Methods for the Analysis of Differential Composition of Gene Expression

Citation

Permanent link

Terms of Use

Share Your Story

Accessibility

Methods for the Analysis of Differential Composition of Gene Expression

A dissertation presented

by

Emmanuel Dimont

to

The Department of Biostatistics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Biostatistics

Harvard University

Cambridge, Massachusetts

December 2014

Methods for the Analysis of Differential Composition of Gene Expression

Abstract

Modern next-generation sequencing and microarray-based assays have empowered the computational biologist to measure various aspects of biological activity. This has led to the growth of genomics, transcriptomics and proteomics as fields of study of the complete set of DNA, RNA and proteins in living cells respectively. One major challenge in the analysis of this data, however, has been the widespread lack of sufficiently large sample sizes due to the high cost of new emerging technologies, making statistical inference difficult. In addition, due to the hierarchical nature of the various types of data, it is important to correctly integrate them to make meaningful biological discoveries and better informed decisions for the successful treatment of disease. In this dissertation I propose: (1) a novel method for more powerful statistical testing of differential digital gene expression between two conditions, (2) a framework for the integration of multi-level biologic data, demonstrated with the compositional analysis of gene expression and its link to promoter structure, and (3) an extension to a more complex generalized linear modeling framework, demonstrated with the compositional analysis of gene expression and its link to pathway structure adjusted for confounding covariates.

Contents

Acknowledgements

In memory of my mother, and all those unrelenting great heroes who courageously fight day and night to the very end, this righteous and honorable war to eradicate the scourge of cancer from the face of humanity.

1 edgeRun: an R package for sensitive, functionally relevant differential expression discovery using an unconditional exact test

Emmanuel Dimont[1], Jiantao Shi[1], Rory Kirchner[1], and Winston Hide[1,2,3]

[1]Department of Biostatistics, Harvard School of Public Health, 655 Huntington Avenue, Boston, MA 02115, USA, [2]Harvard Stem Cell Institute, 1350 Massachusetts Ave, Cambridge, MA 02138, USA, [3]Sheffield Institute of Translational Neuroscience, University of Sheffield, 385a Glossop Road, Sheffield, S10 2HQ, United Kingdom

Abstract

Summary: Next-generation sequencing platforms for measuring digital expression such as RNA-Seq are displacing traditional microarray-based methods in biological experiments. The detection of differentially expressed genes between groups of biological conditions has led to the development of numerous bioinformatics tools, but so far few, exploit the expanded dynamic range afforded by the new technologies. We present edgeRun, an R package that implements an unconditional exact test that is a more powerful version of the exact test in edgeR. This increase in power is especially pronounced for experiments with as few as 2 replicates per condition, for genes with low total expression and with large biological coefficient of variation. In comparison with a panel of other tools, edgeRun consistently captures functionally similar differentially expressed genes.

Availability and implementation: The package is freely available under the MIT license from CRAN (http://cran.r-project.org/web/packages/edgeRun).

Contact: edimont@mail.harvard.edu

1.1 Introduction

Next generation sequencing technologies are steadily replacing microarray-based methods, for instance transcriptome capture with RNA-Seq (Mortazavi et al, 2008) and CAGE-Seq capture for the promoterome (Kanamori-Katayama et al, 2011). All of these approaches result in digital expression data, where reads or tags are sequenced, mapped to the genome and then counted. The discrete nature of the data has required the development of new bioinformatics tools for their analysis that address discrete count data.

Once the expression has been quantified, an important next step is the statistical significance testing of differential expression between two or more groups of conditions. By the far the simplest and most popular approach reduces differential expression to a pairwise comparison of mean parameters, resulting in a fold-change measure of change and a p-value to ascertain statistical significance of the finding. To address this problem, tools such as edgeR (Robinson et al, 2010), DESeq2 (Love et al, 2014) among many others have been developed and can be applied to any experiment in which digital count data is produced.

This vast array of tool choices can be bewildering for the biologist since it is generally not clear under which conditions a tool is more appropriate than its alternates. Traditional metrics used when benchmarking methods such as the false positive rate and power are useful but limited as they are purely statistical concepts that can only be tested on simulated data. Moreover they do not help in determining to what extent methods deliver truly biologically important genes. This is a major challenge because in the vast majority of cases, we do not know what the true positives and negatives are.

In this paper, we propose a novel metric to determine the number of functionally relevant genes reported by a differential expression tool and present edgeRun, an extension of the edgeR package delivering increased power to detect true positive differences be-

tween conditions without sacrificing on the false positive rate. We show using simulations and a real data example that edgeRun is uniformly more powerful than a host of differential expression tools for small sample sizes. We also demonstrate how even though it may be less statistically powerful than DESeq2 in some simulation cases, edgeRun nonetheless produces results that are functionally more relevant.

1.2 Methods

1.2.1 edgeRun: exact unconditional testing

Assuming independent samples, Robinson et al. (2011) proposed edgeR, an R package that eliminates the nuisance mean expression parameter by conditioning on a sufficient statistic for the mean, a strategy first popularized by Fisher (1925) for the binomial distribution. This leads to a calculation of the exact p-value that does not involve the mean. The advantage of this approach is its analytic simplicity and fast computation, however a key disadvantage is that this conditioning approach loses power, especially for genes whose counts are small.

We propose an alternative more powerful approach which eliminates the nuisance mean parameter via maximizing the exact p-value over all possible values for the mean without conditioning which we call "unconditional edgeR" or edgeRun. This technique was initially proposed by Barnard (1945) for the binomial distribution. The main disadvantage of this method is the higher computational burden required for the maximization step. On the other hand, the gain in power can be significant. A thorough derivation and comparison of both methods can be found in the Supplementary Methods.

1.2.2 Benchmarking against other methods

The compcodeR Bioconductor package (Soneson, 2014) was used to benchmark the performance of edgeRun against a panel of available other tools using a combination of simulated and real datasets. edgeRun had the highest area under the curve (AUC) of all methods and it maintained a comparable false discovery rate similar to other tools. In terms of power, only DESeq2 was found to outperform edgeRun. For this reason in the next section, we perform a function-al comparison only with DESeq2. The full results are summarized in Supplementary Methods.

1.2.3 Comparing Functional Relevance

We propose to compare the genes called significant by various differential expression tools. Figure 1.1 compares the results of edgeRun and DESeq2 applied to a prostate cancer dataset (Li et al., 2008) using an FDR less than 5% cutoff. Out of the 4226 genes reported as differentially expressed, 80% were common to both tools. The highest 500 up- or down-regulated of these consensus genes by fold-change are used as a seed signature. It is reasonable to hypothesize that true differentially expressed genes uniquely reported by a differential expression tool are functionally connected to genes in the consensus group. We use GRAIL (Raychaudhuri et al, 2009) coupled with a global coexpression network COXPRESdb (Obayashi et al, 2013) to assess the relatedness between a gene and the consensus group. As expected, nearly half of these seed genes are correlated with other members of the seed group, meaning that these consensus genes form a tightly connected network. Figure 1.1 shows that edgeRun reports 6.6 times more unique DE genes, and a larger proportion of which are coexpressed with the consensus.

This means that the genes reported by edgeRun are more likely to be functionally relevant as they are more correlated with the consensus network.
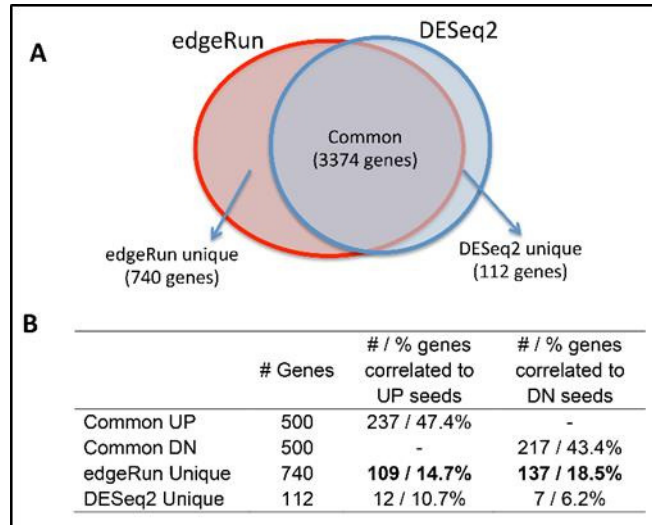
Figure 1.1: Comparing the functional relevance of genes called significantly differentially expressed by edgeRun and DESeq2

1.3 Discussion

We present edgeRun, an R package that improves on the popular package edgeR for differential digital expression by providing the capability to perform unconditional testing, resulting in more power to detect true differences in expression between two biological conditions. Even though the computational burden is increased, the power gained using this approach is significant, allowing researchers to detect more true positives, especially for cases with as few as 2 replicates per condition and for genes with low expression, all the while without sacrificing on type-I error rate control. edgeRun is simple to use, especially for users already experienced with edgeR as it is designed to interface with edgeR objects directly, taking inputs and generating output in the same format.

Acknowledgements

Funding

References

Barnard, G. A. (1945) A new test for 2x2 tables. Nature. 156:177.

Fisher, R. A. (1925) Statistical Methods for Research Workers. Oliver and Boyd, Edinburgh.

Kanamori-Katayama, M. et al. (2011) Unamplified cap analysis of gene expression on a single-molecule sequencer. Genome Re-search. 21(7):1150-9.

Li, H. et al. (2008) Determination of tag density required for digital transcriptome analysis: Application to an androgen-sensitive prostate cancer model. PNAS. 105(51).

Love, M. I, et al. (2014). Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. bioRxiv. doi: http://dx.doi.org/10.1101/002832

Mortazavi, A. et al. (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nature Methods. 5(7):621-8.

Obayashi, T. et al. (2013). COXPRESSdb: a database of comparative gene coexpression networks of eleven species of mammals. Nucleic Acids Research. 41:D1014-20.

Raychaudhuri, S., et al. (2009) Identifying relationships among genomic disease regions: predicting genes at pathogenic SNP associations and rare deletions. PLoS Genetics. 5(6):e1000534.

Robinson, M. D. et al. (2011) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bio-informatics. 26(1):139-40.

Soneson, C. (2014) compcodeR-an R package for benchmarking differential expression methods for RNA-Seq data. Bioinformatics.

2 CAGExploreR: an R package for the analysis and visualization of promoter dynamics across multiple experiments

Emmanuel Dimont[1], Oliver Hofmann[1], Shannan J. Ho Sui[1], Alistair R. R. Forrest[2,3], Hideya Kawaji[2,3,4], the FANTOM Consortium and Winston Hide[1]

[1]Department of Biostatistics, Harvard School of Public Health, 655 Huntington Avenue, Boston, MA 02115, USA, [2]RIKEN Omics Science Center, Yokohama, Kanagawa 230-0045 Japan, [3]Division of Genomic Technologies, RIKEN Center for Life Science Technologies, Yokohama, Kanagawa 230-0045, Japan and [4]RIKEN Preventive Medicine and Diagnosis Innovation Program, Wako, Saitama 351-0198, Japan

Abstract

Summary: Alternate promoter usage is an important molecular mechanism for generating RNA and protein diversity. Cap Analysis Gene Expression (CAGE) is a powerful approach for revealing the multiplicity of transcription start site (TSS) events across experiments and conditions. An understanding of the dynamics of TSS choice across these conditions requires both sensitive quantification and comparative visualization. We have developed CAGExploreR, an R package to detect and visualize changes in the use of specific TSS in wider promoter regions in the context of changes in overall gene expression when comparing different CAGE samples. These changes provide insight into the modification of transcript isoform generation and regulatory network alterations associated with cell types and conditions. CAGExploreR is based on the FANTOM5 and MPromDb promoter set definitions but can also work with user supplied regions. The package compares multiple CAGE libraries simultaneously. Supplementary Materials

describe methods in detail, and a vignette demonstrates a workflow with a real data example.

Availability and implementation: The package is freely available under the MIT license from CRAN (http://cran.r-project.org/web/packages/CAGExploreR).

Contact: edimont@mail.harvard.edu

2.1 Introduction

It has been predicted that the majority of human genes have multiple promoters. The differential use of transcription start sites (TSSs) in alternative promoters is a complementary mechanism to alternate splicing for the generation of RNA diversity that is now becoming better understood (Pal et al., 2011). Tissue specific TSS usage has been identified in mammalian genomes (Carninci et al., 2006), and alternate TSS usage has been identified in cancers when compared with normal cells (Thorsen et al., 2011), implying that promoter-specific transcription coupled with gene expression exists as hallmarks of cell state. The profound impact of switches in transcript isoform production is well recognized for its role in regulation (Trapnell et al., 2010).

Cap Analysis Gene Expression (CAGE) captures, sequences and maps capped 50 RNA tags. In addition to being a platform for measuring gene expression, it has more importantly provided molecular biologists with enhanced resolution of gene regulation by revealing the precise locations of transcription initiation events (Plessy et al., 2010). CAGE data have recently become more plentiful, thanks to the recent ENCODE (2012) and FANTOM5 (Forrest A.R.R. et al., 2014) publications.

Analysis of TSS choice provides insight into the variation of transcription factor binding, epigenetic modifications and regulatory network activation between different cell types. Although CAGE allows for the identification of individual TSS, it is more

convenient to group clusters of TSSs detected in close vicinity into 'promoter' regions. This makes CAGE an attractive platform for de novo promoter identification.

The relative transcription occurring at TSSs among alternative promoters of a gene is termed promoter composition (PC). We describe CAGExploreR, an R package that conveniently summarizes, visualizes and ranks changes in PC (also called promoter 'switching') genome-wide across different samples. The dynamics of differential PC is especially intriguing when this phenomenon leads to changes in the abundance of different transcript isoforms or protein products within the cell population under study. Figure 2.1 highlights the conceptual difference between PC and differential gene expression. Four samples, A–D, are evaluated at four color-coded promoter regions located near or within a gene. Total gene expression measured as mapped tags per million sequenced following optional library normalization with edgeR (Robinson et al., 2010) is obtained by summing the number of tags that map to the union of the gene region, all four promoters including the regions between them and dividing by effective library size. PC is measured as a proportion vector. A and B have no change in PC, but the gene is differentially expressed. A and C have no differential gene expression but there is differential PC. Finally, A and D demonstrate both differential gene expression and differential PC.

Existing bioinformatics tools that analyze count data from sequencing technologies treat genes as elementary indivisible units and usually measure differences in expression via a contrast between two groups of samples. Examples include edgeR (Robinson et al., 2010) and Cuffdiff (Trapnell et al., 2010) for differential gene expression and transcription analysis, respectively. Unlike these tools, CAGExploreR treats genes as multiunit blocks composed of promoter subunits and compares their relative expression within the gene across samples. It is not restricted to the analysis of contrasts between pairs of experiments but rather is designed to scale to any number of experiments for simultaneous comparison.

2.2 Model and Methods

Typical CAGE output consists of a BAM library file that maps sequenced tags to the genome, which CAGExploreR converts to a table of tag counts that correspond to promoter regions using either the built-in FANTOM5, MPromDb (Gupta et al., 2011) or a set of user-specified promoter definitions such as de novo identified regions. Internally, a table of counts is generated for each mapped gene, with rows corresponding to the different samples or libraries and columns corresponding to promoters, and displayed accordingly as in Figure 2.1. The promoter counts are normalized to proportions within each gene and sample. The simplest case would be a 2-by-2 table when comparing the PC across two samples for a gene with two promoters.
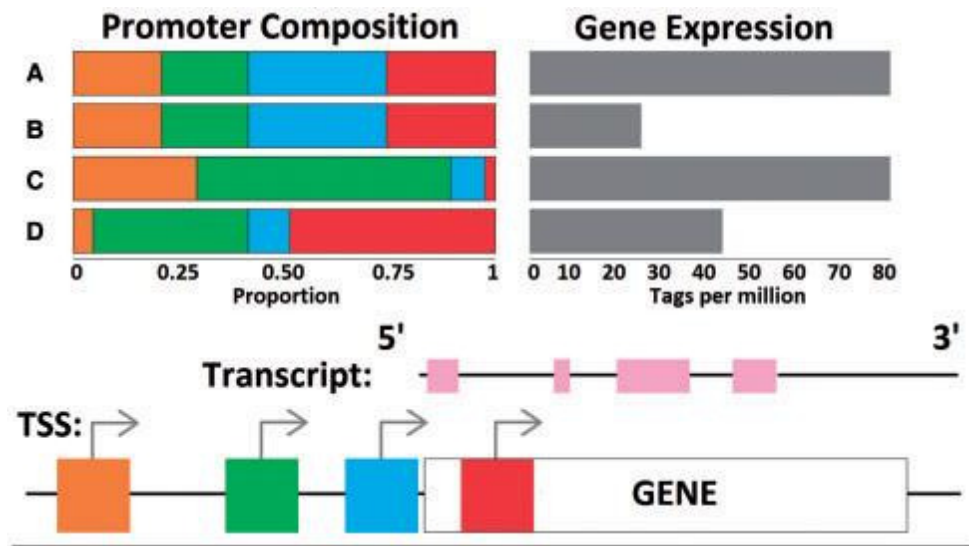

Figure 2.1: Differential PC and differential gene expression. A, B, C and D
are four arbitrary samples being compared. Gene displayed has four
promoters

Genes are assigned a proportional entropy reduction score (Theil, 1970), which ranges from 0, when PC stays constant across every sample, to 1, when every sample

11

transcribes exclusively from a unique promoter, for overall promoter switching. For each gene, the test of the null hypothesis of no differential PC corresponds to the test that the entropy reduction score is 0. P-values for this test are obtained using a Monte Carlo approach (Supplementary Methods). The switching effect size for promoter pairs is reported using the odds ratios for every nested 2-by-2 table within a gene. In addition, entropy-based measures are used to quantify the level of heterogeneity in gene expression across samples. All P-values are adjusted for multiple comparisons using the Benjamini-Hochberg method to control the false discovery rate or some other appropriate user-specified method.

CAGExploreR generates text-based and visual HTML reports and figures similar to Figure 2.1 for further analysis. When several conditions are being compared, they can be grouped together via hierarchical clustering on a gene-by-gene basis, demonstrating which conditions have similar PC profiles. This profiling helps demonstrate whether the PC across replicates clusters together within experimental conditions. Consequently, the user can assess replicate agreement at the gene level and so can gain a sense of biological variability.

2.3 Discussion

We present CAGExploreR, the R package that addresses the important task of detecting changes in PC in CAGE experiments. The method is scalable to any number of conditions and/or promoter regions for simultaneous comparison. The method is flexible and can be applied to any experiment that produces tag counts grouped by classification factors in which the detection of switching or changes in composition is of interest, e.g. gene expression switching within gene sets, pathway activity switching within regulatory and molecular networks, isoform and exon switching using RNA-Seq. To use this software

12

for any of the aforementioned applications, the user need only to change the genomic region definitions from promoter regions to other regions of interest. This work is part of the FANTOM5 project. Data download, genomic tools and co-published manuscripts have been summarized at http://fantom.gsc.riken.jp/5/top/.

Acknowledgements

Funding

References

Carninci, P. et al. (2006) Genome-wide analysis of mammalian promoter architecture and evolution. Nat. Genet., 38, 626–635.

ENCODE Consortium. (2012) An integrated encyclopedia of DNA elements in the human genome. Nature, 489, 57–74.

Forrest, A. R. R. et al. (2014) A promoter level mammalian expression atlas. Nature, http://dx.doi.org/10.1038/nature13182.

Gupta, R. et al. (2011) MPromDb update 2010: an integrated resource for annotation and visualization of mammalian gene promoters and ChIP-seq experimental data. Nucleic Acids Res., 39, D92–D97.

Pal, S. et al. (2011) Alternative transcription exceeds alternative splicing in generating the transcriptome diversity of cerebellar development. Genome Res., 21, 1260–1272.

Plessy, C. et al. (2010) Linking promoters to functional transcripts in small samples with nanoCAGE and CAGEscan. Nat. Methods, 7, 528–534.

Robinson, M. D. et al. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics, 26, 139–140.

Theil, H. (1970) On the estimation of relationships involving qualitative variables. Am. J. Sociol., 76, 103–154.

Thorsen, K. et al. (2011) Tumor-specific usage of alternative transcription start sites in colorectal cancer identified by genome-wide exon array analysis. BMC Genomics, 12, 505.

Trapnell, C. et al. (2010) Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat. Biotechnol., 28, 511–515.

3 pcmR: an R package for pathway composition modeling

Emmanuel Dimont[1], Jiantao Shi[1], Gabriel Altschuler[1], and Winston Hide[1,2,3]

[1]Department of Biostatistics, Harvard School of Public Health, 655 Huntington Avenue, Boston, MA 02115, USA, [2]Harvard Stem Cell Institute, 1350 Massachusetts Ave, Cambridge, MA 02138, USA, [3]Sheffield Institute of Translational Neuroscience, University of Sheffield, 385a Glossop Road, Sheffield, S10 2HQ, United Kingdom

Abstract

Summary: Genes act together in complex interrelationships within the context of numerous biological pathways. This information is difficult to incorporate in differential gene expression analysis, and even current pathway-based or enrichment analyses do not address the individual dynamics of each gene product as part of overall pathway activity. Differential expression analyses yield many potential targets for disease, but in the absence of relative gene activity, provide only a starting point for gene target development. We extend traditional gene expression analysis to the analysis of relative gene expression, termed composition, quantified within the context of pathways. We model the pathway-specific gene composition with respect to covariates using a generalized linear model framework. We describe pcmR, an R package that identifies the composite activity of genes-within-pathways that are most associated with disease, which provides novel context for assessing candidates for drug targeting.

Availability and implementation: The package is freely available under the MIT license from CRAN (http://cran.r-project.org/web/packages/pcmR).

Contact: edimont@mail.harvard.edu

3.1 Introduction

Gene expression analyses using microarray and next-generation sequencing platforms have become ubiquitous in modern biological research. A typical analysis involves the detection and quantification of differentially expressed genes between two or more conditions. Recently, limma (Smyth, 2005) and edgeR (Robinson et al. 2010), designed for continuous and count-based expression signals respectively, introduced the ability to fit linear models, linking the gene expression to a set of observed metadata covariates.

As no provision for gene interaction is currently available in these tools, an implicit fundamental assumption shared by these techniques is that genes act independently of one another. It is known however that genes in fact do interact with one another via biological pathways (Karlebach and Shamir, 2008). Current pathway-based analysis tools can be classified into two main categories. First, a gene list obtained from a differential gene expression analysis can be checked for enrichment in pathways, e.g. GSEA (Subramanian et al. 2005). Alternatively, genes involved in pathways can be collected and their expression summarized, producing a single value describing the overall pathways activity, e.g. PathPrint (Altschuler et al. 2013). Both of these approaches lose a significant amount of internal pathway information in favor of simplicity. A recent approach that overcomes this problem is DIRAC (Eddy et al. 2010) where the relative rank expression within pathways is com-pared between conditions. However tools do not yet exist that allow for the integration of gene expression and pathway information together with the ability of fitting linear models.

We describe pcmR, an R package that integrates gene expression data and sample metadata with biological pathway definitions, allowing for the quantification of differential relative gene expression in a pathway-specific context. The pathway-specific relative expression, termed composition, is obtained by dividing the raw gene expression value by

the sum of expression values across member genes of that pathway. Figure 3.1 illustrates the conceptual difference between gene expression and pathway composition. In addition to being context-specific, composition also has the advantage of being normalization and scaling independent.

3.2 Methods

A pathway is defined as a group of genes that work together to achieve a common goal. These lists have been curated and made available by sources such as KEGG (Kanehisa et al. 2010). A gene may be a member of multiple pathways, hence in pcmR, its composition value is made available for all its parent pathways.

Composition as a proportion is best modeled using a statistical distribution that is bounded between 0 and 1. We assume that the composition follows a Dirichlet distribution and fit a generalized linear model by maximum likelihood, linking the mean composition vector to a set of covariates. Usually one main covariate is of primary interest to the biologist, e.g. case-control status, dose of drug, etc. A multiply-corrected p-value for this main effect regression coefficient is provided using a variety of standard methods. Other covariates are generally used in more complex experimental designs and to adjust for confounding. The regression framework used is similar to the one pioneered in limma, and the Supplementary Methods summarize the key differences in greater detail. One key confounder that is always included in the model by default is the total pathway expression. This allows one to disentangle the effect of the covariates from the effect of the total pathway expression on the pathway composition.
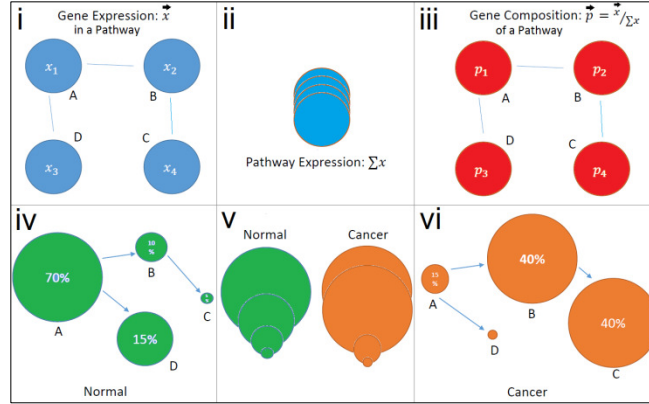
Figure 3.1: Conceptual difference between (i) gene expression, (ii) pathway expression, and (iii) gene composition. Comparing two conditions, e.g. normal v.s. cancer, both (v) pathway expression and (iv/vi) gene composition of the pathway may differ.

3.3 Example

We demonstrate pcmR and compare it to limma on a dataset of non-small cell lung cancer (NSCLC) tumors with matched controls from 60 Taiwanese non-smoking women (Lu et al. 2010). Our aim is to identify the genes that change in their pathway composition the most as opposed to their absolute expression between tumor and control conditions while adjusting for the effect of patient age. Details of the implementation are provided in the Supplementary Methods.

First we look at one pathway, the Non-Small Cell Lung Cancer pathway from KEGG. Supplementary Figure S3.1 compares the results of differential expression (limma) and differential composition (pcmR) at a 5% FDR cutoff. We note that most genes were not found to be statistically significantly differentially expressed, whereas almost all genes were differentially composed. Secondly, pcmR finds EGF as the most strongly over-composed gene in this pathway, making it a prime target for drug development. In fact, gefitinib and erlotinib that target the EGF Receptor (EGFR) have already been approved as a new standard in treating NSCLC, clearly demonstrating that the etiology of this disease in non-smoking Asian women may be directly impacted by overstimulation of the

EGF cascade. On the other hand, limma finds a downstream gene, AKT the most differentially expressed. Though it may not be the initial driver of disease at the time of patient evaluation, studies have shown that aberrant activation of AKT is one of the mechanisms of acquired resistance to targeted EGFR therapy (Fumarola et al., 2014).

Table 3.1 shows the top 3 differential composition results by magnitude across all pathways in KEGG, Reactome and Wikipathways. It is interesting to note that MASP1, involved in the lectin pathway, part of the complement system in immune response, is strongly under-composed in cancer, suggesting that an increase in its composition could be a potential target for drug development. Illustrating this hypothesis, it has been shown that lectins extracted from a mushroom, pleurotus citrinopileatus, are potent anti-tumor agents, resulting in 80% inhibition of tumor growth when administered in mice (Li et al. 2008).

Table 3.1. The 3 most differentially composed genes between NSCLC and healthy matched lung. OR: odds ratio.

Gene	Pathway Context	Source	log OR	p-value*
QDPR	Metabolic Pathways	KEGG	-81,082	0.0250
MASP1	Signaling in Immune System	Reactome	-69,989	0.0087
PIK3C3	Metabolic Pathways	KEGG	-56,302	0.0152

* FDR-adjusted

3.4 Discussion

pcmR is an R package that extends linear modeling to pathway context-specific gene composition data, allowing for the effect of composition to be disentangled from the effect of total pathway expression and other experimental covariates. We demonstrate these results in terms of their potential impact on the understanding of NSCLC. We show that pcmR provides a novel and potentially powerful analytic paradigm that is

complementary to conventional gene expression analysis; providing for a better understanding for the identification and assessment of drug targets in disease.

References

Smyth, G. K. (2005). Limma: linear models for microarray data. In Gentleman, R., Carey, V., Dudoit, S., Irizarry, R. and Huber, W. (eds.), Bioinformatics and Computational Biology Solutions using R and Bioconductor, pp. 397-420. Springer, New York.

Robinson, M. D., McCarthy, D. J. and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics, 26 (1), pp. 139-140.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Elbert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. PNAS, 102(42): 15545-50.

Altschuler, G. M., Hofmann, O., Kalatskaya, I., Payne, R., Ho Sui, S. J., Saxena, U., Krivtsov, A. V., Armstrong, A. V., Cai, T., Stein, L. and Hide, W. A. (2013) Path-printing: An integrative approach to understand the functional basis of disease. Genome Medicine. 5(7):68.

Eddy, J. A., Hood, L., Price, N. D. and Geman, D. (2010). Identifying tightly regulated and variably expressed networks by Differential Rank Conservation (DIRAC). PLoS Comp. Bio. 27;6(5).

Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M. and Hirakawa, M. (2010). KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Research, 38:D355-360.

Lu, T. P., Tsai, M. H., Lee, J. M., Hsu, C. P., Chen, P. C., Lin, C. W, Shih, J. W., Yang, P. C., Hsiao, C. K., Lai, L. C. and Chuang, E. Y. (2010). Identification of a novel biomarker, SEMA5A, for non-small cell lung carcinoma in nonsmoking women. Cancer Epidemiol. Biomarkers Prev. 19(10):2590-7.

Fumarola, C., Bonelli, M. A., Petronini, P. G. and Alfieri, R. R. (2014). Targeting PI3K/AKT/mTOR pathway in non-small cell lung cancer. Biocem. Pharmacol. 90(3):197-207.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. Nature Reviews Molecular Cell Biology. 9, pp.770-780.

Li, Y. R., Liu, Q. H., Wang, H. X. and Ng, T. B. (2008). A novel lectin with potent anti-tumor, mitogenic and HIV-1 reverse transcriptase inhibitory activities from the edible mushroom Pleurotus citrinopileatus. Biochimica et Biophysica Acta. 1780(1):51-57.

Supplementary Materials for edgeRun

S1.1 Data Setup

One simple experimental setup in computational biology is one in which the gene expression from 2 different biological conditions (X and Y) is to be compared to one another. We assume that we have n_1 and n_2 replicates of each condition respectively. Next-generation sequencing technologies generate reads or tags that are mapped to a reference genome. The number of tags that map to various genomic loci of interest (e.g. genes) is a measure of that feature's expression. We represent this data in a table in which rows correspond to different genomic loci (e.g. genes), and columns correspond to the biological samples.

		Condition X				Condition Y		
Gene 1	x_{11}	x_{12}	$\cdots$	x_{1n_1}	y_{11}	y_{12}	$\cdots$	y_{1n_2}
Gene 2	x_{21}	x_{22}	$\cdots$	x_{2n_1}	y_{21}	y_{22}	$\cdots$	y_{2n_2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Gene G	x_{G1}	x_{G2}	$\cdots$	x_{Gn_1}	y_{G1}	y_{G2}	$\cdots$	y_{Gn_2}

S1.2 Distribution Assumptions

The simplest model for an integer count variable X that is not assumed to be bounded above is the Poisson distribution that has the following probability mass function (p.m.f.):

$$X \sim \text{Poisson}(\mu)$$

$$P(X = x|\mu) = \frac{\mu^x e^{-\mu}}{x!}$$

where $\text{E}\,[X] = \mu$ and $\text{Var}\,[X] = \mu$. One major problem with the Poisson model is the strong assumption that the variance is equal to the mean. In practice this assumption rarely holds. To obtain a model that allows for more variation than that assumed by the Poisson, we can assume that the Poisson mean parameter has a distribution of its own rather than being a fixed constant. The simplest distribution for a non-negative

continuous mean parameter is the Gamma distribution with the probability density function (p.d.f.) given below. We now have have the hierarchical model:

$$X|M \sim \text{Poisson}(M) \text{ and } M \sim \text{Gamma}(\alpha, \beta)$$

$$f_M(m|\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} m^{\alpha-1} e^{-\frac{1}{\beta}m}$$

This is also called Gamma *mixing*. Here, M is the random variable and m is the realization of the Poisson mean parameter. For the Gamma distribution, α is the shape and β is the scale parameter respectively. As a consequence, $\text{E}[M] = \alpha\beta$ and $\text{Var}[M] = \alpha\beta^2$. If we let $\alpha = 1/\phi$ and $\beta = \phi\mu$, we then have $\text{E}[M] = \mu$ and $\text{Var}[M] = \phi\mu^2$. With repeated sampling on average, the mean parameter will be μ, but it will have some variation that depends on a new parameter ϕ which we call the *dispersion*. The marginal distribution of X under this structure results in the *negative binomial* distribution with p.m.f.:

$$X \sim \text{NegBin}(\mu, \phi)$$

$$P(X = x|\mu, \phi) = \binom{x + \phi^{-1} - 1}{x} \left(\frac{1}{1 + \phi\mu}\right)^{\phi^{-1}} \left(\frac{\phi\mu}{1 + \phi\mu}\right)^x$$

$$\text{E}[X] = \mu \text{ and } \text{Var}[X] = \mu + \phi\mu^2$$

Using this model, X allows for extra-Poisson variation which we call *overdispersion*, using the parameter ϕ.

With this distribution in place, we now make the following assumptions concerning our data:

$$X_{gr} \overset{iid}{\sim} \text{NegBin}(\mu_g, \phi_g)$$

$$Y_{gr} \overset{iid}{\sim} \text{NegBin}(\mu'_g, \phi_g)$$

Each gene is allowed to have a separate mean and/or dispersion, and X and Y can have different means as well. Independence of samples is assumed.

Note: In practice, each sample will have a different number of total tags that are generated and successfully mapped, i.e. the *sequencing depth* or *library size*.

$$X_{gr} \overset{iid}{\sim} \text{NegBin}(k_r\lambda_g, \phi_g)$$

As a result, μ is equal to the product of k, the library size of the sample and λ, the relative expression of the gene. To make all of the samples comparable requires a procedure called *normalization*. It is beyond

the scope of this paper to discuss *normalization* methods, and the choice of any one method does not affect our discussion that follows. We proceed by assuming that X and Y are transformed into *pseudo-counts* in all further analyses as is standard procedure. For this reason and for clarity, we revert back to our original notation.

S1.3   Hypothesis of Interest

We are interested in testing the G hypotheses of the form:

$$H_0 : \mu_g = \mu'_g$$

$$H_0 : \mu_g \neq \mu'_g$$

Since sample sizes are typically small when sequencing technologies are expensive, we proceed with an *exact* test that makes no asymptotic assumptions. The *p-value* is defined as the probability of getting something as or more extreme as the observed data under the null hypothesis. Obtaining the probability of the observed data is straightforward:

$$P(\text{observed}_g) = P(X_{g1} = x_{g1} \cap \cdots \cap X_{gn_1} = x_{gn_1} \cap Y_{g1} = y_{g1} \cap \cdots \cap Y_{gn_2} = y_{gn_2})$$

The challenge is to identify which data points are as or more extreme than those observed. The trivial approach would require the enumeration of $n_1 n_2$ variables, but this is not feasible. For a particular gene, let $S_{g1} = \sum X_g$ and $S_{g2} = \sum Y_g$. It becomes much easier to work with the probabilities associated with these sums rather than the original random variables. The sum of iid negative binomial random variables is also negative binomial:

$$S_{g1} \sim \text{NegBin}(n_1 \mu_g, \phi_g / n_1)$$

$$S_{g2} \sim \text{NegBin}(n_2 \mu_g, \phi_g / n_2)$$

We can now calculate the probabilities of various (S_1, S_2) independent pairs as follows:

$$P(s_{g1}, s_{g2} | \mu_g) = P(S_{g1} = s_{g1} \cap S_{g2} = s_{g2} | \mu_g) =$$

$$\binom{s_{g1} + n_1/\phi_g - 1}{s_{g1}} \binom{s_{g2} + n_2/\phi_g - 1}{s_{g2}} \left(\frac{1}{1 + \phi_g \mu_g} \right)^{\frac{n_1 + n_2}{\phi_g}} \left(\frac{\phi_g \mu_g}{1 + \phi_g \mu_g} \right)^{s_{g1} + s_{g2}}$$

We can see that any p-value obtained from this formula will depend on the choice of ϕ_g and μ_g. However,

our hypothesis of interest does not depend on these parameters. Under the null, we only assume that the two means are equal to one another and we make no statements about what that value is. As a result, both ϕ_g and μ_g are *nuisance* parameters that need to be eliminated. There is no generally accepted way to eliminate ϕ_g, and so in all ensuing discussion, ϕ_g is assumed to be a known constant. On the other hand there are two ways to perform the elimination of μ_g, one resulting in a *conditional* exact test (CET) and the other, an *unconditional* exact test (UET).

S1.4   edgeR: Conditional Exact Test (CET)

The popular `edgeR` Bioconductor package (Robinson et al., 2010) implements an exact test that eliminates the nuisance mean parameter by conditioning on a sufficient statistic for the mean. This technique was first proposed by Fisher (1925) to eliminate the nuisance parameter when testing equality of parameters in the *binomial* distribution, resulting in Fisher's Exact Test. The sufficient statistic is the total sum $S_g = S_{g1} + S_{g2}$. The p-value is calculated based on the following conditional probability. **Reminder:** The parameter ϕ_g, whether it is estimated from the data or not, is assumed to be known.

$$P(S_{g1} = s_{g1} \cap S_{g2} = s_{g2} | S_g) = \frac{P(S_{g1} \cap S_{g2} \cap S_g)}{P(S_g)} = \frac{P(S_{g1} = s_{g1} \cap S_{g2} = s_g - s_{g1})}{P(S_g)}$$

$$= \frac{\binom{s_{g1} + (n_1/\phi_g) - 1}{s_{g1}} \binom{s_g - s_{g1} + (n_2/\phi_g) - 1}{s_g - s_{g1}}}{\binom{s_g + ((n_1 + n_2)/\phi_g) - 1}{s_g}} = CP(s_{g1})$$

We can see that this expression no longer depends on μ. S_{g1} is taken as a statistic to determine values which are as or more extreme as those observed. The *conditional* two-sided p-value can then be calculated as follows:

$$p_{\texttt{edgeR}} = 2 \times \min \left\{ \sum_{k=0}^{\widehat{s_{g1}}} CP(k), \sum_{k=\widehat{s_{g1}}}^{\widehat{s_g}} CP(k) \right\}$$

This is the default method used in the `exactTest` function in `edgeR`. It is called the *double-tail* method because it calculates both tails of the conditional probability and doubles the smallest of the two.

The p-value from the conditional exact test is very easy to compute, however this approach suffers from a loss in power due to the fact that conditioning is performed. The smaller the value of S_g (e.g. few replicates are available and/or the gene has low expression levels), the greater the loss in power. This happens because a conditional probability by definition restricts the original sample space to a smaller subset. If S_g is small, then $CP(s_{g1})$ may have a small finite number of values that it can take, but since it must sum to 1, the

values are all larger, leading to a larger p-value.

S1.5 edgeRun: Unconditional Exact Test (UET)

We can eliminate the loss in power due to conditioning by performing an alternative exact test that eliminates the nuisance mean parameter in a different way. Instead of working with the conditional probability $CP(s_{g1})$, we use the unconditional probability of the observed data $P(s_{g1}, s_{g2})$ as a basis for calculating a p-value. One challenge involves determining what set of data points s_{g1} and s_{g2} correspond to values as or more extreme than those observed. We use the *pooled z*-statistic T for this purpose:

$$T(s_{g1}, s_{g2}) = \frac{s_{g1} - s_{g2}}{\sqrt{(\bar{\mu}_g + \phi_g \bar{\mu}_g^2)(n_1 + n_2)}} \text{ where } \bar{\mu}_g = \frac{s_{g1} + s_{g2}}{n_1 + n_2}$$

This statistic is adapted from the z-statistic used for a *z-test* for testing the equality of two normal means assuming equal variances with some minor modifications. The numerator is simply the difference in the sums of the two groups, while the denominator is the standard error of this difference, assuming independent negative binomially distributed data with an estimated common mean $\bar{\mu}_g$. The observed sums yield the value $T_0 = T(\widehat{s_{g1}}, \widehat{s_{g2}})$. Larger values of the statistic $|T|$ relative to $|T_0|$ (or alternatively T^2 relative to T_0^2) correspond to increasing evidence of deviation from what we get using the observed data.

We can calculate T for every combination of (s_{g1}, s_{g2}) and compare it with T_0 to determine if that combination is as or more extreme as the one observed. For those values of (s_{g1}, s_{g2}) that are as or more extreme as those observed, we calculate their unconditional probability P, and then sum these probabilities across all such points. This process is then repeated for different values of μ_g until a supremum is obtained, i.e. take the largest sum of P as the two-sided p-value:

$$p_{\texttt{edgeRun}} = \sup_{\mu_g} \left\{ \sum_{\left\{(j,k): T(j,k)^2 \geq T_0^2\right\}} P(j, k | \mu_g) \right\}$$

Reminder: Just like in the CET, the parameter ϕ_g present in P, whether it is estimated from the data or not, is assumed to be known.

This approach of eliminating the nuisance parameter by maximizing over it can be traced back to Barnard (1945) in which he proposed a similar test for binomially distributed data as an alternative to Fisher's exact test. Barnard showed that this technique yields a test that is more powerful than Fisher's.

The main disadvantage of this approach however is the difficulty in its implementation. Unlike in the conditional p-value calculation, the number of terms to be summed in this expression is infinite. This,

together with the fact that there is no closed-form solution to the supremum operator, requires the use of numerical techniques to obtain this p-value which is much more computationally intensive.

S1.6 Numerical Implementation of the UET

S1.6.1 Approximating the Infinite Sum

Ignoring the supremum operation for now, one major challenge in implementing the UET is performing the summation of probabilities across the relevant space of s_{g1} and s_{g2}. One could propose to calculate values of T for all such points, determine which points are relevant, and then perform the summation over these points. This is impractical since the space over which this needs to be done is the upper right quadrant of integers which is countably infinite. The space in question is depicted in the figure below. The red diagonal
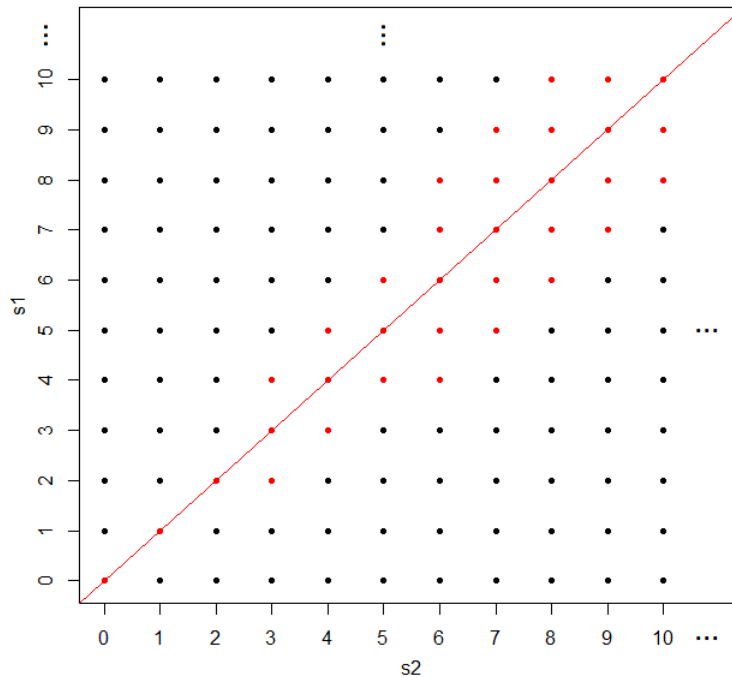
Supplementary Figure S1.1: The Cartesian space of (s_{g1}, s_{g2}). Red points: $T < T_0$

line corresponds to the null case where $s_{g1} = s_{g2}$. Points in red are those for which $T^2 < T_0^2$ and so belong to the null as well. Points in black are those for which $T^2 \geq T_0^2$ and are those over which we want to perform the summation. Let A_0 be the total probability mass occupied by the null points depicted in red and A_1 its complement. The probability mass of black points is then $A_1 = 1 - A_0$, and the problem is simplified if we can find a way to characterize the red area. This summation still requires summing over an infinite array

over 2 dimensions. We can reduce this to an infinite summation over just 1 dimension as follows:

$$A_1 = \sum_{\{(j,k):T(j,k)^2 \geq T_0^2\}} P(j,k|\mu_g)$$

$$= \lim_{K \to \infty} \sum_{s_{g2}=0}^{K} \left\{ 1 - \left[F(s_{g1}^{U|s_{g2}}) - F(s_{g1}^{L-1|s_{g2}}) \right] \right\} P(S_{g2} = s_{g2})$$

In the above expression, $P(S_{g2})$ is the marginal p.m.f. of S_{g2} with mean $n_2\mu_g$ and dispersion ϕ_g/n_2 and $F(\cdot)$ is the c.d.f. of S_{g1} with mean $n_1\mu_g$ and dispersion ϕ_g/n_1. s_{g1}^U and s_{g1}^L correspond to the values of s_{g1} which are the edge points on vertical slices of the red probability mass area for a given value of s_{g2}. In essence what the expression above is doing is calculating the complementary mass of the red area (i.e. the black area) by taking vertical slices across values of s_{g2}.

The next challenge is finding these edge points that are used in the c.d.f. which we find by solving the inequality $T^2 < T_0^2$.

$$\text{Let:} \quad w_1 = 1 + \frac{\phi_g T_0^2}{n_1 + n_2} \quad \text{and} \quad w_2 = 1 - \frac{\phi_g T_0^2}{n_1 + n_2}$$

After some algebraic manipulation, we get the following quadratic inequality:

$$a s_{g1}^2 + b s_{g1} + c < 0 \quad \text{where:}$$

$$a = w_2 \quad b = -(2w_1 s_{g2} + T_0^2) \quad c = s_{g2}(w_2 s_{g2} - T_0^2)$$

The solutions are obtained by applying the quadratic formula. Let L be the smallest and U be the largest of the two solutions. Because $w_2 > 0$ always, the parabola is convex, and since the solutions must be integers, we apply the appropriate floor and ceiling operators.

$$s_{g1}^{L|s_{g2}} = \max(\lceil L \rceil, 0)$$

$$s_{g1}^{U|s_{g2}} = \max(\lfloor U \rfloor, 0)$$

Finally, we take the maximum of the solution with 0 to avoid solutions which are negative. It should be noted that these solutions depend on the specific value of s_{g2}.

We approximate the infinite sum in A_1 by choosing an upper bound K which is sufficiently large. Larger values of K increase computing time but increase the accuracy of the p-value. The figure below shows how

the level of accuracy of the p-value increases with K. Two groups with 3, 5 and 10 samples per group are simulated under the null with $s_{g1} = s_{g2} = 10,000$ for various levels of ϕ. Since this is a null scenario, we expect a true p-value of 1, which as expected, is evident as the asymptote for every curve in the plot. In addition to the p-value, the y-axis in this figure also refers to the goodness of the approximation. Larger values of K are necessary for a good p-value approximation for increasing ϕ and decreasing sample size with fixed s_g (i.e. increasing observed average tag counts per replicate).

By default, `edgeRun` takes $K = 50,000$ as a compromise between accuracy and speed, but this can be adjusted by the user. As seen from the figure, this value of K yields approximately 80% accuracy for a wide range of data scenarios. On an Intel® Core-i7 4700HQ 2.4GHz processor, a computation with 20,000 genes takes approximately 15 minutes to complete.
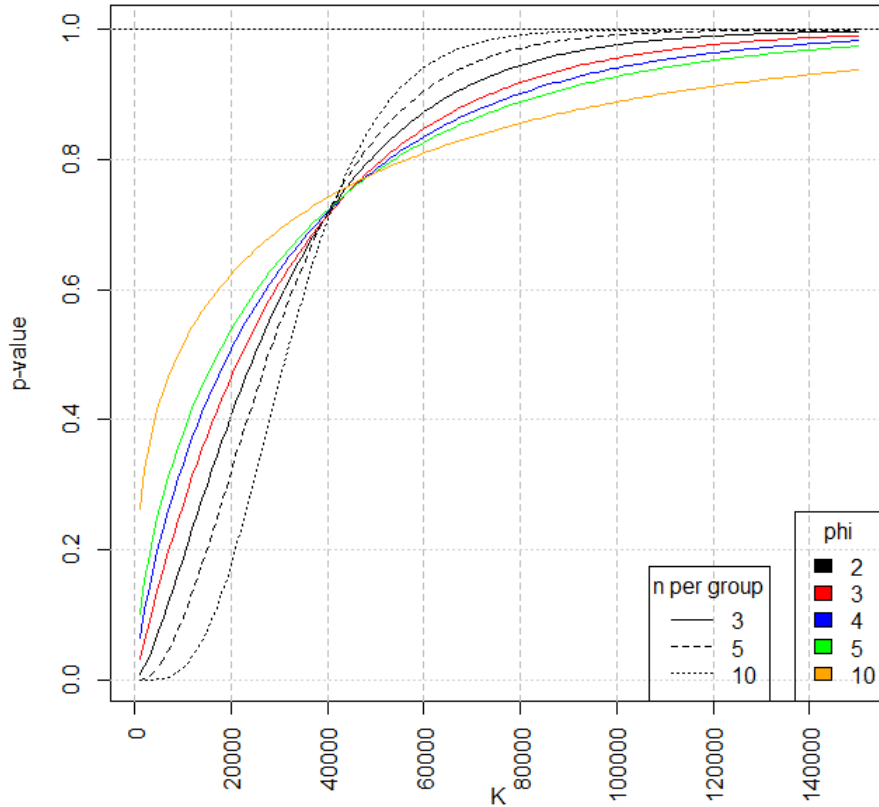
Supplementary Figure S1.2: Relationship between p-value accuracy and K for various levels of ϕ and number of replicates per group with $s_{g1} = s_{g2} = 10,000$

S1.6.2 Approximating the Supremum

In the previous section, we find an approximation to A_1. The next challenge is to find the value of μ_g that will maximize the value of A_1 which we call μ^*, i.e.

$$\mu_g^* = \arg\max_{\mu_g} A_1$$

To solve this problem, we attempt to find a relationship between the μ^* and various factors. To do this, data was simulated under the non-null case for arbitrarily chosen $s_{g1} = 1000$ and $s_{g2} = 1$ and $n_1 = n_2 = 2$ across a range of ϕ_g which were chosen over a range 0-20 in fine increments of 0.01. For each case, μ^* was obtained by evaluating A_1 over an iterative logarithmic grid of μ_g values until an accuracy within ± 1 was obtained. These values are plotted in the figure below.
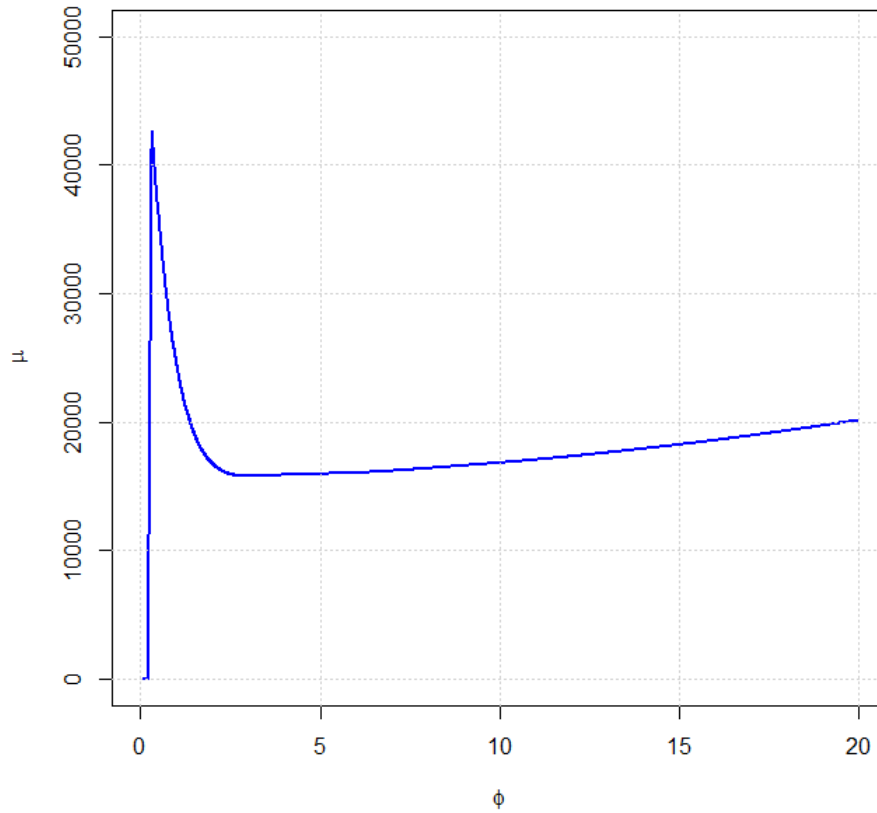
Supplementary Figure S1.3: Relationship between μ^* (μ_g that maximizes A_1), v.s. ϕ_g for the case $s_{g1} = 1000, s_{g2} = 1$ and $n_1 = n_2 = 2$

This relationship is characterized and stored in `edgeRun` as follows:

$$\mu^*(\phi_g) = \begin{cases} 22.98 & : & 0 \le \phi_g < 0.20 \\ -74{,}939 + 374{,}809\phi_g & : & 0.20 \le \phi_g < 0.32 \\ \sum_{i=0}^{11} \beta_i \phi_g^i & : & 0.32 \le \phi_g < 20 \\ \sum_{i=0}^{11} \beta_i 20^i & : & 20 \le \phi_g \end{cases}$$

Very small values of $\phi_g < 0.2$ were simulated using the Poisson distribution since the negative binomial routines were found to be unstable in this range. A discontinuity was detected in the from 0.2-0.32 and it was very difficult to obtain a result in this range, hence a linear interpolation was performed to join to the upper range of ϕ_g. An 11th order polynomial fit was found to approximate the solution for $\phi_g > 0.32$ very well with an R^2 very close to 100%. The β coefficients of the polynomial approximation are stored in the `fit.2` object. For the remaining range of $\phi_g > 20$, since the curve appears to level off, the accuracy of using the exact value of ϕ_g v.s. the upper threshold of 20 in the polynomial fit was not significantly altered (data no shown).

It was found that varying the values of n_1 and n_2 affected this relationship only by the corresponding scaling effect on μ_g and ϕ_g. As a result, the case of $n_1 = n_2 = 2$ is taken as a reference and all data for other n is linearly scaled to the $n = 2$ case for the purposes of obtaining μ^*. Finally, values of s_{g1} and s_{g2} were chosen to lie in the non-null space, but the numbers were chosen arbitrarily. We choose values in the non-null case because we want to be conservative specifically for data that is non-null, since values that are closer to the null are more likely to be non-significant anyway. We find that the solution for μ^* does not significantly alter if other values for s are used (data not shown). Once again, this approximation is a compromise between accuracy and speed.

S1.7   Simulation Studies using compcodeR

We used the Bioconductor package `compcodeR` (Soneson, C., 2014) to benchmark the performance of `edgeRun` against a panel of 26 other differential expression tools using various parameters. We used the B_{625}^{625} simulated dataset with 2 replicates per condition in which 10% of a total of 12500 genes were differentially expressed (625 genes in each condition). More details on how these random testing datasets are generated can be found in Soneson and Delorenzi (2013). The figures below show the performance of `edgeRun` in terms of area under the curve (AUC) and the true positive rate (TPR). The blue line corresponds to the value attained by `edgeRun`.

We find that `edgeRun` has the highest AUC of all methods tested, meaning that on the average, choosing across a range of cutoff values of the false discovery rate (FDR) to determine which genes to call as differ-

entially expressed (e.g. call a gene differentially expressed if it's adjusted p-value < 0.05), `edgeRun` has the most optimal combination of sensitivity (true positive rate) and specificity (true negative rate).
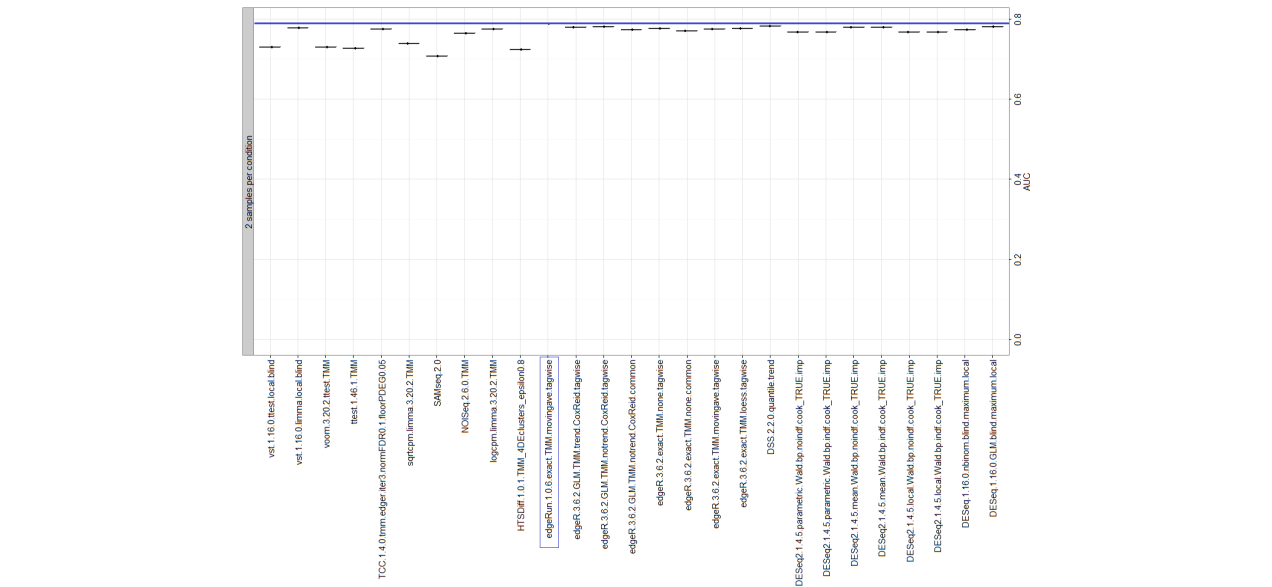
Supplementary Figure S1.4: Area under the curve (AUC) comparison between `edgeRun` (blue) and 26 other tools/parameter combinations using B_{625}^{625} simulation dataset
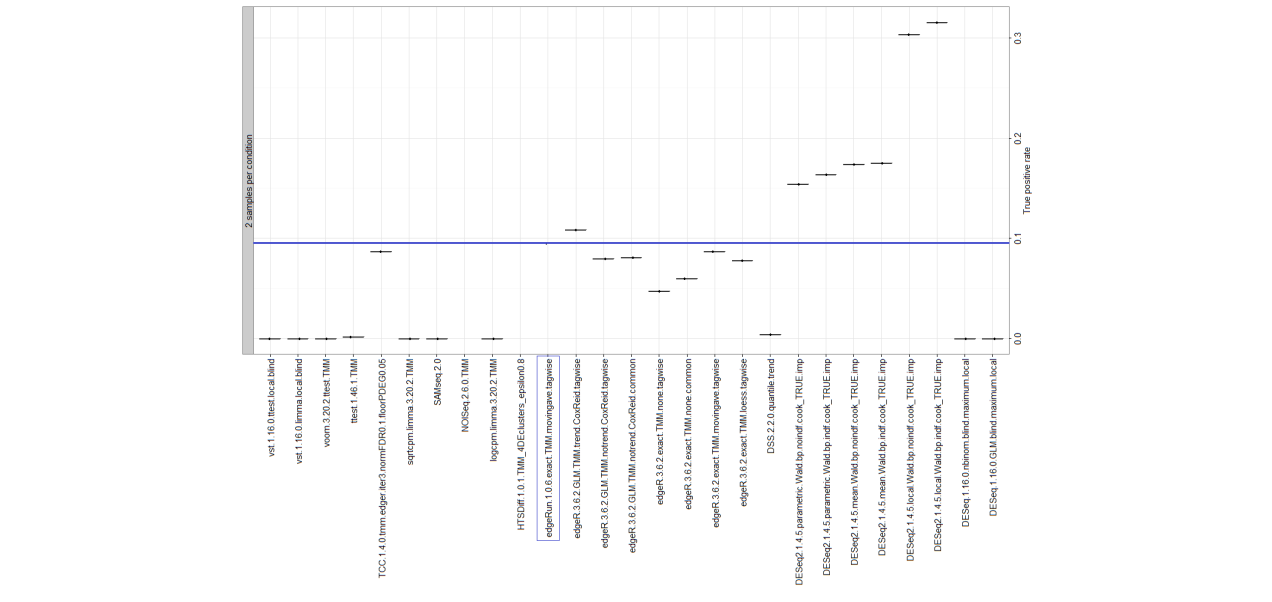
Supplementary Figure S1.5: True positive rate (TPR) comparison at 5% cutoff between `edgeRun` (blue) and 26 other tools/parameter combinations using B_{625}^{625} simulation dataset

Typically however, instead of looking across a range of cutoff values, in practice a single cutoff value is used to determine differentially expressed genes, one of the most popular being the 5% cutoff for adjusted

p-values. In the next figure we can see that the sensitivity or true positive rate is high, but no longer the best across all tools compared. We note that `DESeq2` (Love, M.I. et al. 2014) is by far the most competitive, with sensitivities reaching nearly 3 times that of `edgeRun`. There is always a trade-off between sensitivity and specificity however, and in the figure below we can see how the false discovery rate is higher for those `DESeq2` settings that yielded the highest TPR.
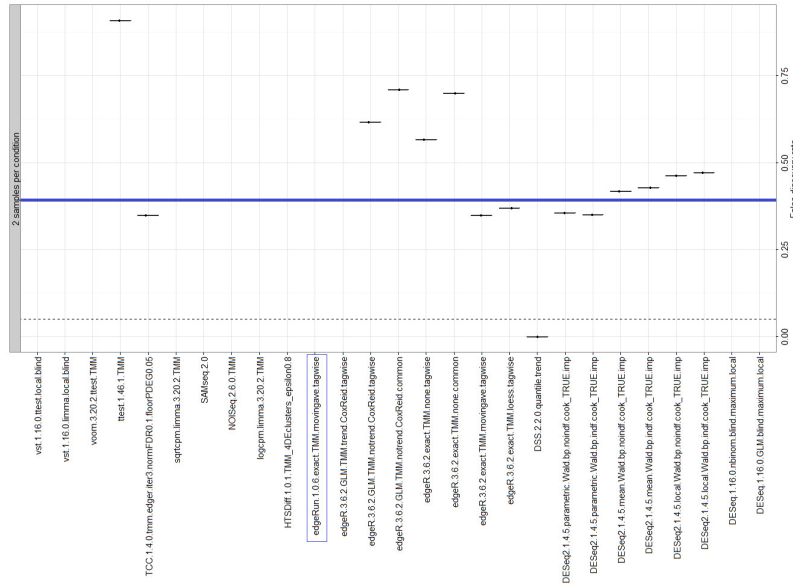
Supplementary Figure S1.6: False discovery rate (FDR) comparison at 5% cutoff between `edgeRun` (blue) and 26 other tools/parameter combinations using B_{625}^{625} simulation dataset

We concede that `DESeq2` can be more powerful than `edgeRun`, however in the next several sections we propose a new approach of comparing these tools from a functional relevance perspective. As a result, from now on we only focus on the comparison between `edgeRun` and `DESeq2`.

S1.8   A Real-Data Example

Having shown that `edgeRun` performs well on simulated data, we next proceed to applying it to some real experimental data. For this purpose we use the prostate cancer model by Li et al. (2008), the same dataset used by the authors of `edgeR` (Robinson et al. 2010) to demonstrate its functionality. The dataset consists of a two sample comparison with $n_1 = 3$ and $n_2 = 4$.

S1.8.1 edgeRun v.s. edgeR

We first compare `edgeRun` to `edgeR`. We can see in the MA plot below that there is wide overlap in the genes called significant at a 5% FDR cutoff between the two methods. We can also see that `edgeRun` is more sensitive than `edgeR` since it is able to call genes significant that for a fixed level of expression, have lower fold change. We see this increased power across the whole range of expression levels, but especially so for more lowly expressed genes as expected. It should be noted that `edgeRun` calls as significant all those genes already called by `edgeR`, thus confirming in practice the expected theoretical gains in power due to the UET over the CET.
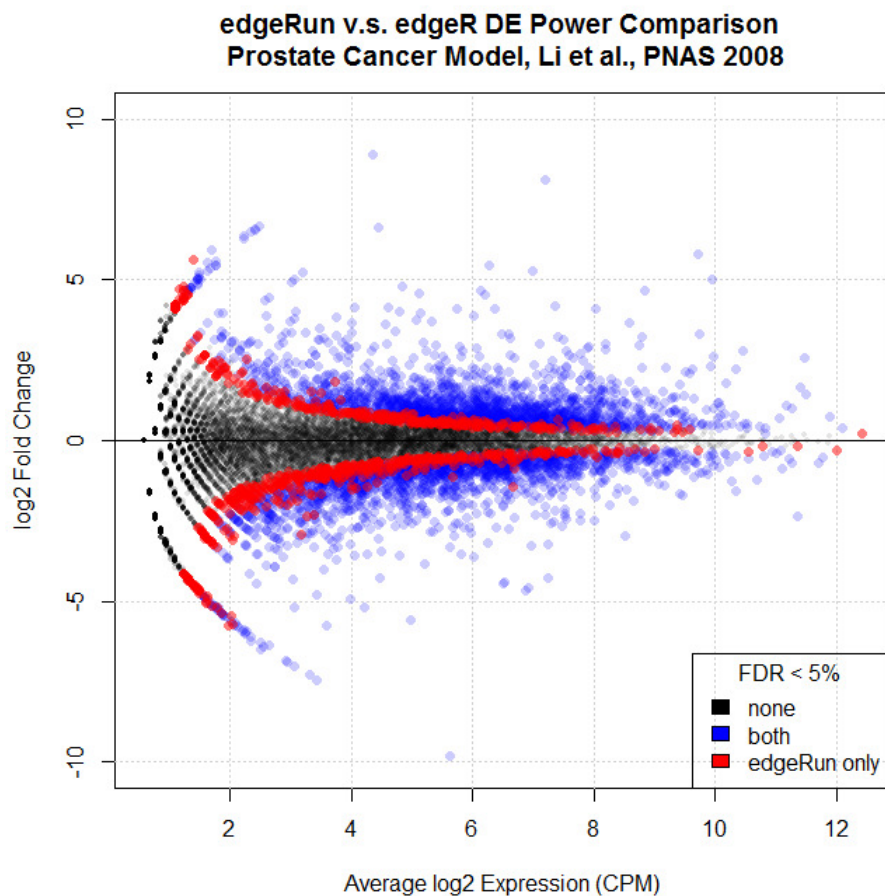
Supplementary Figure S1.7: `edgeRun` (UET) is uniformly more powerful than `edgeR` (CET)

S1.8.2 edgeRun v.s. DESeq2

Previously we have shown in simulations that `DESeq2` is the only other tool that was shown to be more powerful than `edgeRun`. We now perform a comparison between the `edgeR` family (`edgeRun` and `edgeR`)

34

and the `DESeq` family (`DESeq` and `DESeq2`) on the same dataset, the results of which can be seen in the MA plot below. Once again we can see that there is wide agreement between all 4 tools, however we find interesting differences in what classes of genes the different families call as significant. The `DESeq` family is more sensitive at detecting genes with higher expression levels, whereas `edgeRun` is more sensitive at the lower end of the gene expression spectrum.
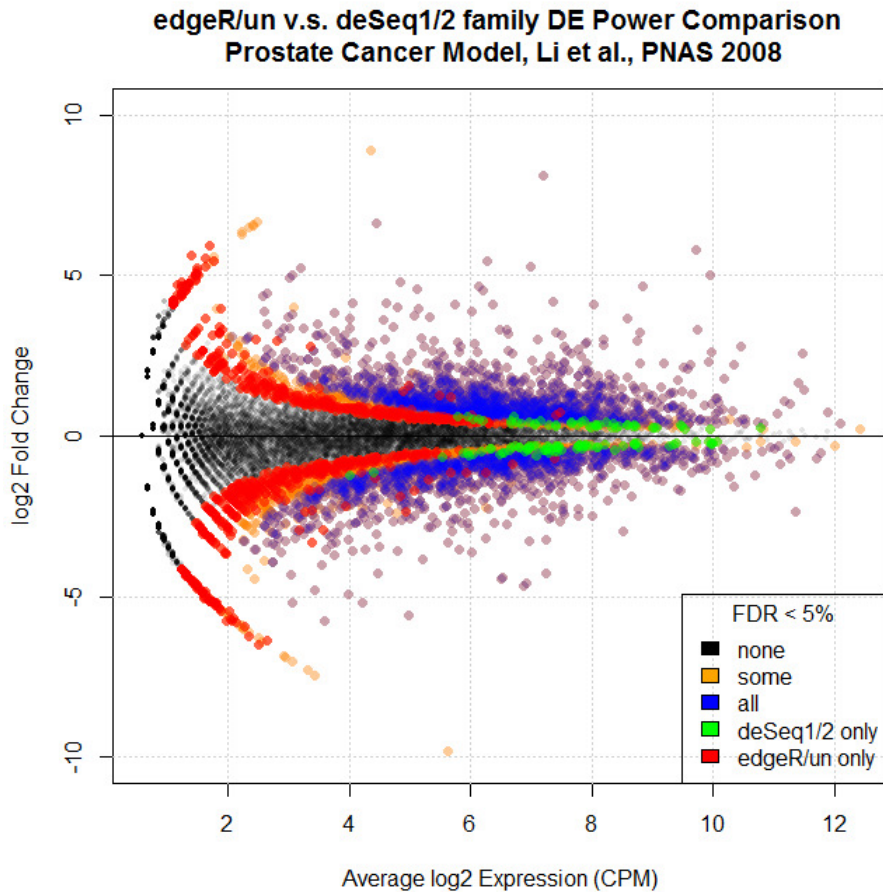
Supplementary Figure S1.8: `edgeRun` is more sensitive for lowly, whereas `DESeq2` is more sensitive for more highly expressed genes

S1.9 Assessing Functional Relevance

As described in the main manuscript, Out of the 4226 genes reported as differentially expressed in a prostate cancer dataset, 80% were common to both `edgeRun` and `DESeq2`. We define these shared genes as consensus genes, which are assumed to be truly differentially expressed (DE). Although `edgeRun` identified 6 times more DE genes compared to `DESeq2` (740 vs. 112), we cannot simply say all the called genes are true DE

genes. But its reasonable to hypothesize that true DE genes are functionally related to consensus genes. We thus used GRAIL (Raychaudhuri et al., 2009) coupled with a global coexpression networks COXPRESdb (Obayashi et al., 2013) to assess the significance of functional relatedness between a gene and the consensus group. GRAIL builds coexpression subnetworks using provided seed genes, and then assesses the relatedness between a query gene and seed networks. To avoid the heterogeneity of subnetworks, we split the consensus genes into up-regulated and down-regulated groups, and only select top 500 genes (by fold change) for each group. Using a cutoff of false discovery rate (FDR) of 5%, more than 40% of genes in each consensus group are significantly correlated to other genes in the same group, suggesting that genes in the consensus group form tightly connected subnetworks. By checking the genes uniquely called by two tools, we can see the genes reported by `edgeRun` are more likely to be functionally relevant (14.7% vs. 10.7% with Up seeds, and 18.5% vs. 6.2% with Down seeds).

S1.10   References

Barnard, G.A. (1945). A new test for 2x2 tables. *Nature.* 156:177.

Fisher, R.A. (1925). Statistical Methods for Research Workers. Oliver and Boyd, Edinburgh.

Li, H., et al. (2008). Determination of tag density required for digital transcriptome analysis: application to androgen-sensitive prostate cancer model. *PNAS* 105(51):20179-84.

Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *bioRxiv* http://dx.doi.org/10.1101/002832

Obayashi,T. et al. (2013) COXPRESdb: a database of comparative gene coexpression networks of eleven species for mammals. Nucleic Acids Res., 41, D101420.

Raychaudhuri,S. et al. (2009) Identifying relationships among genomic disease regions: predicting genes at pathogenic SNP associations and rare deletions. *PLoS Genetics*, 5, e1000534.

Robinson, M.D., McCarthy, D.J., Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26, 139-140.

Soneson, C. and Delorenzi M. (2013). A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics* 14:91.

Soneson, C. (2014). compcodeR: an R package for benchmarking differential expression methods for RNA-seq data. *Bioinformatics* May 9. pii: btu324.

Supplementary Materials for CAGExploreR

S2.1   Model and Methods

S2.1.1   CAGE-seq data overview

CAGE-seq data consists of short (~25bp) tags that correspond to the 5' ends of mRNA. These are mapped to the genome, and their location specifies the transcription start site (TSS). The figure below shows a typical setup: a gene region with CAGE tags colored in red that map to this region. Areas where the
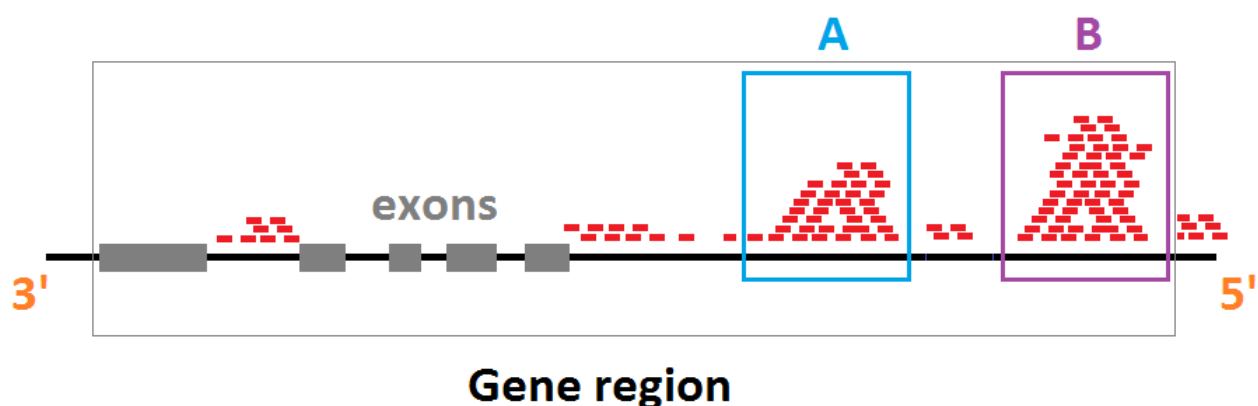
Supplementary Figure S2.1: CAGE-seq tags mapping to a gene region in one experiment with two promoter regions A and B

number of mapped tags is significantly enriched over the background (e.g. regions A and B above) are called promoter regions. Tags landing outside such promoter regions may be noise or other transcription events. Our primary goal however is to quantify the dynamics of transcription between different samples or conditions and between promoter regions within genes. It is beyond the scope of **CAGExploreR** to empirically define gene and promoter regions. For this purpose, the following pre-defined promoter regions come included with the package:

- **FANTOM5**. These are regions from the FANTOM5 Consortium (2013) and are based on CAGE-seq tag clusters determined from a comprehensive assessment of over 1000 conditions using CAGE-seq

performed at RIKEN as part of the **F**unctional **An**notation **o**f the **M**ammalian Genome international consortium. We include only those promoter regions that associate with genes with at least 2 promoters (multi-promoter genes). Only genes with valid HGNC names are included. NOTE: regions are based on hg19 coordinates.

- **MPromDB**. These are regions from Gupta et al. (2011) available online at `http://mpromdb.wistar.upenn.edu/`. The **M**ammalian **Prom**oter **D**atab**a**se is a curated database that strives to annotate gene promoters identified from ChIP-Seq experiment results. According to the website, the latest version of MPromDb is based on 507 million uniquely aligned RNA Pol-II ChIP-seq reads from 26 different datasets. We include only those promoters that are associated with genes with at least 2 promoters (multi-promoter genes). Only genes with valid HGNC names are included. NOTE: MPromDB regions were converted from hg18 to hg19 coordinates using the liftover tool at `http://genome.ucsc.edu/cgi-bin/hgLiftOver`, and gene names were converted to HGNC names whenever possible.

As long as the formatting is compatible, users can supply their own promoter regions of interest as well. Updates of these or other databases can also be easily incorporated for use with **CAGExploreR**. Whole gene regions are obtained from the **txdb.hsapiens.ucsc.hg19.knowngene** Bioconductor package. To determine the quality of the promoter definitions used, **coverage** is calculated for each gene as the mean of the ratio of the number of tags that map to defined promoter regions relative to the total number of tags that map to the entire gene region (which includes all promoter regions and everything else in between).

S2.1.2 Quantifying transcription levels

We first determine the TSS from each CAGE-seq tag by obtaining the coordinate of the first 5′ base. We can then count the number of these TSS tags that map to a set of P promoters for a given gene g simultaneously across C conditions. This generates the following table of tag counts that we call Y_g:

	promoter 1	promoter 2	$\cdots$	promoter P	
condition 1	y_{11}	y_{12}	$\cdots$	y_{1P}	n_{1+}
condition 2	y_{21}	y_{22}	$\cdots$	y_{2P}	n_{2+}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
condition C	y_{C1}	y_{C2}	$\cdots$	y_{CP}	n_{C+}
	n_{+1}	n_{+2}	$\cdots$	n_{+P}	n_{++}

Where the row, column and grand totals are obtained by summing across columns, rows and both respectively:

$$n_{i+} = \sum_{j=1}^{P} y_{ij} \quad , \quad n_{+j} = \sum_{i=1}^{C} y_{ij} \quad , \quad n_{++} = \sum_{i=1}^{C} \sum_{j=1}^{P} y_{ij}$$

If R replicates are present for one or more of the conditions, one can imagine incorporating counts for these replicates in a third dimension by adding an additional subscript in the form $y_{condition,promoter,replicate} = y_{i,j,k}$. Nonetheless, counts are pooled across replicates by summation as follows, and we always end up with the form of the table as seen above with only two subscripts.

$$y_{ij} = \sum_{k=1}^{R} y_{ijk}$$

S2.1.3 Defining promoter composition

The table of counts is converted to a table of row proportions by dividing each cell by the row totals. Margin proportions are defined in the usual way.

	promoter 1	promoter 2	$\cdots$	promoter P	
condition 1	$p_{1\|1}$	$p_{2\|1}$	$\cdots$	$p_{P\|1}$	p_{1+}
condition 2	$p_{1\|2}$	$p_{2\|2}$	$\cdots$	$p_{P\|2}$	p_{2+}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
condition C	$p_{1\|C}$	$p_{2\|C}$	$\cdots$	$p_{P\|C}$	p_{C+}
	p_{+1}	p_{+2}	$\cdots$	p_{+P}	

$$p_{j|i} = \frac{y_{ij}}{n_{i+}} \quad , \quad p_{+j} = \frac{n_{+j}}{n_{++}} \quad , \quad p_{i+} = \frac{n_{i+}}{n_{++}}$$

The vector of proportion values $p_i = (p_{1|i}, \cdots, p_{P|i})'$ is termed the *observed* **promoter composition** for condition i. This set of values shows how transcription in a particular condition is divided between the various promoters available.

We assume that these observed composition values are sampled randomly from a population with corresponding parameter vector $\pi_i = (\pi_{1|i}, \cdots, \pi_{P|i})'$ which is the *true* **promoter composition** for condition i. We have dropped the subscript g to avoid cluttering, but it is important to remember that each composition vector is defined within a given gene.

S2.1.4 Differential promoter composition

Our primary goal is to determine whether the promoter composition vector for a gene is homogeneous across conditions, i.e. our null alternative hypotheses are:

$$H_0 : \boldsymbol{\pi}_1 = \boldsymbol{\pi}_2 = \cdots = \boldsymbol{\pi}_C$$

$$H_1 : \text{some } \boldsymbol{\pi}_r \neq \boldsymbol{\pi}_s$$

We need a statistic that measures the magnitude of differential promoter composition as a "distance" away from the null case. For this we use the **proportional entropy reduction** or *uncertainty coefficient* (Theil, 1970):

$$U = -\frac{\sum_{i=1}^{C} \sum_{j=1}^{P} p_{ij} \log \left(\frac{p_{ij}}{p_{i+} p_{+j}} \right)}{\sum_{j=1}^{P} p_{+j} \log p_{+j}} \quad , \quad \text{where} \quad p_{ij} = \frac{y_{ij}}{n_{++}}$$

This value is well-defined as long as at least 2 promoters are active in at least 1 condition. It ranges from 0 to 1. When it is 0, this means that the promoter composition across all conditions is identical (independence) and when it is 1, this means that each condition transcribes from a single promoter, i.e. there is promoter specificity to condition. We can now conveniently restate our set of null and alternative hypotheses in terms of this statistic:

$$H_0 : U = 0$$

$$H_1 : U \neq 0$$

To test this null hypothesis for each gene, we use a three-step approach (again we drop the g subscript for clarity):

1. Perform Monte Carlo sampling to obtain a set of B matrices $(Y_1^*, ..., Y_B^*)$ of pseudo-counts under the null hypothesis such that:

$$((Y_b^*)_{1j}, ..., (Y_b^*)_{Cj}) \overset{iid}{\sim} \text{Negative Binomial}(\mu_j, \phi) \quad \text{where} \quad \mu_j = \frac{1}{C} \sum_{i=1}^{C} y_{ij}$$

We use the negative binomial parametrized in terms of its mean and dispersion, such that

$$E[\text{Negative Binomial}(\mu, \phi)] = \mu$$

$$\text{Var}[\text{Negative Binomial}(\mu, \phi)] = \mu + \phi \mu^2$$

This strategy allows for variation in the pseudo-counts for various promoters based on their empirical means, but restricts the pseudo-counts across conditions to be identically and independently

distributed. For our purposes here, we choose the Negative Binomial distribution as the most popular distribution for count data. The user has a choice when running **CAGExploreR** whether they want to estimate the dispersion coefficient ϕ, or whether they want to set it to 0 and alternatively use the Poisson distribution. Because of its lower variance, the Poisson distribution will generate more statistically significant results than the negative binomial and this may be desirable for initial data exploration.

2. Calculate the entropy reduction U for each of the B resampled Y^* matrices to obtain it's null distribution.

3. Obtain a one-sided p-value by calculating the upper tail probability from a fitted Beta distribution to the set of resampled values of U in step 2. The beta distribution has two parameters α and β:

$$f_U(u) = \frac{u^{\alpha-1}(1-u)^{\beta-1}}{B(\alpha,\beta)} \quad \text{where} \quad B(\alpha,\beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du$$

We fit a Beta distribution by obtaining parameter esimates α_{MOM} and β_{MOM} by using the Method of Moments. This involves solving the following system of equations where the first two empirical moments are equated to the population moments:

$$\bar{u} = \frac{\alpha_{MOM}}{\alpha_{MOM} + \beta_{MOM}}$$

$$s_u^2 + \bar{u}^2 = \frac{\alpha_{MOM}\beta_{MOM}}{(\alpha_{MOM} + \beta_{MOM})^2(\alpha_{MOM} + \beta_{MOM} + 1)} + \left(\frac{\alpha_{MOM}}{\alpha_{MOM} + \beta_{MOM}}\right)^2$$

This results in the following estimates:

$$\alpha_{MOM} = \bar{u}\left(\frac{\bar{u}(1-\bar{u})}{s_u^2} - 1\right) \quad \text{and} \quad \beta_{MOM} = (1-\bar{u})\left(\frac{\bar{u}(1-\bar{u})}{s_u^2} - 1\right)$$

The Beta distribution is the natural distribution for values that range between 0 and 1, and depending on its two parameters, it can take on a wide variety of shapes. After simulating many different scenarios, we find that the Beta distribution has extremely good fit for U under the null (data not shown).

4. Perform correction for multiple comparisons using the Benjamini-Hochberg method (1995) for controlling False Discovery Rate or any other appropriate method.

S2.1.5 Pair-wise promoter switching

Once we obtain a list of genes that appear to have differential promoter composition, it is naturally of interest to determine which promoters switch from one condition to another the most as this gives us insight into potential changes in transcript generation and differential regulation. To achieve this goal, we calculate pair-wise switching effect measure statistics between all $\binom{P}{2} \times \binom{C}{2}$ promoter and condition comparisons.

		promoter r		promoter s	
	$\ddots$	$\vdots$	$\cdots$	$\vdots$	$\cdot^{\cdot^{\cdot}}$
condition c	$\cdots$	p_{cr}	$\cdots$	p_{cs}	$\cdots$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
condition d	$\cdots$	p_{dr}	$\cdots$	p_{ds}	$\cdots$
	$\cdot^{\cdot^{\cdot}}$	$\vdots$	$\cdots$	$\vdots$	$\ddots$

As a measure of pairwise promoter switching we use the odds ratio which we calculate directly from the raw tag counts:

$$OR = \frac{p_{cr}/p_{cs}}{p_{dr}/p_{ds}} = \frac{p_{cr}p_{ds}}{p_{dr}p_{cs}} = \frac{y_{cr}y_{ds}}{y_{dr}y_{cs}}$$

It has interpretation as follows: the odds of transcription from promoter r relative to that from promoter s is OR times higher (or lower) in condition c v.s. condition d.

NOTE: We can simultaneously exchange the ordering of the promoter pair and condition pair indexes without changing the interpretation: i.e. this is identical to the interpretation as the odds of transcription from promoter s relative to that from promoter r is OR times higher (or lower) in condition d v.s. condition c.

NOTE: If we exchange the ordering of only one of the pairs (either promoter or condition), then the the effect size is the inverse of the odds ratio, i.e. the odds of transcription from promoter s relative to that from promoter r is $1/OR$ times higher (or lower) in condition c v.s. condition d.

NOTE: The words higher or lower used in the paragraphs above refer to whether the OR is greater or less than 1 respectively.

We choose the odds ratio as the switching effect measure for the following attractive attributes that it possesses:

1. Can be calculated directly from the original tag count data.

2. It is independent of the sequencing depth or library sizes involved. Similarly it does not depend on differences in gene expression between conditions (note lack of y_{i+} and y_{+j} terms).

3. It is immune to multiplicative transformation of the data in both rows and columns. This is useful because it makes the *OR* the same irrespective of any normalization method used on the data where either the promoter expression columns or condition samples or both are normalized by multiplying by a set of constants. This also means that the data need not be integer counts, but any real set of numbers.

4. If future normalization methods are applied which scale every promoter/condition combination of cell counts, i.e. where every cell is multiplied by a separate normalizing constant, then the *OR* based on such normalized data will be just a scalar multiple of the *OR* based on the raw data.

We can obtain p-values for the test of $H_0 : OR = 1$ by using the log-transformed odds ratio statistic below. We perform the log transformation because it approaches asymptotic normality much faster than the raw odds ratio (Agresti p.71, 2002).

$$z = \frac{\log(OR)}{\sqrt{\frac{1}{y_{cr}} + \frac{1}{y_{dr}} + \frac{1}{y_{ds}} + \frac{1}{y_{cs}}}}$$

We compare this value to a standard normal distribution and calculate both upper and lower tail probabilities to obtain a two-sided p-value.

One problem with the odds ratio is that it is of little use if any of the tag counts y are 0. In such a case the odds ratio will be either 0 or ∞. In situations where such a case occurs, we use an adjusted form of the odds ratio defined as follows:

$$\widetilde{OR} = \frac{(y_{cr} + k)(y_{ds} + k)}{(y_{dr} + k)(y_{cs} + k)} \quad , \quad \text{where} \quad k = \frac{1}{2}$$

By adding a constant to all count cells we avoid the problem of zero counts. It has been shown that the value of $k = 1/2$ is optimal in terms of minimizing the bias of the odds ratio estimator (Gart, 1966). We then use the following adjusted test statistic for a Wald test:

$$z = \frac{\log(\widetilde{OR})}{\sqrt{\frac{1}{y_{cr}+0.5} + \frac{1}{y_{dr}+0.5} + \frac{1}{y_{ds}+0.5} + \frac{1}{y_{cs}+0.5}}}$$

Assuming we have G genes of interest and P_g and C_g are the number of promoters and conditions available for gene g respectively, we have a total of $\sum_{g=1}^{G} \binom{P_g}{2} \times \binom{C_g}{2}$ odds ratios. After obtaining p-values for each one, we then adjust them for false discovery rate to obtain q-values (Benjamini-Hochberg, 1995).

S2.1.6   Gene Expression Heterogeneity

Independently of looking at changes in promoter composition, it is of interest to quantify the variability in total gene expression across conditions. In order to do this across multiple conditions simultaneously, we define the following *heterogeneity index* for each gene.

$$\text{GeneHetero} = 1 - \frac{\sum_{i=1}^{C} P_i \log P_i}{\log 1/C}$$

There are C conditions and P_i is the gene expression proportion for condition i. The index is based on the entropy as a measure of variability, except that it is normalized by the maximum entropy to make sure that it can take on values between 0 and 1. A value of 0 corresponds to the gene expression remaining constant across all conditions, and a value of 1 corresponds to condition specificity for gene expression (i.e. one condition is expressed but all others are not). Values between 0 and 1 are hard to interpret, and are only meant to rank genes based on the variability in their expression. For the case of only 2 conditions being compared, the following graph shows the relationship between the expression proportion and the value of the heterogeneity index. We can see that it takes on a minimum at 0.5, and maxima at the edges as expected. In between, the relationship is non-linear.
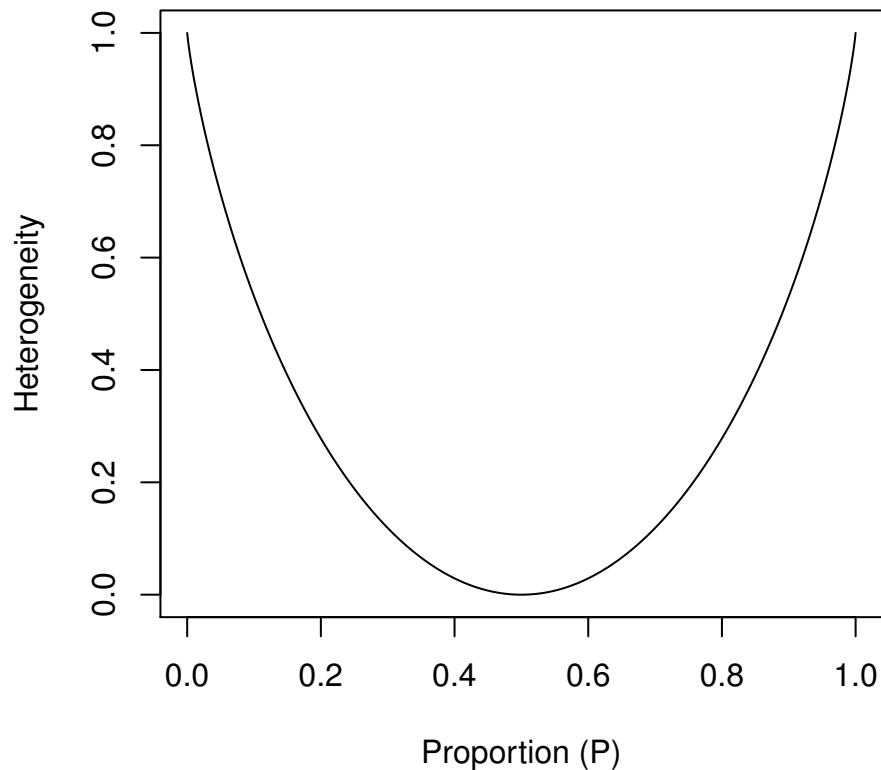
Supplementary Figure S2.2: Gene expression heterogeneity measure as a function of composition proportion

S2.2 References

Agresti, A. (2002). Categorical Data Analysis. Second Edition. John Wiley Sons, Inc., New York, New York, USA.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*. 57 (1): 289âĂŞ300.

Carlson, M. TxDb.Hsapiens.UCSC.hg19.knownGene: Annotation package for TranscriptDb object(s). R package version 2.9.2.

FANTOM5 Consortium (2013). A promoter level mammalian expression atlas. Under review at Nature.

Gart, J.J. (1966). Alternative analyses of contingency tables. *Journal of the Royal Statistical Society Series B*. 28:164-179.

Gupta, R., Bhattacharyya, A., Agosto-Perez, F. J., Wickramasinghe, P. and Davuluri, R. V. (2011). MPromDb update 2010: An integrated resource for annotation and visualization of mammalian gene promoters and ChIP-seq experimental data. *Nucleic Acids Research*, Vol. 39, D92-97.

Theil, H. (1970). On the estimation of relationships involving qualitative variables. *American Journal of Sociology*. 76: 103-154.

Supplementary Materials for pcmR

S3.1 Biological Pathways

Assuming that an organism has a total of N genes $\{G_1, ... G_N\}$, a pathway P is defined as a collection of genes: $\{G_i : i \in P\}$ that share some functional relationship. Users of `pcmR` may supply their own pathways or use any of the ones that come with the package. These include a total of 633 pathways from KEGG (Kanehisa et al. 2010), Reactome (Croft et al. 2011), Wikipathways (Pico et al. 2008), Netpath (Kandasamy et al. 2010) and Static Modules (Wu et al. 2010).

S3.2 Expression v.s. Composition

For a given biological sample, the *gene expression* for a set of genes $\{G_1, ... G_N\}$ is a vector of intensity values (in microarrays) or tag counts (in RNA-Seq) $\mathbf{E} = \{E_1, ... E_N\}$ which may be normalized between samples.

The *pathway expression* of a pathway P is the equivalent *total* gene expression across the constituent genes of that pathway: $T_P = \sum_{i \in P} E_i$.

The gene *composition* of a pathway P is defined as the proportion vector $\mathbf{C}_P = \{E_i/T_P : i \in P\}$. Both the original expression vector and the pathway context are required to unambiguously define a composition measure. It can be seen that the composition vector sums to 1 and can only be defined for pathways that are expressed, i.e. $T_P > 0$.

In RNA-Seq and other digital expression assays, it is standard practice to normalize the gene expression for sampling depth by dividing the tag counts by the library size. When multiplied by a scaling factor usually taken to be $k = 1 \times 10^6$, such values are referred to as *tags per million* or *tpm*. This transforms the original expression vector $\mathbf{E}$ to a scaled composition vector $k\mathbf{C}$ where the "pathway" is the entire genome. These vectors are then compared between conditions using what are inaccurately termed differential "expression" analyses. Problems arise when a subset of genes in the genome are strongly differentially expressed, resulting in apparent differential expression in other, unrelated and truly non-differentially expressed genes. For

example, if we have two conditions, A and B with a total of 1000 genes in the genome:

$$\text{Condition A: } \mathbf{C} = \{C_1, ..., C_{1000}\}$$

$$\text{Condition B: } \mathbf{C}' = \{C'_1, ..., C'_{1000}\}$$

If gene 1000 is truly differentially expressed, we may find that $C_{1000} \neq C'_{1000}$ as expected. If at the same time, gene 1 is *not* differentially expressed and all other genes remain constant however, it is not guaranteed that $C_1 = C'_1$ since the composition vector sums to 1 (or to a constant if the composition is scaled). Any change in C_{1000} between conditions will inevitably lead to a corresponding change in the opposite direction in C_1 leading to a false positive finding. Similarly, false negatives can also occur in slightly more complex scenarios. This problem is called *RNA composition bias*. Current solutions do not solve the problem, but attempt to reduce its effect by scaling the expression vectors by constants such that differences between conditions are minimized in the hope that this will minimize the impact on the composition bias. `edgeR` includes the option to perform TMM normalization (edgeR User Guide) which multiplies the library size by some constant to perform said minimization, resulting in what is called a scaled or *effective* library size.

Note: RNA composition bias happens due to the fact that composition vectors sum to a constant, resulting in negatively correlated elements. It is a problem only when the composition is calculated over a pathway that is overly heterogeneous in its members and does not describe a system that works together to achieve a single common goal. Having too many members is usually a symptom of this problem as seen above in the case of the entire genome being used as a pathway. On the other hand, this problem is of no concern if the pathway is chosen such that it is small and self-contained. In fact, what is previously called a composition *bias*, becomes of primary interest and should not be mitigated but explored. The pathway definition is crucial to the success of this analysis, and from here on, we assume that pathways are defined correctly using the best available knowledge.

S3.3   Modeling Pathway Composition

The association graph below gives a top-level view of the main factors at play and their relationships as we shall consider them. This setup is widespread in its use across the fields of epidemiology and the design of experiments and clinical trials.
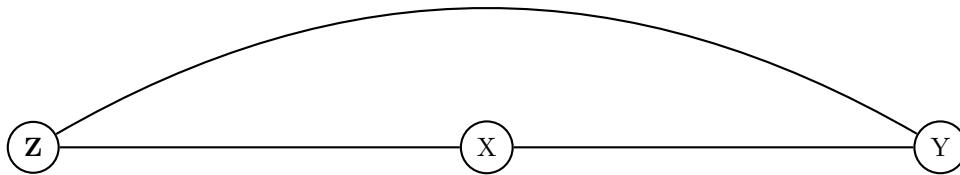
Y is the outcome of interest such as a binary disease state, i.e. $Y = 0$ for healthy and $Y = 1$ for subjects with disease. Y can also be continuous, e.g. systolic blood pressure.

X is the main determinant under study that affects the outcome, e.g. $X = 0$ for placebo and $X = 1$ for individuals in the treatment arm. X can also be continuous, e.g. drug dosage in grams.

$\mathbf{Z}$ are the set of covariates that may confound the relationship between X and Y. One wishes to adjust for the effect of these confounders in order to remove their effect on the outcome Y so that the pure effect of X on outcome Y can be elicited. Examples of confounders are patient age, measurement batch, smoking status, etc.

In `pcmR`, we distinguish between two types of models:

1) The **Forward Model**, where the primary determinant under study is $X = \mathbf{C}_P$. We assume for now that Y is a binary outcome. In other words, `pcmR` answers the question, **"What are the effect of changes in gene composition on the risk of disease outcome, adjusting for the confounding effect of covariates?"**.

2) The **Reverse Model**, where the X and Y are flipped. In other words, `pcmR` answers the question, **"How does mean gene composition differ when comparing diseased to healthy individuals, adjusting for the confounding effect of covariates?"**.

This question is answered by fitting a generalized linear model (GLM) (McCullagh and Nelder, 1972):

$$g(\mu) = \beta_0 + \beta_1 X + \sum_{j=1} \gamma_j Z_j \quad \text{where} \quad \mu = \mathrm{E}\left[Y\right]$$

S3.3.1 Distribution Assumptions

We assume that the composition vector $\mathbf{C}_P$ follows a Dirichlet($\boldsymbol{\mu}$,ϕ) distribution with the following p.d.f. (Maier, 2014):

$$f(\mathbf{c}|\boldsymbol{\mu}, \phi) = \frac{1}{B(\boldsymbol{\mu}\phi)} \prod_{i=1}^{|P|} c_i^{\mu_i \phi - 1} \quad \mu \in (0,1) \quad \phi > 0$$

Here, $B(\cdot)$ is the Beta function, $\mathrm{E}\left[C_i\right] = \mu_i$ and ϕ is the precision parameter such that $\mathrm{Var}\left[C_i\right] = \frac{\mu_i(1-\mu_i)}{1+\phi}$ and $\mathrm{Cov}\left[C_i, C_j\right] = \frac{-\mu_i \mu_j}{1+\phi}$. Maier (2014) provides plots to visualize this distribution in 3 dimensions.

Binary exposures and outcomes are assumed to be Bernoulli(p) distributed.

No assumptions are necessary for the set of confounders $\mathbf{Z}$.

S3.3.2   Forward Model

In this model, the outcome is the composition vector while the effect of interest is the disease state. The following Dirichlet regression model is fit via maximum likelihood using the R package DirichReg (Maier, 2014):

$$\text{logit}(\mu_i) = \beta_0 + \beta_1 Y + \gamma_0 T_P + \sum_{j=1} \gamma_j Z_j \quad \text{where} \quad \mu_i = \text{E}\,[C_i]$$
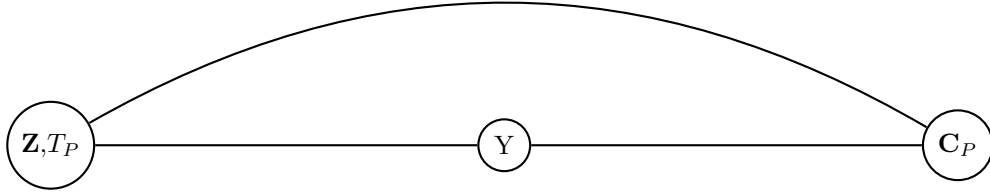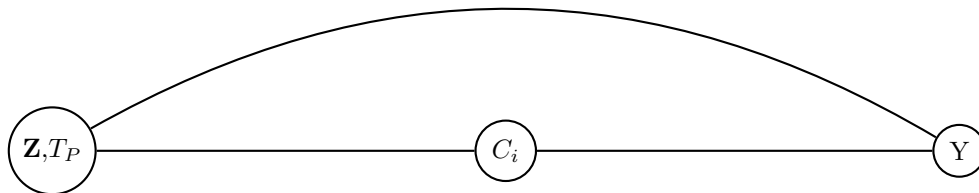

It is important to note that the total pathway expression T_P is always included in the model as a potential confounder. Just because the proportion involves dividing the expression by the total, this does not remove the effect of total expression on the outcome of interest. In fact, the expression and the total both contribute to the proportion together. If not adjusted for in a model, what would appear to be the effect of composition on outcome could also be due to changes in total pathway expression, leading to confounding.

β_1 is reported as the effect of interest and is interpreted as the log odds ratio (log OR) of composition when comparing diseased to healthy subjects while adjusting for total pathway expression and the set of confounders $\mathbf{Z}$. For pathways with a large number of member genes, this log OR becomes approximately the log fold change (log FC) in composition between diseased and healthy subjects while adjusting for confounding. In other words, how does the composition of a gene in pathway P change when comparing between diseased and healthy individuals, while keeping total pathway expression and other confounders fixed.

S3.3.3   Reverse Model: Binary Outcome

In this model, the outcome is the disease state while the effect of interest is the composition vector. The following set of logistic regression models is fit via maximum likelihood using base R for each pathway:

$$\text{logit}(p) = \beta_0 + \beta_1 C_i + \gamma_0 T_P + \sum_{j=1} \gamma_j Z_j \quad \text{where} \quad p = \text{E}\,[Y]$$

It is important to note that the total pathway expression T_P is always included in the model as a potential confounder for the same reasons as described in the previous section.

β_1 is reported as the effect of interest and is interpreted as the log odds ratio (log OR) of risk of disease for a unit increase in composition of a gene in pathway P while adjusting for total pathway expression and the set of confounders $\mathbf{Z}$. This is the effect on risk of disease of going from a state in which this gene is completely insignificant (is not active at all) in pathway P to the state in which this gene completely dominates (is the only gene active) in that pathway while the total expression of the pathway remains unchanged, and while keeping other confoudners $\mathbf{Z}$ constant.

S3.4   Visualization

Visualizations of pathways are made possible by interfacing with the `R` package `pathview` (Luo and Brouwer, 2013), however only KEGG pathways can be visualized in this manner. Both gene expression and gene composition data can be overlayed on the directed KEGG pathway maps.

S3.5   A Real Data Example

We demonstrate `pcmR` and compare it to `limma` (Smyth, 2005) on a dataset of non-small cell lung cancer (NSCLC) tumors with matched controls from 60 Taiwanese non-smoking women (Lu et al. 2010). Gene expression was measured using the Affymetrix U133Plus2.0 microarray platform. We chose this dataset for several reasons. First of all, it is freely available from the Gene Expression Omnibus (GEO) as a curated dataset (GDS3837). In addition to having a large sample size, most importantly, it also provides metadata covariates beyond the gene expression values: the age of the patients. This allows us to compare and contrast the linear modeling capabilities of `pcmR` and `limma`.

To confirm that the results from `pcmR` make sense, we first look at the Non-Small Cell Lung Cancer pathway from KEGG, a pathway that has a direct relation to the dataset. In `limma`, a model was fit linking the gene expression of genes from this pathway (outcome) to the disease state of the sample (tumor or normal - predictor) while adjusting for the age of the patient (confounder). The main effect of disease state measured

as the log FC was extracted for each gene. Using `pcmR`, we fit the Forward Model with the same predictor and confounder, but with the gene composition values as the outcome instead of the gene expression. Total pathway expression was included as an additional confounder to the model for reasons described previously. The main effect of interest as measured by the log OR was extracted for each gene. Only probes that map to Entrez gene identifiers were kept in the analysis. The average expression was obtained for Entrez genes that had multiple probes mapping to them. The results of this comparison are summarized in the pathway visualization in the figure below.
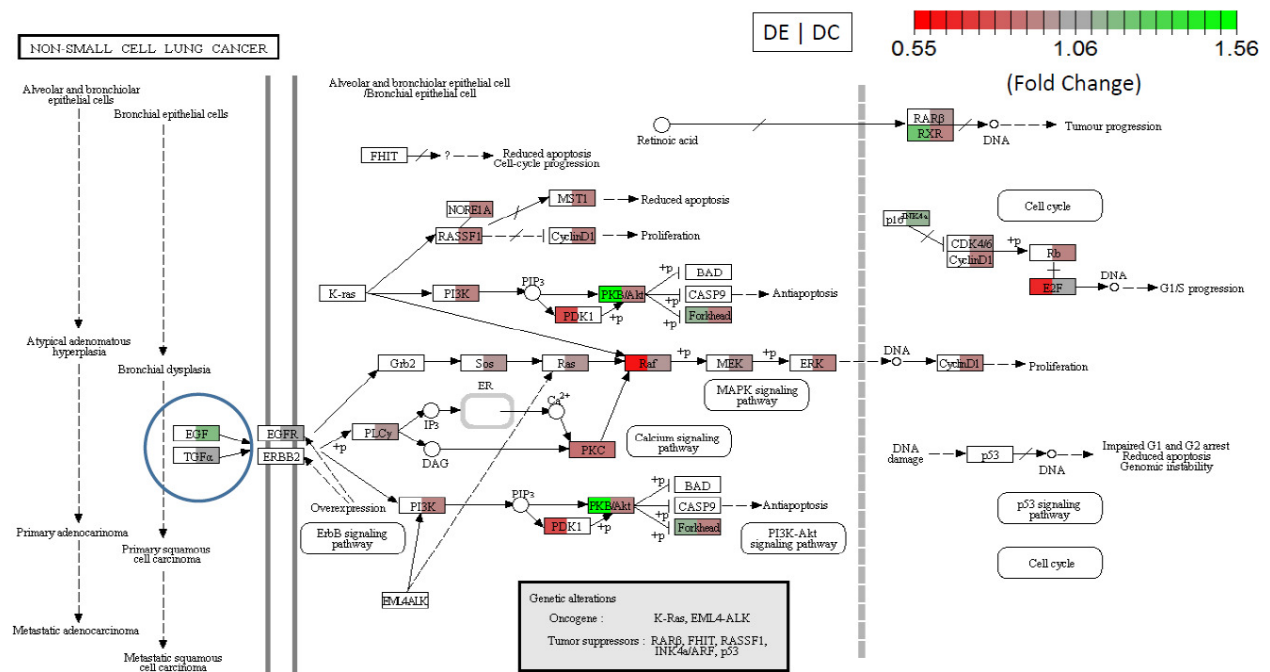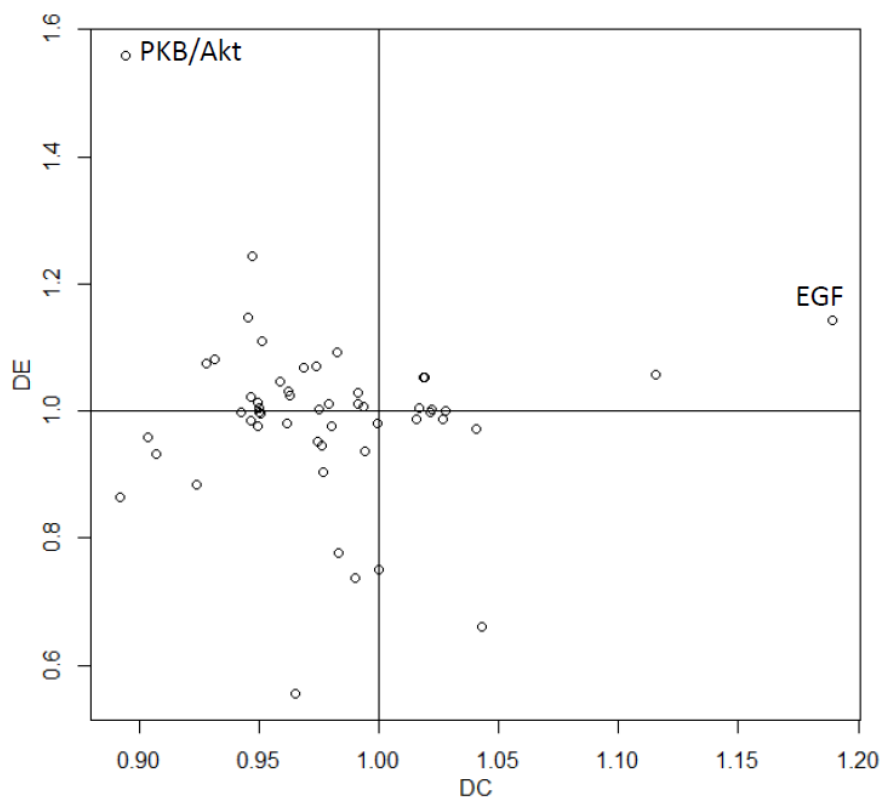

Supplementary Figure S3.1: NSCLC pathway from KEGG. Each gene box color coded by effect size comparing tumor to healthy controls, adjusted for patient age. Left/right half of each gene: `limma` FC/`pcmR` OR.

Only effect sizes that were statistically significant using the criterion of false discovery rate (FDR)-adjusted (Benjamini and Hochberg, 1995) p-value < 0.05 were included in the figure above. Genes that are not statistically significant do not have a color associated with them and remain white.

We first note that the majority of genes in this pathway do not pass the statistical significance threshold when using `limma`, whereas `pcmR` detects many more statistically significant results. `pcmR` finds *EGF* and *p16* as over-composed, whereas `limma` finds a different set of genes: *PKB/Akt*, *RXR* and *Forkhead* as up-regulated in cancer. This suggests that the results obtained from differential expression are different from those obtained from a differential composition analysis. The figure below plots the effect size (FC and OR) obtained from both `limma` and `pcmR` for this pathway. The correlation between the two effect measures is low at -0.087 (p-value > 0.05), once again suggesting that differential expression and differential composition

measure different unrelated effects.

Supplementary Figure S3.2: Differential Expression (DE) FC v.s. Differential Composition (DC) OR measures for the NSCLC pathway. *PKB/Akt* and *EGF* are highlighted as the most strongly differentially expressed/composed genes in this pathway.

As discussed in the main manuscript, *EGF* is located most upstream of the entire pathway, suggesting that in the tumor, an increase in its composition is a hallmark of the disease. Current research specifically focusing on non-smoking Asian women who have NSCLC has consistently highlighted the receptor of *EGF* as strongly implicated in this disease. In fact, current treatment actively targets this gene with great success. This is again confirmed in our study here, where *EGF* is by far the strongest differentially-composed in the NSCLC pathway. It is natural in current differential expression analyses to focus on the most strongly differentially expressed genes, however *EGF* was not found to be statistically significantly differentially expressed, and so would have been missed by a conventional analysis. On the other hand, a set of downstream genes were found to be differentially expressed, ones which were not over-composed in the differential composition analysis. As mentioned in the main manuscript, the most differentially expressed gene, *PKB/Akt* has been implicated in NSCLC as a driver following resistance to anti-*EGFR* treatment. Even though it was not found to be over-composed in our analysis here, we speculate that if a repeat sample was made of the tumor from patients following resistance to anti-*EGF* drugs, it is likely that *PKB/Akt* would be found to be over-composed then.

Future research is needed to confirm this hypothesis.

We next look at the differential composition results across all 633 pathways included with `pcmR` in order to check if the results look reasonable and could be useful in providing potential drug targets. The figure below shows sample output from `pcmR`, displaying the log OR (estimate), standard error, z-value, p-value, FDR-adjusted p-value, the corresponding gene, its pathway context and the pathway source. As mentioned in the main manuscript, *MASP1* from the Signaling in Immune System pathway from Reactome comes second as most differentially composed. The under-composition of this gene in NSCLC suggests that an increase in its composition could be a potential approach to treat the disease. We focus on this gene as opposed to any others in the top list because a literature review of this gene, the lectin pathway in which it is involved, and a mushroom as a source of lectins in the treatment of cancer provides a coherent story and an interesting justification that a composition analysis can yield results that are interpretable and potentially useful.

```
  Estimate Std..Error   z.value     Pr...z..         FDR     gene entrez                         pathway   source
 -81082.64  16246.023 -4.990922 6.009162e-07 0.0250389768     QDPR   5860              Metabolic pathways     KEGG
 -69989.48  13471.023 -5.195558 2.041072e-07 0.0086829241    MASP1   5648 Signaling in Immune system Reactome
 -56302.27  11065.611 -5.088040 3.617838e-07 0.0152310968   PIK3C3   5289              Metabolic pathways     KEGG
  55679.75  10882.060  5.116655 3.110014e-07 0.0131332770 ALDH18A1   5832              Metabolic pathways     KEGG
  47933.87   9689.367  4.947059 7.534305e-07 0.0312229119     AHCY    191              Metabolic pathways     KEGG
 -47266.86   9704.544 -4.870591 1.112651e-06 0.0456899077    ACAT1     38              Metabolic pathways     KEGG
 -47204.73   9329.351 -5.059809 4.196765e-07 0.0176205358    PLCE1  51196              Metabolic pathways     KEGG
 -46376.42   9416.250 -4.925147 8.429700e-07 0.0348542791   CYP3A7   1551              Metabolic pathways     KEGG
  46356.00   9533.285  4.862542 1.158875e-06 0.0475393869     CHPF  79586              Metabolic pathways     KEGG
```

Supplementary Figure S3.3: Top 9 differentially composed genes across 633 pathways included with `pcmR`

S3.6   Appendix: a comparison of model paradigms

In this appendix, the various gene expression methodologies that allow for the fitting of linear models are compared to each other. Key differences and commonalities are highlighted and explored. In all cases, we assume a simple model with one main binary effect of interest, x. To make interpretation simpler, we assume that $x_i = 1$ corresponds to a person who has disease and $x_i = 0$ corresponds to one who is healthy. However these can be any 2 groups of samples being compared, not necessarily human or related to disease.

S3.6.1   limma

`limma` was the first tool to introduce the ability to fit linear models to gene expression data from microarray platforms. For each gene, the following linear model is fit. Y_i is the intensity of the signal on the microarray for subject i, x_i is subject i's binary disease status and z_{ij} is the j'th confounding covariate in the model for subject i. β_1 is the *difference in the mean log-intensities of a gene* between diseased and healthy samples. e^{β_1}

is usually reported as the fold change in mean expression between diseased and healthy, however in reality it is a value that is *smaller than the true fold change in the mean expression of a gene*, a result due to Jensen's inequality ($e^{E[x]} \leq E[e^x]$ since e^x is a convex function).

$$E\left[\log(Y_i)\right] = \beta_0 + \beta_1 x_i + \sum_{j=1}^{k} \gamma_j z_{ij}$$

S3.6.2   edgeR

`edgeR` was developed to analyze gene expression data arising from digital expression technologies such as SAGE and RNA-Seq. Instead of intensities, counts of tags measure expression. For each gene, the following generalized linear model (GLM) is fit. Y_i is the number of tags mapping to the gene for subject i, N_i is the library size (total number of tags mapped to the genome) for subject i. e^{β_1} is the *fold change in the mean tag counts of a gene* between diseased and healthy for a fixed arbitrary library size.

$$\log E\left[Y_i\right] = \beta_0 + \beta_1 x_i + 1 \cdot \log N_i + \sum_{j=1}^{k} \gamma_j z_{ij}$$

A key difference from `limma` is that the GLM framework allows for e^{β_1} to be an unbiased estimate of the true fold change. Also, an additional covariate, $\log N_i$ is always added to the model. This is called an "offset", and it does not have an estimable beta coefficient; it is taken to be always 1.

S3.6.3   pcmR

Let's first look at what happens when we take the entire genome as one "pathway" so that we can directly compare with `edgeR`. The following model is fit to each gene. The key difference now is that the effect of the total library size is no longer fixed at a value of 1, but is estimated from the data. Also, instead of the log, the logit is instead used as the link between the mean outcome and the linear predictor. e^{β_1} is the *odds ratio in the mean composition of a gene* comparing diseased v.s. healthy subjects for a fixed arbitrary library size.

$$\text{logit } E\left[\frac{Y_i}{N_i}\right] = \beta_0 + \beta_1 x_i + \gamma_0 N_i + \sum_{j=1} \gamma_j z_{ij}$$

Note: as the number of genes in a pathway increases, the individual composition of each gene in that pathway approaches zero and as a result, the odds ratio reported by `pcmR` approaches the more easily interpretable fold change.

Finally in practice, `pcmR` would be applied to analyse smaller-sized pathways, in which case the following model is fitted to each gene where $(T_P)_i$ is the total expression (measured as an intensity or tag count) of

pathway P for subject i. e^{β_1} is the *odds ratio in the mean composition of a gene in pathway P* comparing diseased v.s. healthy subjects for a fixed arbitrary library size.

$$\text{logit } E\left[\frac{Y_i}{(T_P)_i}\right] = \beta_0 + \beta_1 x_i + \gamma_0 (T_P)_i + \sum_{j=1} \gamma_j z_{ij}$$

One major advantage of `pcmR` is that it can work on data that arises from both microarrays and digital expression technologies.

S3.7   References

Croft D, O'Kelly G, Wu G, Haw R, Gillespie M, Matthews L, Caudy M, Garapati P, Gopinath G, Jassal B, Jupe S, Kalatskaya I, Mahajan S, May B, Ndegwa N, Schmidt E, Shamovsky V, Yung C, Birney E, Hermjakob H, D'Eustachio P, Stein L. (2011). Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res*, 39:D691-697.

Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M. (2010). KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Res*, 38:D355-360.

Pico AR, Kelder T, van Iersel MP, Hanspers K, Conklin BR, Evelo C. (2008). WikiPathways: pathway editing for the people. *PLoS Biol*, 6:e184.

Kandasamy K, Mohan SS, Raju R, Keerthikumar S, Kumar GSS, Venugopal AK, Telikicherla D, Navarro JD, Mathivanan S, Pecquet C, Gollapudi SK, Tattikota SG, Mohan S, Padhukasahasram H, Subbannayya Y, Goel R, Jacob HKC, Zhong J, Sekhar R, Nanjappa V, Balakrishnan L, Subbaiah R, Ramachandra YL, Rahiman BA, Prasad TSK, Lin J-X, Houtman JCD, Desiderio S, Renauld J-C, Constantinescu SN, et al. (2010). NetPath: a public resource of curated signal transduction pathways. *Genome Biol*, 11:R3.

Wu G, Feng X, Stein L. (2010). A human functional protein interaction network and its application to cancer data analysis. *Genome Biol*, 11:R53.

Maier, M. (2014). DirichletReg: Dirichlet Regression for Compositional Data in R. *Research Report Series*, Report 125, January 2014. Institute for Statistics and Mathematics. Vienna University of Economics and Business.

Luo, W. and Brouwer, C. (2013). Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics*, 29(14):1830-1831.

Lu, T.P., Tsai, M.H., Lee, J.M., Hsu, C.P., Chen, P.C., Lin, C.W, Shih, J.W., Yang, P.C., Hsiao, C.K.,

Lai, L.C. and Chuang, E.Y. (2010). Identification of a novel biomarker, SEMA5A, for non-small cell lung carcinoma in nonsmoking women. *Cancer Epidemiol. Biomarkers Prev.* 19(10):2590-7.

Smyth, G.K. (2005). Limma: linear models for microarray data. In Gentleman, R., Carey, V., Dudoit, S., Irizarry, R. and Huber, W. (eds.), *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pp. 397-420. Springer, New York.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B* 57, 289300.