
GO:0019003	GDP binding	4/26	0.632	0.029	0.882
GO:0043022	ribosome binding	7/26	0.731	0.03	0.152
GO:0006890	retrograde vesicle-mediated transport, G	10/24	0.971	0.909	0.002
GO:0005689	U12-type spliceosomal complex	8/24	1	0.834	0.026
GO:0005839	proteasome core complex	6/20	0.972	0.391	0.169
GO:0002089	lens morphogenesis in camera-type eye	5/19	0.696	0.48	0.404
GO:0001502	cartilage condensation	5/16	0.997	0.971	0.25
GO:0004708	MAP kinase kinase activity	3/16	0.609	0.022	0.882
GO:0051183	vitamin transporter activity	5/15	0.619	0.674	0.21
GO:0005852	eukaryotic translation initiation factor	9/14	1	0.992	0
GO:0042776	mitochondrial ATP synthesis coupled prot	4/13	0.933	0.007	0.417
GO:0051233	spindle midzone	4/12	0.708	0.16	0.365
GO:0008195	phosphatidate phosphatase activity	4/12	0.985	0.954	0.365
GO:0008536	Ran GTPase binding	3/12	0.813	0.65	0.882
GO:0005487	nucleocytoplasmic transporter activity	5/11	0.993	0.976	0.065
GO:0015450	P-P-bond-hydrolysis-driven protein trans	5/11	0.612	0.688	0.065
GO:0019104	DNA N-glycosylase activity	3/11	0.618	0.054	0.846
GO:0050321	tau-protein kinase activity	5/11	0.994	0.901	0.065
GO:0006465	signal peptide processing	3/10	0.856	0.569	0.775
GO:0006600	creatine metabolic process	4/10	0.995	0.221	0.239
GO:0005719	nuclear euchromatin	3/10	0.591	0.367	0.775
GO:0033179	proton-transporting V-type ATPase, V0 do	5/10	0.999	0.005	0.046
GO:0033180	proton-transporting V-type ATPase, V1 do	3/10	0.865	0.004	0.775
GO:0008641	small protein activating enzyme activity	3/10	0.866	0.605	0.775
GO:0006013	mannose metabolic process	3/9	0.847	0.437	0.662
GO:0005838	proteasome regulatory particle	3/9	0.934	0.636	0.662
GO:0036002	pre-mRNA binding	3/9	0.832	0.369	0.662
GO:0006527	arginine catabolic process	3/8	0.514	0.074	0.519
GO:0009103	lipopolysaccharide biosynthetic process	3/8	0.909	0.135	0.519
GO:0019773	proteasome core complex, alpha-subunit c	1/8	0.976	0.004	0.882
GO:0002116	semaphorin receptor complex	4/7	0.545	0.144	0.079
GO:0004703	G-protein coupled receptor kinase activi	3/7	0.943	0.892	0.41
GO:0019798	procollagen-proline dioxygenase activity	3/7	0.885	0.403	0.41
GO:0007100	mitotic centrosome separation	2/5	0.509	0.013	0.846
GO:0005007	fibroblast growth factor-activated recep	3/5	0.866	0.42	0.22

calculation favors the explanation whereby the smaller set is activated; this fails to cover two of the 12 GO:0032434 influenza genes, but it also simplifies the explanation of non-listed genes in that set. If the larger set (GO:0032434) is activated, then we have a lot more mis-covered genes, i.e. those in the set but undetected by RNAi [$15 = (48-12) - (31-10)$]. With this example, it may be that the more specific *positive regulation* term better characterizes the experimental gene list.

Table 2.12 summarizes the performance of the three methods (Fisher’s test, MGSA and MFA-ILP) under two different functional records: GO and KEGG. It shows that MFA-ILP delivers relatively high coverage and low mis-coverage in all cases.

Table 2.12: Coverage, miscoverage, and other statistics for three methods in two case studies. Recall y_g indicates whether or not gene g is on the observed gene list; $\hat{A}_g$ is whether or not the gene is in a set that is inferred to be active. Entries (except last column) are numbers of genes satisfying the condition. In each case and gene-set system, the first two numerical columns add to a constant, as do the next two columns. The final column indicates the number of sets inferred to be active, after trimming. MFA-ILP delivers relatively high coverage and low mis-coverage in all cases.

		coverage		mis-coverage		# of sets
		$y_g=1, \hat{A}_g = 1$	$y_g=1, \hat{A}_g = 0$	$y_g=0, \hat{A}_g = 1$	$y_g=0, \hat{A}_g = 0$	
T2D, GO	Fisher	22	36	265	10303	16
	MGSA	13	45	169	10399	5
	MFA-ILP	26	32	156	10412	10
RNAi, GO	Fisher	97	586	206	8749	13
	MGSA	226	457	634	8321	36
	MFA-ILP	245	438	635	8320	50
T2D, KEGG	Fisher	12	8	55	1846	2
	MGSA	20	0	174	1727	6
	MFA-ILP	20	0	174	1727	6
RNAi, KEGG	Fisher	32	132	71	1686	3
	MGSA	111	53	522	1235	22
	MFA-ILP	108	56	497	1260	21

2.7 Estimating false-positive and true-positive rates

In MFA, it is required to provide the values of false positive α and true positive γ rates. One can usually estimate them by utilizing the information from gene-level data. The techniques used by us in real data examples are provided below, to some extent, which can serve as the pipelines to estimate α and γ .

Type-2 diabetes example

Some R source is given below to utilize the mixture-model fit from Figure 1, Morris *et al.* (2012), in order to estimate the false positive (α) and true positive (γ) rates:

```
## Fig 1 in Morris et al 2012 shows a fitted mixture, applied to SNP-level Z scores,
## aiming to find out how many other T2D genes are out there

p1 <- 0.1465    ## proportion of involved genes
mu0 <- 0
mu1 <- 1.719
var0 <- 1
var1 <- 0.843

N <- 22000 ## guess on total number of genes in genome
x <- 77 ## number of transcripts significantly associated with disease, according to
        ## Morris et al; at least, used in their pathway analysis...
```

```
## task 1: on the z-score scale, find an effective threshold that puts mass  $x/N$ 
## beyond in the mixture model
```

```
mcdf <- function(z)
{
  tmp1 <- (1-p1)*pnorm(z)
  tmp2 <- p1*pnorm(z, mean=mu1, sd=sqrt(var1) )
  return( tmp1+tmp2 )
}
```

```
foo <- function(z)
{
  dd <- (1-mcdf(z)) - x/N
  dd
}
```

```
cc <- ( uniroot( foo, interval=c(0,10) ) )$root
## cc    3.555
```

```
## task 2: on to role model parameters
```

```
## A = gene level activity indicator [truly affects T2D risk]
```

```
## Y = gene level call [found in GWAS]
```

```
pY1 <- x/N
```

```

pA0 <- 1-p1

pY1gA0 <- pnorm( cc, lower.tail=FALSE )          ## alpha
pY1gA1 <- pnorm( cc, mean=mu1, sd=sqrt(var1), lower.tail=FALSE )  ## gamma

## 0.00019  alpha
## 0.02279  gamma

## check on implied FDR
pA0gY1 <- pY1gA0 * pA0/pY1  ## 0.046 ....reasonable

```

Influenza RNAi example

The 984 influenza-involved human genes reported in Hao *et al.* (2013) are those detected in the primary genome-wide screen of at least one of the four studies examined. From the detection model in that paper, we estimated α , the probability of detection in at least one study given the gene is not truly involved in influenza, and $1 - \gamma$, the probability of no detection in any study given the gene is truly involved. Using posterior mean estimates of the *metaflu* model parameters, and noting the key event is pattern ‘0000’, we find:

$$\begin{aligned}\alpha &= P(\text{detected at least once}|\text{not involved}) = 1 - P(\text{‘0000’}|I = 0) = 0.01306943 \\ 1 - \gamma &= P(\text{not detected at all}|\text{involved}) = P(\text{‘0000’}|I = 1) = 0.73197374.\end{aligned}$$

Chapter 3

Implementations and Extensions of Multi-Functional Analyzer

3.1 Forward

In order to implement the MFA analysis, I build an R package for biostatisticians and other scientific researchers, who are interested in gene set analysis, to use. Section 3.2 is a vignette for the package `Rolemodel`. The MAP estimate and the MCMC posterior sampling are both realized in this package. The original MCMC code was written in C++, and then converted into R language. With the help of this package, biostatisticians can apply the MFA analysis to their problems conveniently and efficiently.

Besides the implementation of MFA, I also investigate other important issues regarding the *role model* statistics. Section 3.3 discusses the model diagnostics. Section

3.4 applies MFA to a meta-analysis of four genome-wide studies of influenza virus, and it demonstrates that studies agree substantially more at the level of functional categories than genes. Section 3.5 investigates one possible application of MFA to the peptide-protein inference problem. Section 3.6 presents an approximate algorithm of MFA-ILP.

3.2 Rolemodel: an R package to implement MFA analysis

Overview

The `Rolemodel` package is intended to implement the MFA method described in Wang *et al.* (2015). As mentioned in Chapter 2, MFA is based on the *role model*, first proposed by Bauer *et al.* (2010). In the role model, we have a finite number of wholes and parts; each whole is a set of parts. In the gene set analysis problem, gene sets can be treated as wholes and genes as parts; each gene set is a set of unordered genes. Two computational advances are developed to do the posterior inference: one is the integer linear programming (MFA-ILP), from which we can obtain the MAP (maximum a posteriori) estimate of active gene sets; the other is the penalized MCMC (MFA-MCMC), from which we can obtain the posterior active probability for each gene set. We would like to use MFA-ILP as our primary tool, as it gives a summary functional decoding of the gene list, while posterior probabilities from the MCMC computation provide a measure of confidence in the inferred sets.

In this section, I describe how to use the functions in the `Rolemodel` package. Basically, the function `sequentialRM` performs the MFA-ILP computation, and the function `bp` performs the MFA-MCMC. I also provide a summary table function `rmTable` and a plotting function `rmPlot`. The input for MFA usually include a list of experimentally derived genes (*e.g.*, differentially expressed genes under different cell type conditions), the 0-1 incidence matrix characterizing the inclusion relationship between wholes (gene sets) and parts (genes), the system parameters α (the false positive rate), γ (the true positive rate) and p [denoted as π in Chapters 1 and 2] (the prior active probability for whole nodes), where α , γ and p are all numeric values between 0 and 1, and α is supposed to be less than γ . We can estimate the values of α and γ from the gene-level data (see Section 2.7), and then estimate the value of p via the R package `mgsa`. In this section, let's assume that α , γ and p are known parameters.

Setup

Say we have a list of human genes, and we are interested in their gene-set enrichment. For this example, I will sample a random set of 100 genes from the `org.Hs.eg.db` database.

```
> library(org.Hs.eg.db)
> genes <- sample(mappedRkeys(org.Hs.egSYMBOL), 100)
> head(genes)
```

```
[1] "OR6C4"      "SLC7A4"      "HLFP1"      "CHD5"
[5] "RLN2"       "LOC100418530"
```

Load the `Rolemodel` library.

```
> library(Rolemodel)
```

The `gs2edge` function will take this vector of genes, and create input objects for the `ILP` or `sequentialRM`, and `bp` functions. Gene-set enrichment is based on Entrez IDs; if the gene list is composed of gene symbols, these will be converted to Entrez IDs. `n.upp` and `n.low` are the maximum and minimum gene-set sizes to consider in the enrichment analysis. We recommend an `n.upp` ≤ 50 for `ILP` and `sequentialRM` due to computing time and memory allocation.

```
> gs <- gs2edge(genes, n.upp=30, idtype="SYMBOL", lib="org.Hs.eg")
> I <- gs$I
> I[1:6,1:6]
```

	GO:0000002	GO:0000012	GO:0000014	GO:0000028	GO:0000030	GO:0000038
100	0	0	0	0	0	0
1000	0	0	0	0	0	0
10000	1	0	0	0	0	0
10003	0	0	0	0	0	0
10005	0	0	0	0	0	0
10006	0	0	0	0	0	0

```

> y <- gs$y
> head(y)

100  1000 10000 10003 10005 10006
    0     0     0     0     0     0

> edge <- gs$edge
> head(edge)

      go_id gene_id
3   GO:0001867      2
18  GO:0002921      2
34  GO:0007597      2
75  GO:0030449      2
108 GO:0045824      2
109 GO:0045916      2

```

Gene-set enrichment

I demonstrate the use of the `ILP`, `sequentialRM`, and `bp` using the `t2d` dataset, which is included in the `Rolemodel` library. To speed processing time, consider gene-sets with 5 to 20 genes. Use the `subRM` function to subset the incidence matrix.

```

> data(t2d)
> dim(t2d$I)

[1] 10626  6037

```

```
> sum(t2d$y)
```

```
[1] 58
```

```
> newI <- subRM(t2d$I, 5, 20)
```

```
> newy <- t2d$y[rownames(newI)]
```

```
> dim(newI)
```

```
[1] 8437 4449
```

```
> sum(newy)
```

```
[1] 52
```

As mentioned before, we need to specify the values of α , γ and p before running MFA. The p parameter can be estimated using the `mgsa` library, via the `estP` function. In this example, we are already given $p=0.00331$. As an exercise, however, we estimate p :

```
> alpha <- 0.00019
```

```
> gamma <- 0.02279
```

```
> pest <- estP(I=newI, y=newy, alpha, gamma)
```

```
> pest
```

```
[1] 0.004495392
```

An internal study indicates that the algorithm is not overly sensitive to the exact settings of these parameters, but to their range.

MFA-ILP: MAP estimate via integer linear programming

We use the R package `Rglpk` to do the integer linear programming, which is an R interface to GNU Linear Programming Kit intending to solve large-scale linear programming problems. The function `ILP` in our package is a wrapper designed for the specific gene-set problem by invoking the function `Rglpk_solve_LP` in `Rglpk` directly.

```
> alpha <- 0.00019
> gamma <- 0.02279
> p <- 0.00331
> # do not run
> # res <- ILP(newI, newy, alpha, gamma, p)
```

It is very time-consuming to run `ILP` directly when the dimension of the incidence matrix is big. So one suggestion is that using the function `sequentialRM` rather than `ILP` to do the ILP related computation. `SequentialRM` is a wrapper to implement the two approaches (shrinkage and sequential strategy) developed in Wang *et al.* (2015) to speed up the computation.

```
> res <- sequentialRM(newI, newy, nupstart=10, by=1, alpha, gamma, p)

[1] 11
[1] 12
[1] 13
[1] 14
```

```
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
```

```
> str(res)
```

```
List of 2
```

```
$ onwholes: chr [1:8] "GO:0001714" "GO:0045725" "GO:2000679" "GO:0006983" ...
$ sol      :List of 3
..$ optimum : num 44.1
..$ solution: Named num [1:122] 0 0 0 0 0 0 1 1 1 1 ...
.. ..- attr(*, "names")= chr [1:122] "GO:0032460" "GO:0032461" "GO:0032462"
"GO:0032464" ...
..$ status  : int 0
```

Here `nupstart` indicates the starting upper bound of the number of genes in a set, which is used to subset the incidence matrix, and `by` indicates the increment of the upper bound for each iteration in the sequential strategy. One suggestion is to set `nupstart` ≤ 10 , and `by` = 1. The output `res` is a list consisting of two parts. The first part `onwholes` includes the names of identified active GO terms (gene sets), *i.e.*, the MAP estimate. In this example, there are totally 8 GO terms identified as active. The second part `sol` is extracted directly from the output of function

`Rglpk_solve_LP` in the package `Rglpk`, where `optimum` is the value of the objective function at the optimum, `solution` is the vector of optimal coefficients (0-1vector), and `status` is an integer with status information about the solution returned. It will return 0 for the optimal solution being found, and non-zero otherwise. Actually, the MAP estimate is just the whole nodes whose values are taking 1 in the optimal solution.

```
> names(res$sol$solution)[res$sol$solution == 1][1:8]

[1] "G0:0001714" "G0:0045725" "G0:2000679" "G0:0006983" "G0:0031017"
[6] "G0:0070365" "G0:0005638" "G0:0060395"
```

MFA-MCMC: the posterior active probability

The function `bp` is used to do the MCMC sampling. The input for `bp` is a bit different from that in `sequentialRM` and is listed as follows:

- `whole` Vector of character strings with names of whole nodes
- `part` Vector of 0's and 1's indicating active part nodes; names are the names of the part nodes
- `edge` Matrix with two columns; each row indicates an edge between a whole node and a part node
- `alpha`, `gamma`, `p` the same as before
- `nburn` Number of burn-in generations

- `ngen` Number of sample generations
- `sub` Subsample rate for burn-in and sample files
- `penalty` Penalty per illegal node to loglikelihood
- `initial` Initial states of whole nodes, which can take one of three values:
`inactive` - all whole nodes inactive; `random` - all whole nodes active with probability `p`, no illegal nodes; or `high` - all nodes with proportion of connected part nodes with response equal to 1 above 0.4 are active, no illegal nodes.

```
> eout <- I2edge(newI)
> bp.out <- bp(whole=colnames(newI), part=newy, edge=eout, alpha, gamma, p,
  nburn=1000000, ngen=10000000, sub=1000, penalty=5, initial="random")
> str(bp.out)

'data.frame':      4449 obs. of  6 variables:
 $ Name           : chr  "G0:0000002" "G0:0000012" "G0:0000028" "G0:0000042" ...
 $ ActiveProbability: num  0 0.000734 0.001493 0.003283 0.002278 ...
 $ Count          : int  0 6699 13625 29955 20790 0 16839 0 12197 50745 ...
 $ Sample         : int  9124582 9124582 9124582 9124582 9124582 9124582 9124582 ...
 $ Degree         : int  13 8 6 14 11 10 10 11 5 12 ...
 $ Response       : int  0 0 0 0 0 0 0 0 0 0 ...
- attr(*, "samples")= num [1:11000, 1:10] 2.70e-03 2.90e-03 8.08e-10 2.96e-03 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : NULL
.. ..$ : chr  "AcceptanceProb" "LogLikelihood" "LogPrior" "ActiveOne" ...
```

Among the output of `bp` (see the help file for a full explanation), we are most interest in the `ActiveProbability`, which is the posterior active probability for each whole node. Generally, cut-off value 0.5 is used to pick GO terms as active.

```
> bp.out$Name[bp.out$ActiveProbability>0.5]

[1] "GO:0001714" "GO:0006983" "GO:0031017" "GO:0045725" "GO:0060395"
[6] "GO:0060612" "GO:0005638"
```

As a result, these GO terms can be seen as the active ones inferred from the MCMC sampling.

An integrative analysis function: summary of the result

As mentioned before, we would like to use MFA-ILP as our primary tool, while posterior probabilities from the MCMC computation provide a measure of confidence in the inferred sets. Of course, one can use the MCMC sampling result independently. We would like to report the MAP estimate in the decreasing order of the posterior active probabilities.

```
> tmp <- bp.out$ActiveProbability[bp.out$Name%in%res$onwholes]
> ord <- order(tmp, decreasing = T)
> output<- data.frame(MAP = res$onwholes[ord], P.MFA = tmp[ord])
> print(output)
```

	MAP	P.MFA
1	GO:0006983	0.9872778

```

2 GO:2000679 0.8859441
3 GO:0060395 0.8854952
4 GO:0045725 0.8627853
5 GO:0001714 0.5656231
6 GO:0031017 0.5393540
7 GO:0070365 0.3934441
8 GO:0005638 0.2145722

```

On the other hand, we have also created an overall function `rmTable` to do both MFA-ILP and MFA-MCMC analysis simultaneously. Thus users can take total control of the parameters and interpret the result in an easier way. The input here is just a list of genes of interest, e.g., which can be a list of differentially expressed genes according to some different cellular conditions. The type of this gene list can be in Entrez ids, gene symbols, Ensembles, etc. The output includes four parts: the first part is a summary data frame about the active gene sets, including their GO terms, set size, number of associated genes included, and so on; the second part is the solution of ILP calculation (the same as the output of `sequentialRM`); the third part is the solution of MCMC calculation (the same as the output of `bp`); and the last part is the incidence matrix and the response vector based on the input gene list. Notice that since the up-to-date data source is used here, the result is slightly different from the one obtained previously.

```

> data(T2D_genelist)
> idlist <- idlist[-57] ## no Entrez id for "KLHDC5"

```

```
> res <- rmTable (idlist, lib="org.Hs.eg", n.upp=20, n.low=5, nupstart=10,
by=1, alpha=0.00019, gamma=0.02279, p=0.00331)
```

```
[1] 11
```

```
[1] 12
```

```
[1] 13
```

```
[1] 14
```

```
[1] 15
```

```
[1] 16
```

```
[1] 17
```

```
[1] 18
```

```
[1] 19
```

```
[1] 20
```

```
> str(res)
```

```
List of 4
```

```
$ rm.table      : 'data.frame':      6 obs. of  7 variables:
```

```
..$ Term          : chr [1:6] "lamin filament" "axolemma" "positive
regulation of transcription initiation from RNA polymerase II promoter"
"regulation of transcription involved in cell fate commitment" ...
```

```
..$ Ontology       : chr [1:6] "CC" "CC" "BP" "BP" ...
```

```
..$ MAP           : Factor w/ 6 levels "G0:0005638","G0:0030673",...: 1 ...
```

```
..$ P.MFA         : num [1:6] 0.987 0.939 0.882 0.878 0.871 ...
```

```
..$ set.size      : num [1:6] 5 12 7 20 15 6
```

```

..$ number.associated.genes: num [1:6] 2 2 2 3 5 4
..$ genes                  : Factor w/ 6 levels "AKT2;GCK;INS;INSR;IRS1",...: 4 ...
$ rm.sol                  :List of 2
..$ onwholes: chr [1:6] "G0:0045725" "G0:0060261" "G0:0005638" "G0:2000675" ...
..$ sol                   :List of 3
.. ..$ optimum : num 50.8
.. ..$ solution: Named num [1:178] 0 0 0 0 0 0 0 1 1 1 ...
.. ..- attr(*, "names")= chr [1:178] "G0:0010898" "G0:0032000" "G0:0055015" ...
.. ..$ status   : int 0
$ bp.sol          : 'data.frame':      5146 obs. of  6 variables:
..$ Name          : chr [1:5146] "G0:0000002" "G0:0000012" "G0:0000014" ...
..$ ActiveProbability: num [1:5146] 0 0.002126 0.00157 0.003097 0.000553 ...
..$ Count          : int [1:5146] 0 18907 13959 27536 4920 6976 64457 0 0 0 ...
..$ Sample         : int [1:5146] 8891255 8891255 8891255 8891255 8891255 ...
..$ Degree         : int [1:5146] 20 8 6 8 17 15 5 11 13 6 ...
..$ Response       : int [1:5146] 0 0 0 0 0 0 0 0 0 0 ...
..- attr(*, "samples")= num [1:11000, 1:10] 2.70e-03 1.86e-05 2.90e-03 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$ : NULL
.. .. ..$ : chr [1:10] "AcceptanceProb" "LogLikelihood" "LogPrior" "ActiveOne" ...
$ incidence.mat:List of 3
..$ I      : num [1:9397, 1:5146] 0 0 1 0 0 0 0 0 0 0 ...
.. ..- attr(*, "dimnames")=List of 2

```

```

.. .. ..$ : chr [1:9397] "100" "1000" "10000" "10003" ...
.. .. ..$ : chr [1:5146] "G0:0000002" "G0:0000012" "G0:0000014" "G0:0000028" ...
..$ y      : Named num [1:9397] 0 0 0 0 0 0 0 0 0 0 ...
.. ..- attr(*, "names")= chr [1:9397] "100" "1000" "10000" "10003" ...
..$ edge:'data.frame':      51180 obs. of  2 variables:
.. ..$ go_id  : chr [1:51180] "G0:0001867" "G0:0002921" "G0:0007597" ...
.. ..$ gene_id: chr [1:51180] "2" "2" "2" "2" ...

```

A plot function

In order to visualize the active gene sets, we make a plot function `rmPlot` to display an image about the enriched gene sets and the associated genes contained by them. At the top row of the plot, it's the gene set having the highest overlap with the input gene list; then this gene set and genes in this set are removed from the system. The second row will plot the gene set having the highest overlap with the remainder, and so on.

```
> rmPlot(res[[4]]$I, res[[4]]$y, res[[2]]$onwholes)
```

Discussion

The R package `Rolemodel` mainly implements two computational approaches in the gene-set analysis problem: MFA-ILP and MFA-MCMC. When the system in consideration is very big, MFA-ILP usually has a long running time and a high requirement on the computer memory size, due to the complexity of integer linear

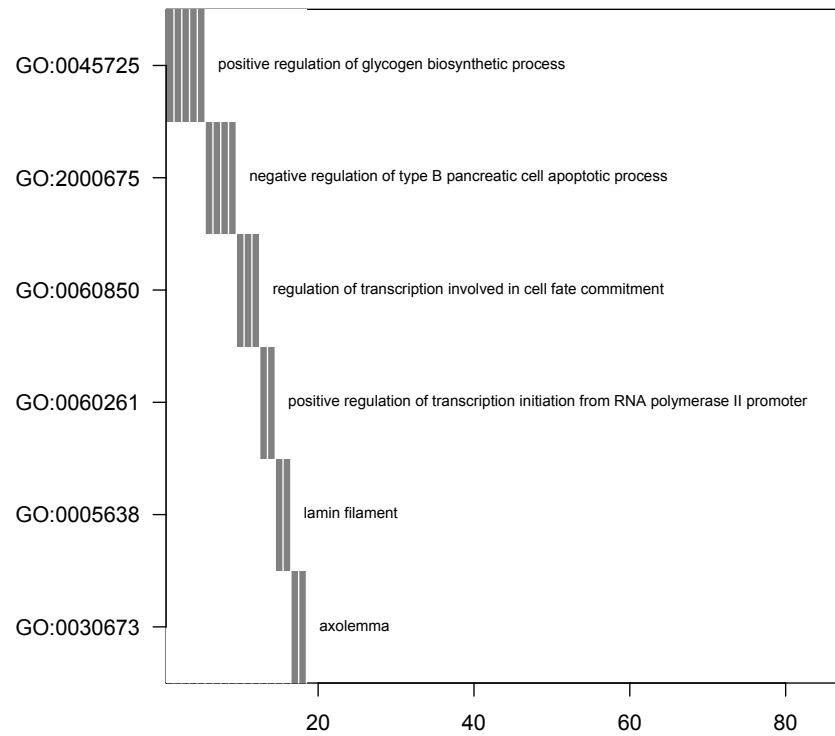


Figure 3.1: Display the active gene sets. Top row is the gene set having the highest overlap with the input gene list; then this gene set and genes in this set are removed from the system. The second row is the gene set having the highest overlap with the remainder, and so on.

programming. Section 3.6 proposes an approximate algorithm to compute a local maximum of the likelihood of role model. The local maximum obtained is close to the real MAP estimate, while the computation time is much faster. Additionally, the estimation of parameters α , γ and p is a more difficult problem. We have developed some pipelines (see Section 2.7) to estimate the values of α and γ ; to develop an approach to estimate p without the help of package `MGSA` can contribute to make MFA as a standalone gene-set enrichment method.

3.3 Model diagnostics and other issues

The full effects of prior choice in model-based gene-set analysis require further investigation. As to the practical importance of one choice over another, we do not examine the biological distinctions between the inferences produced by different methods. A close reading of the T2D and RNAi case studies in Chapter 2 provides an initial indication of how and why reported set lists can differ, but assessing the biological significance of these differences is beyond the present scope. The procedures have distinct statistical properties and MFA more efficiently captures the functional content of the reported gene list in terms of model fit. We point out that the distinctions present themselves when using the relatively complex GO system. Control calculations show that `MGSA` and MFA give essentially the same results when applied to the less complex KEGG system (see Figures 3.2 and 3.3).

In our comparisons we used `MGSA` to obtain an estimate of the hyper-parameter π , which affects the overall rate of set activity. In order to control the comparison,

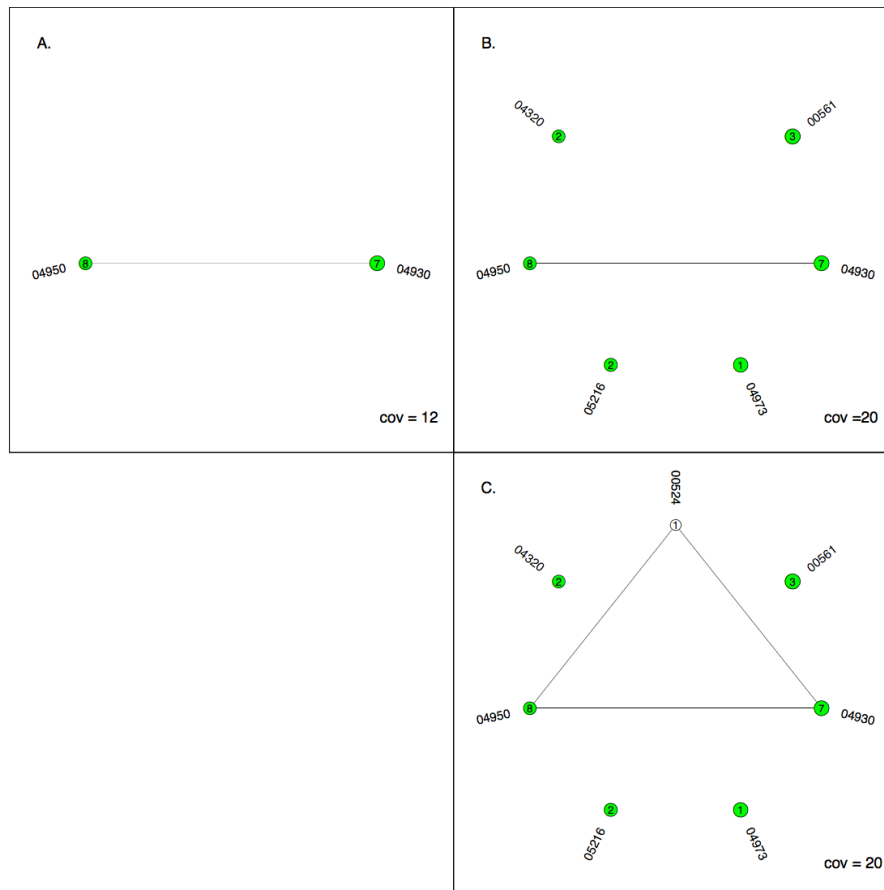


Figure 3.2: KEGG pathways identified by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) as activated to explain the type 2 diabetes associated genes. Layout as in Figures 2.3 and 2.4. There is very close agreement between MGSA and MFA-ILP.

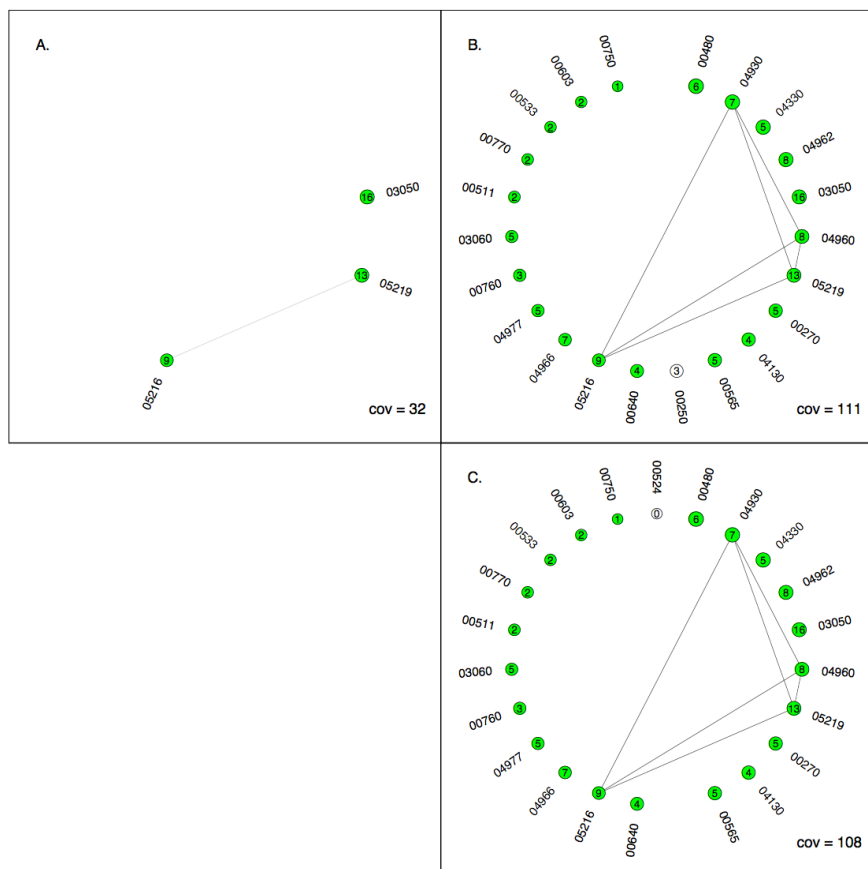


Figure 3.3: KEGG pathways identified by three methods (A, Fisher's test; B, MGSA; C, MFA-ILP) as activated to explain the influenza RNAi genes. Layout as in Figures 2.3 and 2.4. There is very close agreement between MGSA and MFA-ILP.

we used the same numerical value of π in the MFA calculations (in both MFA and MGSA we fixed the other parameters α and γ at externally derived values). Further improvements of MFA may be possible using alternative estimates of π . Other model elaborations which could be useful in some applications include extending MFA beyond binary X_p and allowing dependence in gene level measurement errors.

Compute times for MFA depend on the size and content of the gene list, the incidence matrix I , and the model parameters: MFA-ILP used 2.5 CPU hours for T2D, and 23 CPU hours for RNAi (R was running on a 4×AMD Opteron(TM) Processor 6174 (48 cores) with 128 GB RAM); less time was required for MFA-MCMC (20 and 45 CPU minutes, resp.).

We have argued that temporal changes in GO reflect an increase in the complexity of the functional record that justifies the more refined prior distribution used in MFA (Figure 2.1). These changes also tell us that the results of a given analysis naturally depend on the GO version, since the sets involved and the annotations of genes to sets continue to evolve. To assess the version effect, we applied MFA to four recent GO versions (2010-2013) in both the T2D and RNAi examples, and using sets annotating between 5 and 50 human genes (GO[5:50]). Table 3.1 records how similar are time-adjacent versions of GO as well as how similar are results of MFA. The changes in GO reflect new terms and new annotations, as well as sets moving in and out of GO[5:50] as more genes become annotated. Against this substantial evolution of GO we see that MFA results do also change, but that the changes are less the more stringent is the marginal posterior probability cutoff.

A feature of MGSA and MFA is that activation of the set implies activation of all

Table 3.1: Version effects: Tabulated are similarity scores comparing set structure and inferred active sets over time-adjacent versions of GO[5:50]. The collections of sets annotating between 5 and 50 human genes fluctuate over recent versions of GO. Respectively, in the four most recent fall versions of **Bioconductor**, the collections contain 4830, 5546, 6037 and 6488 gene sets. The first row shows the Jaccard index (size of intersection over size of union) comparing subsequent versions of these gene-set collections. In addition to the collections changing, the annotations recording which genes are in which sets also change over time. The second row measures similarity of the sets of annotations. Subsequent rows show similarity of reported lists of gene sets in the two main examples. In this comparison, the set is reported if it is in the MAP estimate by MFA-ILP and if its marginal posterior probability exceeds a threshold (50% or 80%). Inferred active sets depend to some extent on the GO version in view; setting stronger marginal posterior thresholds reduces the false-discovery rate and reduces the version effect.

	2010-2011	2011-2012	2012-2013
Sets	0.72	0.82	0.80
Annotations	0.54	0.70	0.66
T2D 50/MAP	0.33	0.60	0.30
T2D 80/MAP	1.00	0.75	0.60
RNAi 50/MAP	0.72	0.70	0.88
RNAi 80/MAP	0.90	0.79	0.85

genes in the set. This strict form of nonnull relationship is quite different from many univariate methods, which would claim a set is nonnull if any of its contained genes is nonnull (e.g., the *self-contained* tests of Goeman and Buhlmann, 2007). It is precisely this relationship, however, that enables multivariate (i.e., multi-set) analysis, as the role model offers a straightforward approach to deal with the complex overlaps in the functional record. We note that the role model allows a weaker interpretation; for instance, we could continue to assert that a gene is activated if it is contained in any activated set, while allowing that only a fraction of activated genes are nonnull. The difference would be in the tabulation of errors (e.g., $X_p = 0$ might not be a false negative when $A_p = 1$) and in the interpretation of α and γ ; the family of joint

distributions would be the same. As GO and other repositories record functions of ever more specific gene combinations, it is reasonable to expect that a combination of genes relevant in the cells on test is within the repository. The strict interpretation of activation is parsimonious and is justified when the repository is sufficiently well endowed with relevant sets. We performed a small simulation study, using the T2D data structure, to assess MFA’s ability to recover small activated sets. In each of 100 simulated cases, we fixed the repository (GO[5:50]), we randomized the response vector $X = (X_p)$ by appending to the T2D genes a randomly selected small (5-10 genes) set from GO, and we inferred the activated sets using MFA. In 91 cases the appended set was identified as active by MFA-ILP, demonstrating in a limited way the ability of the methodology to recover signals represented in the repository.

3.4 A meta-analysis of four genome-wide studies of influenza virus

Genome-wide RNA interference (RNAi) analysis is a powerful approach to identify gene functions that support or modulate selected biological processes. An emerging challenge shared with some other genome-wide approaches is that independent RNAi studies often show limited agreement in their lists of implicated genes. In a meta-analysis of four genome-wide studies of influenza virus, Hao et al. (2013) reported that 984 human genes had been detected by RNA interference as possibly being associated with viral replication. From these 984 unique human genes identified in the primary screens of these four studies, 614 unique genes were confirmed that promoted

influenza virus replication. Pair-wise overlap among the four gene lists was only 3% to 15% (average 6.7%). However, a number of functional categories (gene sets) were overrepresented in multiple studies. The pair-wise overlap of these enriched-category lists was high, $\sim 19\%$, implying more agreement among studies than apparent at the gene level.

Application of MFA

Gene-set analysis was performed using confirmed genes from each of the four studies and functional-category information from the Gene Ontology (GO) project, as accessed using the R and Bioconductor systems (www.bioconductor.org). MFA was used to process each gene list (four study-specific gene lists and one combined list). Gene-permutation was used to calibrate the between-study agreement of MFA-derived set lists. Specifically, data were organized in an incidence matrix with rows for genes and columns for studies, and containing indicators of confirmation. Permutation proceeded by shuffling row labels (gene IDs) and retaining the numbers of genes co-confirmed in two, three, and four studies, and expressed the null assumption of no functional association between studies. We computed Monte Carlo p-values using the MFA results from the observed data and from 999 permuted versions. Computations used `GO.db` version 2.8.0; and `org.Hs.eg.db` version 2.8.0 (September, 2012). In total 2472 GO terms were used; these are all terms that have non-empty intersection with the combined list of influenza-related genes and that are moderate in size (between 5 and 50 human genes). The size constraint improved computational efficiency without much loss in functional information.

Studies agree substantially more at the level of functional categories than genes

Although all four studies aimed to perform a general identification of host genes affecting influenza virus replication, their gene lists exhibited relatively little overlap. Only one gene was detected and confirmed by all four studies; nine genes were confirmed by three of four studies, and 35 by two studies. Pairwise overlap between studies ranged from 3% to 15%, with a mean pairwise overlap of 6.7%.

To complement comparisons at the gene level, we examined the relationship of the four confirmed-gene lists at the level of functional categories obtained by applying MFA to each gene list. Within-study lists of over-represented GO terms exhibited relatively strong agreement (Figure 3.4), with a mean overlap fraction of 19% (compared to 6.7% for gene lists). This agreement is substantially more than would be expected in the absence of functional associations between the studies (Monte Carlo p -value = 0.001), which suggests that the four studies were probing common functional signals in influenza virus dependence on host genes. The functional categories over-represented in the combined list of 614 confirmed genes from all four studies represent a wide variety of functions. The top 29 categories are illustrated in Figure 3.5, including with categories associated with mRNA translation (ribosomal small subunit, eIF3 initiation complex, ribosome binding), vesicular transport (Golgi to ER, vacuolar ATPase), RNA metabolism (RNA splicing, transport, poly(A) regulation), regulated protein degradation (proteasome, ubiquitination factors), and other functions. Cellular functions like regulation of type I interferon production and nucleo-cytoplasmic transport are known to play important roles in influenza virus replication, while other

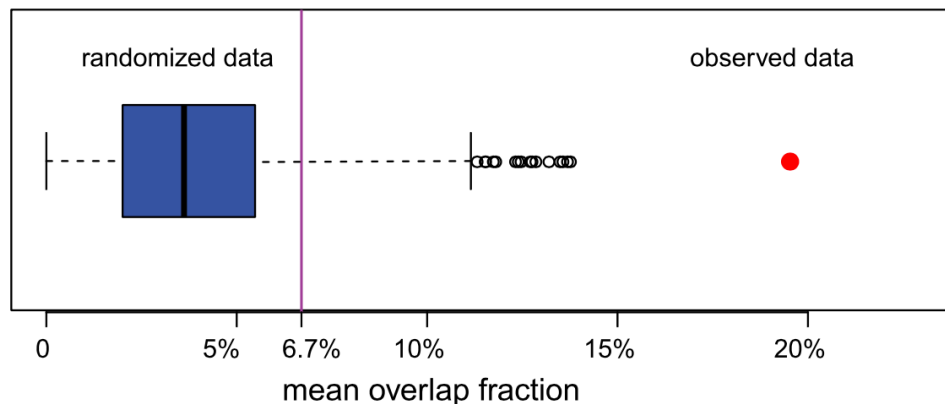


Figure 3.4: Studies agree substantially at the level of functional categories. When each study’s confirmed gene list is processed for enriched functional categories from GO (Gene Ontology), substantially more categories are identified in multiple studies than one would expect by chance, suggesting a positive association in functional signals underlying the study-specific gene lists. Shown is the distribution of an agreement statistic (mean overlap fraction) generated by gene-permutation, where the multi-functional analyzer (MFA) approach is applied to every gene list to produce a list of over-represented GO terms. The actual agreement between studies (red) is strikingly higher than expected by chance (blue) and also much higher than the 6.7% gene-list agreement (purple).

functions like NADP binding and vitamin transporter activities are novel findings identified through our analysis. Figure 3.5 also shows that, for many categories, two or more studies isolated distinct sets of genes in the same functional category, so that the same cellular functions were independently but repeatedly identified to promote influenza virus replication.

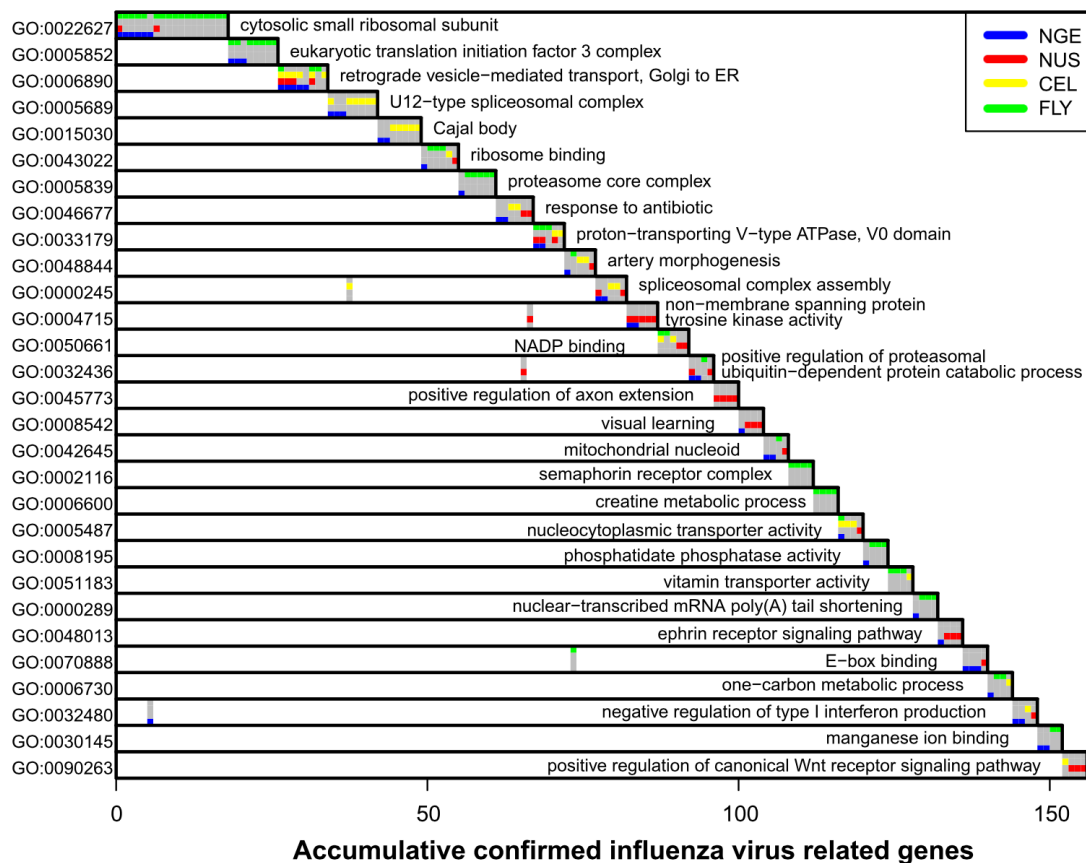


Figure 3.5: Dominant Gene Ontology terms identified in analysis of combined gene list. The multi-functional analyzer (MFA) applied to the combined list of 614 influenza-involved genes produced 91 GO terms in the maximum a posteriori (MAP) estimate of activated terms. The dominant terms are shown here, with color bars to indicate the study responsible for identifying the gene. The MFA-identified category having highest overlap with the 614-gene list is GO:0022627 (top row), the overlap being 18 genes (horizontal axis). These genes and this category are removed from the system, and then the category having the highest overlap with the remainder is identified (GO:0005852), and so on until the overlap is less than four genes. Shading in the sub-diagonal indicates genes also annotated to previously named categories.

3.5 The application of MFA to the peptide-protein inference problem

The protein-peptide inference problem is to assemble peptide identification back to the protein level. The most common approach in mass spectrometry (MS)-based bottom-up proteomics for the large-scale identification of proteins is the matching of peptide tandem mass spectra (MS/ MS) to theoretical spectra constructed from an *in-silico* digested protein sequence database (Steen, 2004). In this process, the protein sequence database constitutes the available sequence space which should ideally be both complete and correct. The most widely used and well annotated databases in the proteomics field are the International Protein Index (IPI), the UniProt Knowledgebase (UniProtKB), the NCBI Protein sequence database and the NCBI Reference Sequence collection (RefSeq).

There is usually a pipeline to choose candidate proteins, and other information (*e.g.*, mass spectrometry sequence in the database and quantitative scores for each peptide) is often used in the identification process. Some Bayesian approaches have been proposed for the protein inference in shotgun proteomics. To some extent, the peptide-protein inference problem is similar to the gene set analysis problem. We believe the role model has a potential to shed light on the peptide-protein inference problem.

The similarity between protein sequences and multiple protein isoforms causes the massive redundancy between proteins. There are, however, also some background differences: in the GSA problem, gene set is only a concept; in the peptide-protein

problem, proteins and peptides are real substances, and peptides are the outcome of the digestion of proteins. For example, if peptide A only exists in protein B and peptide A is found in the digestion outcome, one can determine that protein B must be present in the sample; but in the GSA case, one cannot conclude that B (gene set) must be active. If some protein is in the sample, only a limited proportion of its peptide components will be identified. On the other hand, there is little chance to identify peptides which are not in any proteins in the sample. In this sense, small gamma value (true positive rate) and much smaller alpha value (false positive rate) should be a good fit to our model.

One example

Fortunately, we have a real data example about some proteins and their digestion peptides. The original data include 2,085,887 peptides and 13,297 proteins, and 21,978 identified peptides. We can use a bipartite graph to characterize the relationship between peptides and proteins. The original graph has 1,925 components with a super big one. It's found that there are a lot of duplicate peptides (*i.e.*, they are contained by the same proteins) in the original data. We collapsed these duplicates into a single peptide, which resulted in 35,216 peptides. This process will not change the structure of the original graph and the peptide-level information was also retained during the process: if some peptide is identified, it will be preserved by being picked as the exemplar among its duplicates. 3433 peptides are treated as being identified in the reduced data. This process does not change the number of components of the original graph either (still 1925 components with a super big one).

Next we carried out the role model calculations based on the reduced data (3,433 identified peptides out of 35,216). As for the parameters involved in the role model, we think that small γ and smaller α should be used. A series of parameters (π is fixed as 0.2) are implemented, and the results are given in Table 3.2.

Table 3.2: A series of parameters are implemented in the role model calculations. Coverage is the number of identified peptides explained by the identified proteins, and mis-coverage is the number of non-identified peptides contained by the identified proteins. There are totally 3,433 identified peptides.

(α, γ)	# of identified proteins	Coverage	Mis-coverage
(0.01, 0.1)	2944	3252	10451
(0.001, 0.1)	3084	3409	11772
(0.0001, 0.1)	3092	3414	11814
(0.001, 0.01)	3044	3345	11635
(0.0001, 0.01)	3086	3412	11810
(0.00001, 0.01)	3094	3412	11827

As a comparison, the shotgun technique introduced by Nesvizhskii (2005) identified 2706 proteins with coverage = 855 and mis-coverage = 11790. As shown in the table, the efficiency of MFA has been improved a lot. Take 3092 identified proteins under $\alpha = 0.0001$ and $\gamma = 0.1$ for example. For the sake of visualization, we focus on the identified proteins with medium size: there are 338 proteins holding between 10 and 15 peptides which role model found but the shotgun technique did not (see Figure 3.6), and 444 proteins holding between 10 and 20 peptides which role model found but shotgun did not (see Figure 3.7).

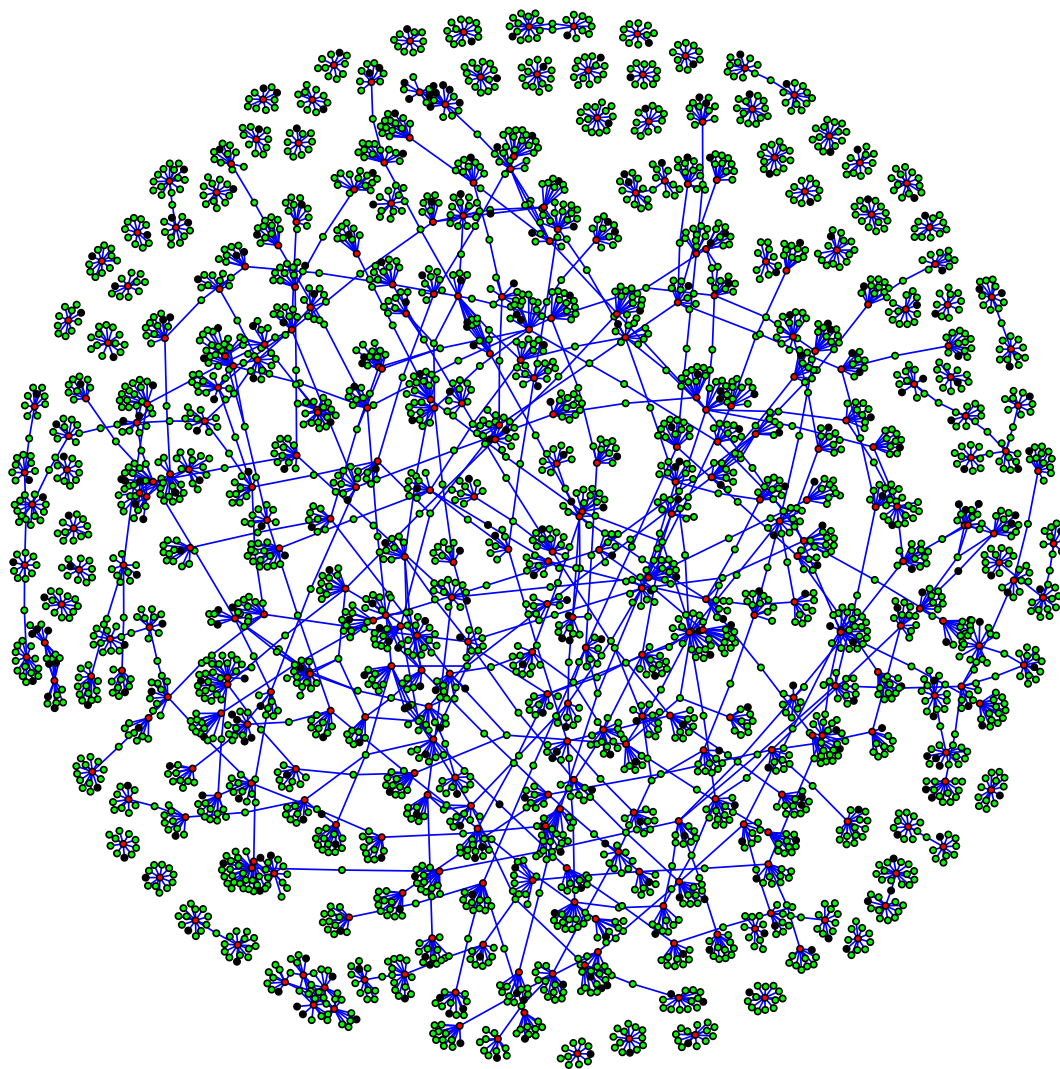


Figure 3.6: 338 proteins (red) holding between 10 and 15 peptides found only by role model, and the associated 3649 peptides (green or black; black are identified peptides), presented as a bipartite graph. The edges (blue) connect peptides to their containing proteins.

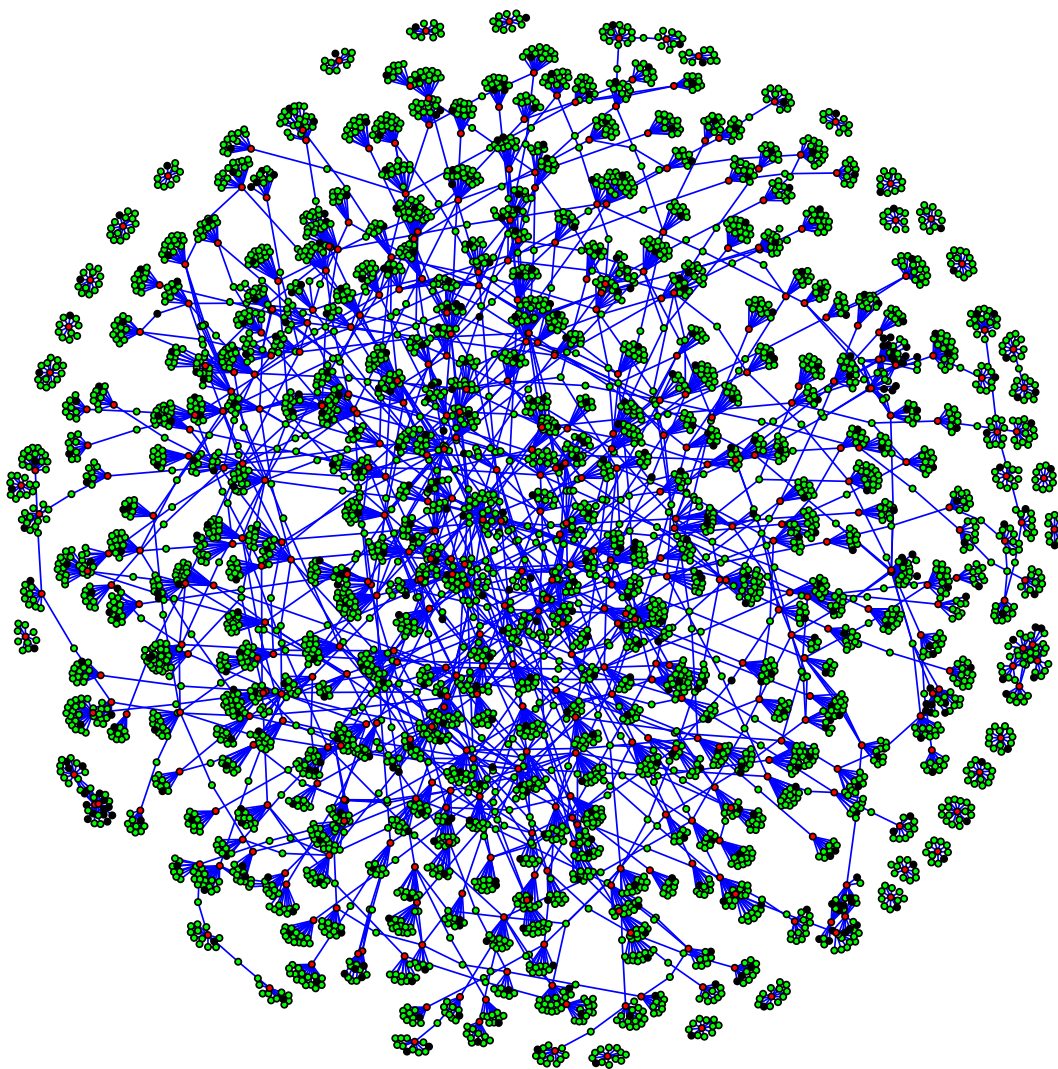


Figure 3.7: 444 proteins (red) holding between 10 and 20 peptides found only by role model, and the associated 5068 peptides (green or black; black are identified peptides), presented as a bipartite graph. The edges (blue) connect peptides to their containing proteins.

3.6 An approximate algorithm for MFA-ILP

The computation by integer linear programming can still be a challenge when the system in consideration is very big, although we already develop two techniques, which adapt to our specific problem, to save the computation time. One idea is to develop an approximate algorithm to compute a local maximum of the likelihood. The local maximum should be close to the real MAP estimate, while the computation would be much faster.

We already know that up to an additive constant, the log posterior (objective function) of role model is:

$$\begin{aligned}
 l(C) = l(Z_w, A_p) = & \sum_w \{Z_w \log(\pi) + (1 - Z_w) \log(1 - \pi)\} + \\
 & \sum_p \{A_p [x_p \log(\gamma) + (1 - x_p) \log(1 - \gamma)] + \\
 & (1 - A_p) [x_p \log(\alpha) + (1 - x_p) \log(1 - \alpha)]\},
 \end{aligned}$$

where C is a list of active GO sets (thus values of Z_w and A_p can be decided from C), x_p is the realized value of the gene-list indicator X_p , and α , γ , and π are system parameters, which are considered fixed in the present calculation.

Given an input list of active genes, we would like to determine a list of active GO sets (C) that maximizes the likelihood function above. This is an NP-hard problem. Motivated by Lu *et al.* (2008), we propose a simple and fast greedy search algorithm to look for a local maximum of the likelihood function. At each iteration, the algorithm searches for sets to add to (or remove from) to the current list of active

sets in order to improve the global likelihood. If such sets exist, the one leading to the highest improvement in likelihood is added to (or removed from) to the list and the search continues. Otherwise the current list is returned as the optimal list of active sets.

An approximate algorithm for optimization by greedy search

- 1) Initialize C_0 to be the empty set.
- 2) At iteration i , consider all possible one-step changes of the current set of active GO sets (C_i), and compare the likelihood of the resulting sets. Let

$$t_1^i = \arg \max_{t \in C_i} L(C_i \setminus \{t\}) \text{ and } t_2^i = \arg \max_{t \in T \setminus C_i} L(C_i \cup \{t\}),$$

where T is the set of all GO sets. Thus among all possible reductions of C_i , $C_i^- = C_i \setminus \{t_1^i\}$ has the highest likelihood. Similarly, among all possible expansions of C_i , $C_i^+ = C_i \cup \{t_2^i\}$ has the highest likelihood.

- 3) If the likelihood of C_i^- is higher than that of both C_i^+ and C_i , let $C_{i+1} = C_i^-$ and go to Step 2.
 - 4) If the likelihood of C_i^+ is higher than that of C_i , let $C_{i+1} = C_i^+$ and go to Step 2; otherwise go to next step.
 - 5) Return C .
-

In the calculations of the current list of active gene sets (C) or the values of Z_w and A_p , the activation hypothesis (AH) is always taken into account. Some R codes to implement the algorithm above are given as follows:

```

## input information:

## I (incidence matrix), y (0-1 response vector), alpha, gamma and p
for(i in 1:nrow(I))
  store[[i]] <- colnames(I)[I[i,]==1]

## function to calculate the likelihood
obj <- function(Ap, alpha, gamma, p )
{
  chose<-unique(unlist(store[Ap==1]))
  chose2<-unique(unlist(store[Ap==0]))
  Zw <- as.numeric(colnames(I)%in%setdiff(chose,chose2))
  loglike <- sum(Zw)*log(p) + (ncol(I)-sum(Zw))*log(1-p) + log(gamma)*Ap%*%y
    + log(1-gamma)*Ap%*%(1-y) + log(alpha)*(1-Ap)%*%y
    + log(1-alpha)*(1-Ap)%*%(1-y)

  return(list(loglike=loglike, C=colnames(I)[Zw==1], Ap=Ap))
}

## function to realize the iterative approach
iteration <- function(C, Ap, alpha, gamma, p)
{
  minusopt <- -Inf

```

```

if(length(C)>=1){
minus <- sapply(C, function(x) {
                                tmp <- setdiff(C,x);
                                if(length(tmp)>1)
                                    Ap <- as.numeric(apply(I[,tmp],1,sum)>0)
                                else if (length(tmp)==1) Ap <- I[,tmp]
                                else Ap <- rep(0, nrow(I));
                                obj( Ap, alpha, gamma, p)})
minusT <- minus[2,][[which.max(unlist(minus[1,]))]]
minusopt <- max(unlist(minus[1,]))
minusAp <- minus[3,][[which.max(unlist(minus[1,]))]]
}

plus <- sapply(setdiff(colnames(I), C),
               function(x) { Ap<-as.numeric(Ap+I[,x]>0); obj( Ap, alpha, gamma, p)})
plusT <- plus[2,][[which.max(unlist(plus[1,]))]]
plusopt <- max(unlist(plus[1,]))
plusAp <- plus[3,][[which.max(unlist(plus[1,]))]]

Cvalue <- obj(Ap, alpha, gamma, p)$loglike
if(max(Cvalue,minusopt,plusopt) == Cvalue)
    return(list(C=C, Ap=Ap))
else if(max(Cvalue,minusopt,plusopt) == plusopt)
    return(list(C=plusT, Ap=plusAp))
else
    return(list(C=minusT, Ap=minusAp))

```

```

}

Greedy <- function(I, y, alpha, gamma, p)
{
  C <- NULL
  Ap<- rep(0, nrow(I))
  new <- iteration(C, Ap, alpha, gamma, p)
  newC <- new$C
  newAp <- new$Ap

  count <- 0
  while(!setequal(C,newC))
  {
    C <- newC
    Ap <- newAp
    new <- iteration(C,Ap, alpha, gamma, p)
    newC <- new$C
    newAp <- new$Ap
    count <- count + 1
    print(count)
  }

  return(list(C=C, Ap=Ap, count=count))
}

```

Some numerical experiments show that the proposed approximate algorithm will generate an output list of gene sets that are very close to the real MAP estimate obtained by integer linear programming. So this approximate algorithm might be a field worth further exploring.

References

- Ackermann, M., Strimmer, K. (2009). A general modular framework for gene set enrichment analysis. *BMC Bioinformatics* 10(1):47.
- Alexa, A., Rahnenfuhrer, J., Lengauer, T. (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics* 22:1600-1607.
- Arratia, R., Goldstein, L., and Gordon, L. (1990). Poisson approximations and the Chen-Stein method. *Statistical Science*, **5**, 403-424.
- Barry, W.T., Nobel, A.B., Wright, F.A. (2005). Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics* 21:1943-1949.
- Barry, W.T., Nobel, A.B., Wright, F.A. (2008). A statistical framework for testing functional categories in microarray data. *Ann. Appl. Stat.* 2:286-315.
- Bauer, S., Gagneur, J., Robinson, P.N. (2010). GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research* 38(11):3523-3532.
- Bauer, S., Robinson, P.N., Gagneur, J. (2011). Model-based gene set analysis for Bioconductor. *Bioinformatics* 27(13):1882-1883.
- Carvalho, L. E. and Lawrence, C. E. (2008). Centroid estimators for inference in high-dimensional discrete spaces. *PNAS: USA*, **105**, 3209-3214.

- Croft, D., Mundo, A.F., Haw, R., et al. (2014). The Reactome pathway knowledgebase. *Nucleic Acids Res.* 42(1):D472-D477.
- Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of the null hypothesis. *J. Amer. Statist. Assoc.*, 99, 96-104.
- Efron, B., Tibshirani, R. (2007). On testing the significance of sets of genes. *Ann. Appl. Stat.* 1: 107-129.
- Fridley, B.L., Biernacka, J.M. (2011). Gene set analysis of SNP data: benefits challenges and future directions. *European Journal of Human Genetics* 19(8):837-843.
- Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., ... & Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10), R80.
- Gillis, J., Pavlidis, P. (2013). Assessing identity redundancy and confounds in Gene Ontology annotations over time. *Bioinformatics* 29(4):476-482.
- Goeman, J.J., Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics* 23(8):980-987.
- Grossmann, S, Bauer, S, Robinson, P.N., Vingron, M. (2007). Improved detection of overrepresentation of gene-ontology annotations with parent-child analysis. *Bioinformatics* 23:3024-3031.
- Hao, L., He, Q., Wang, Z., Craven, M., Newton, M.A., and Ahlquist, P. (2013). Limited agreement of independent RNAi screens for virus-required host genes

owes more to false-negative than false-positive factors. *PLoS Computational Biology*, **9**(9): e10032355.

Joshi-Tope, G., et al. (2005). Reactome: a knowledgebase of biological pathways. *Nucleic acids research* 33(suppl 1):428-432. <http://www.reactome.org>.

Kanehisa, M., Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 28:27-30. <http://www.genome.jp/kegg>.

Khatri, P., Draghici, S. (2005). Ontological analysis of gene expression data: current tools limitations and open problems. *Bioinformatics* 21:3587-3595.

Khatri, P., Sirota, M., Butte, A.J. (2012). Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Comput. Biol.* 8:e1002375.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdottir, H., Tamayo, P., Mesirov, J. (2011). Molecular signatures database (MSigDB) 3.0. *Bioinformatics* 27:1739-1740. <http://www.broadinstitute.org/gsea/msigdb>.

Lu, Y., Rosenfeld, R., Simon, I., Nau, G.J., Bar-Joseph, Z. (2008). A probabilistic generative model for GO enrichment analysis. *Nucleic Acids Res.* 36:e109.

Maciejewski, H. (2013). Gene set analysis methods: statistical models and methodological differences. *Briefings in Bioinformatics* doi: 10.1093/bib/bbt002.

Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S.,

- Mahajan, S., May, B., Schmidt, E., Vastrik, I., Wu, G., Birney, E., Stein, L., D'Eustachio, P. (2008). Reactome knowledgebase of biological pathways and processes. *Nucleic Acids Res.* Nov 3. Epub ahead of print.
- Nesvizhskii, A. I., Aebersold, R. (2005). Interpretation of shotgun proteomic data the protein inference problem. *Molecular & Cellular Proteomics*, 4(10):1419-1440.
- Newton, M.A., He, Q., Kendzioriski, C. (2012). A model-based analysis to infer the functional content of a gene list. *Statistical Applications in Genetics and Molecular Biology* 11(2):Art 9.
- Newton, M.A., Quintana, F.A., den Boon, J.A., Sengupta, S., Ahlquist, P. (2007). Random-set methods identify distinct aspects of the enrichment signal in gene-set analysis. *Ann. Appl. Stat.* 1(1):85-106.
- Newton, M.A., Wang, Z. (2015). Multiset statistics for gene set analysis. *Annual Review of Statistics and Its Application* 2:95-111.
- R Development Core Team. (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna Austria. ISBN 3-900051-07-0. <http://www.r-project.org>.
- Robert, C.P. and Cassella, G. (2002). *Monte Carlo Statistical Methods*, New York: Springer.

- Sartor, M.A., Leikauf, G.D., Medvedovic, M. (2009). LRpath: a logistic regression approach for identifying enriched biological groups in gene expression data. *Bionformatics* 25:211-217.
- Sengupta, S, den Boon, J.A., Chen, I-H., Newton, M.A., Dahl, D.B., Chen, M., Cheng, Y-J., Westra, W.H., Chen, C-J., Hildesheim, A., Sugden, B., Ahlquist, P. (2006). Genome-wide expression profiling reveals Epstein Barr virus-associated inhibition of MHC class I expression in nasopharyngeal carcinoma. *Cancer Research* 66(16):7999-8006.
- Shojaie, A., Michailidis, G. (2009). Analysis of Gene Sets Based on the Underlying Regulatory Network. *Journal of Computational Biology* 16:407-426.
- Steen, H., Mann, M. (2004). The ABC's (and XYZ's) of peptide sequencing. *Nat. Rev. Mol. Cell Biol* 5 (9):699-711.
- Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA* 102:15545-15550.
- Supek, F., Bosnjak, M., Skunca, N., Smuc, T. (2011). REVIGO Summarizes and Visualizes Long Lists of Gene Ontology Terms. *PLoS ONE* 6(7):e21800.
- The Gene Ontology Consortium. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics* 25:25-29. <http://www.geneontology.org>.

- Tian, L., Greenberg, S.A., Kong, S.W., Altschuler, J., Kohane, I.S., et al. (2005). Discovering statistically significant pathways in expression profiling studies. *Proc Natl Acad Sci USA* 102:13544-13549.
- Vencio, R.Z., Koide, T., Gomes, S.L., de B Pereira, C.A. (2006). BayGO: Bayesian analysis of ontology term enrichment in microarray data. *BMC bioinformatics* 7(1):86.
- Wang, K., Li, M., Bucan, M. (2007). Pathway-based approaches for analysis of genomewide association studies. *The American Journal of Human Genetics* 81(6):1278-1283.
- Wang, Z., He, Q., Larget, B., Newton, M. A. (2015). A multi-functional analyzer uses parameter constraints to improve the efficiency of model-based gene-set analysis. *The Annals of Applied Statistics*, 9(1):225-246.
- Wu, D., Lim, E., Vaillant, F., Asselin-Labat, M.L., Visvader, J.E., Smyth, G.K. (2010). ROAST: rotation gene set tests for complex microarray experiments. *Bioinformatics* 26(17):2176-2182.
- Wu, D., Smyth, G.K. (2012). Camera: a competitive gene set test accounting for inter-gene correlation. *Nucleic acids research* 40(17):e133-e133.