11-2012

COMPARATIVE STUDIES OF DIFFERENTIAL GENE CALLING METHODS FOR RNA-SEQ DATA

Ximeng Zheng
University of Nebraska - Lincoln, seymourzxm@gmail.com

COMPARATIVE STUDIES OF DIFFERENTIAL GENE CALLING METHODS FOR RNA-SEQ DATA

by

Ximeng Zheng

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Biological Sciences

Under the Supervision of Professor Etsuko Moriyama

Lincoln, Nebraska

November, 2012

COMPARATIVE STUDIES OF DIFFERENTIAL GENE CALLING METHODS FOR RNA-SEQ DATA

Ximeng Zheng, M.S.
University of Nebraska, 2012

Advisor: Etsuko Moriyama

RNA-Seq is a recently developed technology that can reveal RNA expression profile by taking advantage of deep sequencing. This new technology has many advantages over microarray technologies. Although RNA-Seq is expected to overtake microarray experiments due to their massive amounts of data produced, it presents many challenges to bioinformatics research regarding efficient data processing and storage, and accuracy in data interpretation. One of the challenges and also an important aspect of expression profiling is to detect differentially expressed genes between different experimental conditions. Several statistical methods have been developed over the past few years. In this study, we chose two representative methods: one parametric method, DESeq, and one nonparametric method, NOISeq. We compared the performance of these two methods using simulated and real datasets. We showed that both DESeq and NOISeq identified over-expressed genes more correctly than under-expressed ones. While DESeq was more likely to call longer genes as differentially expressed, NOISeq did not show such bias. When the underlying variation increased, both methods showed higher false-positive rates at the same threshold. When replicates were not available, both methods

showed lower true-positive and higher false-positive rates. Finally, we explored a strategy to combine the results from DESeq and NOISeq when replicates are available. We showed that it is possible to improve differential gene-calling results by combining the results obtained from the two methods. NOISeq is recommended when no replicate is available.

Acknowledgements

I would like to thank my advisor, Dr. Etsuko Moriyama whose sincerity and encouragement I will never forget, for her continuous guidance and support. She has been my inspiration as I hurdle all the obstacles through this research work. I am also very grateful for my other committee members, Drs. Brisson, Wang and Zhang, for their helpful input and discussion. I would also like to show my gratitude to Drs. Brisson and Cerutti, and Dr. Wilkie from University of Texas Southwestern Medical Center for letting me use their experimental data. Last but not least, I am truly indebted and thankful to all my family members, friends and colleagues for their help and support.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Gene Expression Analysis

Gene expression analysis has long been of interest to molecular biologists. It is done, for example, to identify genes differentially expressed among tissues or among different experimental conditions, to discriminate heterogeneous diseases such as cancer, or to elucidate the relationship between gene expression and covariates such as survival or tumor grade (Barry, et al., 2005). The transcriptome is the complete set of transcripts in a cell and a summary of all gene expressions. It is essential to construct and understand the transcriptome accurately in order to interpret the functional elements of the genome, molecular constituents of cells, development of organisms, and mechanism of diseases (Wang, et al., 2009).

Various technologies have been developed to quantify and analyze transcriptomes over the years. Early technologies such as labor-intensive and expensive cDNA cloning and expressed sequence tag (EST) provided only a limited insight to the complexity and intricacy of transcriptomes (Garber, et al., 2011). Microarray technology was then developed to overcome such limitations for its high throughput and relatively low cost. It is a hybridization-based approach that incubates fluorescently labeled cDNA with either custom-made microarrays or commercial oligo-based microarrays. Microarray provided more comprehension to the transcriptome analysis since it can generate the expression data for thousands of genes simultaneously. Limitations, however, exist also in microarray, *e.g.*, requiring prior knowledge about the genome sequence, high background noise due to cross-hybridization, and a limited dynamic range of detection owing to background noise and saturation of signals (Wang, et al., 2009).

1.2 Next-Generation Sequencing and RNA-Seq

Next-generation sequencing technologies have been developed in recent years. It is significantly different from the traditional Sanger sequencing technology. First of all, next-generation sequencing is ultra high-throughput, which processes millions of sequence reads in a parallel fashion instead of sequentially. Secondly, the workflow to produce next-generation sequencing libraries is different. Instead of using vector-based cloning and *Escherichia coli* based amplification, next-generation sequencing workflow is to ligate specific adaptor oligos to both ends of each DNA fragment and then sequence these DNA fragments. Moreover, next-generation sequencers produce shorter reads in length compared to Sanger sequencing. Depending on different technologies or platforms (*e.g.,* Illumina[1], SOLiD[2], Roche 454[3], Ion Torrent[4], and PacBio[5]), reads can vary in length between around 30bp and 400bp (or ~1,000bp for PacBio), which can be ~20 times shorter than those from Sanger sequencing.

With the massive amount of short reads produced by next-generation sequencers, traditional alignment programs, *e.g.,* BLAST (Altschul, et al., 1990), are too slow and not suitable for mapping all short reads to reference genomes. Various programs have been

[1] http://www.illumina.com/
[2] http://www.appliedbiosystems.com/absite/us/en/home.html
[3] http://www.my454.com/
[4] http://www.iontorrent.com/
[5] http://www.pacificbiosciences.com/

developed to resolve the issue. Some examples include Bowtie (Langmead, et al., 2009), TopHat (Trapnell, et al., 2009), BWA (Li and Durbin, 2009), and SOAP2 (Li, et al., 2009). Unlike regular pairwise-alignment methods based on dynamic programming such as the Smith-Waterman algorithm (Smith and Waterman, 1981) or a heuristic algorithm as BLAST, these short-read aligners use algorithms based on hash tables or Burrows-Wheeler transform (Burrows and Wheeler, 1994). Ruffalo et al. (2011) compared seven popular short-read alignment programs: Bowtie, BWA, SOAP2, mrFAST (Alkan, et al., 2009), mrsFAST (Hach, et al., 2010), Novoalign[6], and SHRiMP (Rumble, et al., 2009). They reported that among these programs, SOAP2 performed quite well and had a consistently high accuracy (above 90%) even when the short-read error rates were as high as 10%.

With these newly developed tools, next-generation sequencing are now used in many aspects of biological research, *e.g.*, mutation discovery, sequencing clinical isolates in strain-to-reference comparisons, enabling metagenomics, defining DNA-protein interactions, discovering noncoding RNAs, and even *de-novo* assembly of transcriptomic and genomic sequences (Mardis, 2008). One particular use of next-generation sequencing technology this thesis focuses on is for quantifying gene expression, which is called RNA-Seq.

RNA-Seq is a recently developed technology based on next-generation sequencing. It is used, for example, to obtain gene-expression profiles, transcriptional structure of genes, and post-transcriptional modifications. RNA-Seq measures the levels of transcripts and their isoforms (alternatively spliced transcripts from the same gene)

[6] http://www.novocraft.com/main/index.php

much more precisely than many other methods. RNA-Seq data are typically generated from a library of cDNA fragments made from a population of RNAs. The cDNA fragments are attached with adaptors on one or both ends (single or paired-end sequencing). Then each molecule is sequenced in a high-throughput fashion with or without amplification. The short reads obtained are aligned to a reference genome or transcriptome. They can be assembled *de novo* if the reference is not available. The number of short reads that are mapped to each reference gene region or transcript can be interpreted as the expression level of the gene or transcript. Illumina and SOLiD platforms are usually used for RNA-Seq experiments. Many different types of analyses, *e.g.*, single nucleotide polymorphism discovery, alternative transcript identification, and gene expression profiling, can be applied on the result of short-read alignment. Compared with aforementioned microarray technology, RNA-Seq has many advantages, *e.g.*, high resolution, low background noise, no prior knowledge of reference sequence required, and being able to distinguish isoforms and allelic expression (Wang, et al., 2009).

1.3 RNA-Seq Data Processing and Analysis

It is expected that digital gene expression (DGE) technologies (*e.g.,* RNA-Seq) will overtake microarray technologies in the near future for many functional genomics applications (Robinson, et al., 2010). It necessitates, however, development of accurate and efficient methods and software to analyze DGE data. The number of short reads mapped onto one gene is the *count* that can be viewed as the expression level of the gene. These count data are different from those obtained from bead and array technologies.

DGE data are fundamentally discrete, rather than continuous as microarray data are, in nature (Hardcastle and Kelly, 2010). Therefore, the techniques developed for analyzing microarray data may not be directly applicable on these DGE data.

Two types of variations exist in microarray as well as in RNA-Seq experiments: biological variation and technical variation. Biological variation is the normal stochastic variation in gene expression between biological samples. Technical variation is the inherent variation of the experimental process. For example, in microarray experiments, technical variation is produced as different signals in different runs of microarray for a same biological sample. In RNA-Seq experiments, different numbers of short reads are produced in different runs of sequencing for a same biological sample. For continuous data obtained by microarray, a normal distribution is usually used to model biological and technical variations after log transformation (Smyth, et al., 2005). However, as mentioned previously, since RNA-Seq count data are not continuous but discrete, the techniques developed for analyzing microarray data are not applicable on RNA-Seq data. Since Poisson distribution cannot model the over-dispersion observed in RNA-Seq data, both technical and biological variations have been modeled using negative binomial (over-dispersed Poisson) distributions (Robinson, et al., 2010).

Since the number of reads generated from each transcript depends on the length of the transcript and the depth of the sequencing, Mortazavi et al. (2008) introduced RPKM (Reads Per Kilobase of exon model per Million mapped reads) to normalize the estimated expression level of each transcript based on the length. RPKM, however, has several drawbacks. The fact that a small number of highly expressed genes can generate a big portion of the total reads (Bullard, et al., 2010) complicates normalization. It also has been reported that even after normalization based on length (*e.g.*, RPKM), longer transcripts or genes are still more prone to be called as differentially expressed than shorter ones using *t*-test (Oshlack and Wakefield, 2009). Moreover, expression levels of genes or transcripts cause bias in detecting differential expression; highly expressed genes or transcripts are more likely to be called as differentially expressed (Wu, et al., 2010). Non-uniform read coverage as results of experimental protocols and local sequence context also exists and some correction methods have been developed (Benjamini and Speed, 2012; Hansen, et al., 2010; Li, et al., 2010).

1.4 Methods Used to Analyze RNA-Seq Differential Gene Expression

Despite all the challenges, several methods have been developed to analyze DGE data over the last few years. They include DESeq (Anders and Huber, 2010), edgeR (Robinson, et al., 2010), Cuffdiff (Trapnell, et al., 2010), baySeq (Hardcastle and Kelly, 2010), TSPM (Auer and Doerge, 2011), BitSeq (Glaus, et al., 2012), and NOISeq (Tarazona, et al., 2011). Each of these methods is described next.

1.4.1 DESeq

DESeq (Anders and Huber, 2010) is a parametric statistics method based on a negative binomial model (an over-dispersed Poisson model; NB). It takes raw read counts as input. It assumes that the counts follow:

$$K_{ij} \sim NB(\mu_{ij}, \sigma_{ij}^2), \tag{1}$$

where K_{ij} is the number of read counts in sample *j* for gene *i,* and μ_{ij} and σ_{ij} are the mean and the variance of the distribution. Since in practice, μ_{ij} and σ_{ij} are unknown, they are estimated from the data. Since RNA-Seq experiments usually have only a small number of replicates, DESeq estimates the gene expression variance between replicates by pooling genes with similar expression levels to enhance the variance estimation. DESeq estimates the mean μ_{ij} as:

$$\mu_{ij} = q_{i,\rho(j)} S_j \tag{2}$$

where $q_{i,\rho(j)}$ is a condition-dependent per-gene parameter, which is proportional to the expected value of the unknown true concentration of fragments from gene *i* under the experimental condition $\rho(j)$ for sample *j*, and S_j is the sampling depth of sample *j*. The variance σ_{ij}^2 is the sum of a shot noise (technical variation) and a raw variance (biological variation):

$$\sigma_{ij}^2 = \mu_{ij} + S_j^2 \upsilon_{i,\rho(j)} \tag{3}$$

The shot noise is assumed to be Poisson distributed. Thus the variance equals the mean μ_{ij}. The per-gene raw variance parameter $\upsilon_{i,\rho}$ is a smooth function of $q_{i,\rho}$:

$$\upsilon_{i,\rho(j)} = \upsilon_\rho(q_{i,\rho(j)}) \tag{4}$$

Equation (4) is used to pool the data from genes with expression levels similar to gene *i* and to estimate the variance.

After fitting data to the model, DESeq weighs the evidence in the data for differential expression. Suppose that there are m_A replicates in condition A and m_B replicates in condition B. For each gene *i*, DESeq tests the null hypothesis $q_{iA}=q_{iB}$, where q_{iA} is the expression strength parameter (see equation (2)) for the samples in condition A, and q_{iB} for condition B. The total counts in each condition are defined as follows:

$$K_{iA} = \sum_{j:\rho(j)=A} K_{ij}, \qquad K_{iB} = \sum_{j:\rho(j)=B} K_{ij} \tag{5}$$

Then the overall sum is

$$K_{iS} = K_{iA} + K_{iB} \tag{6}$$

Anders and Huber (2010) showed that, under the null hypothesis, the probabilities of the events $K_{iA} = a$ and $K_{iB} = b$ for any given pair of numbers *a* and *b,* denoted as *p(a,b),* can be calculated. The *P* value of a pair of observed count sums (k_{iA}, k_{iB}) is calculated as the sum of all the probabilities less or equal to $p(k_{iA}, k_{iB})$ given the overall sum as k_{iS}:

$$p_i = \frac{\sum_{\substack{a+b=k_{iS} \\ p(a,b)\leq p(k_{iA},k_{iB})}} p(a,b)}{\sum_{a+b=k_{iS}} p(a,b)} \tag{7}$$

When no replicates are available, DESeq treats gene expressions between two experimental conditions as replicates (method="blind"). In this case, the determination of differentially expressed genes will be very conservative

Anders and Huber (2010) tested DESeq on the simulated read-counts generated from a negative binomial model and also on several real RNA-Seq datasets. They reported that DESeq controlled type-I error properly where the observed type-I error rate is at or lower than the DESeq claimed type-I error rate. They also compared DESeq with edgeR (described next). They showed that DESeq and edgeR both controlled type-I error well. DESeq produced more balanced results. DESeq reported a similar level of differentially expressed genes for both lowly and highly expressed genes, whereas edgeR reported more differentially expressed genes for lowly expressed genes and less differentially expressed genes for highly expressed genes.

1.4.2 edgeR

edgeR (Robinson, et al., 2010) is another parametric statistics method, which is also based on a negative binomial model. It was the first statistical method developed for DGE data, initially developed for "serial analysis of gene expression (SAGE)" data (Robinson and Smyth, 2008). The model formulation is very similar to that of DESeq. When estimating variances, DESeq and edgeR both borrow information between genes but in different ways. edgeR estimates the gene-wise variance or dispersion by conditional maximum likelihood conditioning on the total count for that gene (Smyth and Verbyla, 1996). It shrinks the dispersions towards a consensus value using an empirical

Bayes procedure (Robinson and Smyth, 2007). Differential expression is assessed by using an exact test similar to that of DESeq with modification for over-dispersed data for each gene. edgeR requires each condition having at least one replicate for input data.

One substantial difference between DESeq and edgeR is that, in their variance estimation, edgeR estimates a single common dispersion parameter for all genes, whereas DESeq estimates the variance using a more flexible, mean-dependent local regression. edgeR has the option to estimate per-gene empirical variance, which, as Anders and Huber (2010) pointed out, has little effect on the result; results are similar to those obtained using the common dispersion option.

1.4.3 Cufflinks/Cuffdiff

Cufflinks (v2.0.1, updated in June, 2012) (Trapnell, et al., 2012; Trapnell, et al., 2010) takes the aligned paired-end, as well as single-end, cDNA fragment sequences as input. It takes advantage of spliced alignments, which allows large gaps in the alignment, produced by, *e.g.*, TopHat (Trapnell, et al., 2009). It then assigns the fragments or short reads to different isoforms through following steps: 1) identifying short reads that must have originated from distinct spliced isoforms (mutually incompatible fragments), 2) assembling isoforms based on the mutually incompatible fragments, and 3) assigning compatible fragments to isoforms and estimating the isoform abundance that best explains the observed fragments (thus Cufflinks can align short reads that can be mapped to multiple isoforms). The expression estimation can be further improved by correcting for positional and sequence-specific biases. With positional bias, fragments (short reads) are preferentially located towards either the beginning or the end of transcripts. Some features of sequences also affect their probability of being selected for sequencing. Cufflinks incorporates the method described by Roberts et al. (2011) to correct these biases.

Cufflinks includes a program, Cuffdiff, for testing differentially expressed genes. Cuffdiff uses very similar procedure to that of DESeq. It estimates the variance based on a negative binomial model but uses *t*-test to calculate the test statistics instead. When no replicates are available, Cuffdiff treats two experimental conditions as if they were replicates just like DESeq. A unique property of Cuffdiff is that, with Cufflinks' ability for assigning read counts to multiple isoforms as mentioned above, it can evaluate expression at the transcript level. Cuffdiff uses a beta negative binomial model (a mixture of several negative binomial distributions) to estimate the variance of the count data of transcripts to include the uncertainty of assigning short reads to individual transcripts.

1.4.4 baySeq

baySeq (Hardcastle and Kelly, 2010) is also a parametric statistics method using a negative binomial model. baySeq takes a Bayesian approach where it assumes that non-differentially expressed genes should possess the same prior distribution on the underlying parameters across conditions (a model groups samples across conditions together), whereas differentially expressed genes should possess different parameters for prior distributions (a model separates different conditions). It compares the posterior probability of the observed data given the model that groups samples across conditions together (non-differentially expressed) with the posterior probability of the observed data given the model that separates samples between conditions (differentially expressed). One of the advantages of baySeq is that it enables the analysis of experimental designs with multiple groups (more than two conditions). baySeq requires replicates in each condition for input data.

The authors compared baySeq with DESeq and edgeR on simulated count data generated from a negative binomial model and real datasets. They showed that baySeq performed at least as well as or better (more sensitive at discovering differentially expressed genes at the same false discovery rate) than other methods in general. baySeq performed especially better when the dispersion of data was constant, the proportion of differentially expressed genes was high, or the differential expression was unidirectional (all differentially expressed genes are either over-expressed or under-expressed).

1.4.5 BitSeq

BitSeq (Glaus, et al., 2012) is a recently developed method based on a Bayesian approach. It utilizes the short-read mapping data where multiple-location information is available (using a program, such as Bowtie). Then, as with Cufflinks, it attempts to estimate transcript expression levels incorporating possibilities of having isoforms. BitSeq, unlike Cufflinks, estimates the distribution of transcript expression levels based on a probabilistic model of the read generation process and a Markov chain Monte Carlo (MCMC) algorithm. BitSeq estimates the variance in the transcript expression based on a hierarchical log-normal model and determines the probability of differential expression by Bayesian model averaging. BitSeq also requires replicates for each condition for input data. Compared with Cufflinks/Cuffdiff, DESeq, edgeR, and baySeq, the authors showed that, overall, BitSeq performed slightly better than baySeq, followed by DESeq and edgeR, and lastly Cufflinks/Cuffdiff.

1.4.6 Two-Stage Poisson Model (TSPM)

TSPM (Auer and Doerge, 2011) is another parametric statistics method. A unique feature of this method from aforementioned parametric methods is that it assumes that not all gene expressions are overly dispersed across samples. Therefore, the first stage of this method is to determine which gene-counts follow an over-dispersed Poisson model and which ones follow a simple Poisson model. It then applies a different likelihood ratio test for differential gene calling separately on the over-dispersed group and on the non-overdispersed group.

The authors compared their method with edgeR. TSPM performed better than edgeR when the data were derived from a simple Poisson distribution, but less sensitive when the data had an over-dispersed distribution. TSPM was not recommended if only a small number of replicates are available. TSPM requires replicates in each condition for input data.

Kvam et al. (2012) later compared DESeq, edgeR, and baySeq with TSPM methods on simulated data under various scenarios, *e.g.*, using 2 or 4 replicates, and using a Poisson or negative binomial model to generate count data. They reported a congruous result with previously mentioned comparison results; baySeq performed slightly better than DESeq and edgeR in general, and TSPM performed the poorest especially when only a few replicates were available.

1.4.7 NOISeq

NOISeq (Tarazona, et al., 2011), in contrast to aforementioned methods, is a non-parametric statistics method. Several normalization methods for the raw read counts are implemented with NOISeq. It includes: the number of read counts per million reads, RPKM (Mortazavi, et al., 2008), TMM (Robinson and Oshlack, 2010), and UQUA (Bullard, et al., 2010). "Trimmed mean of M-values" (TMM) calculates a normalization factor based on a weighted average expression ratio of all genes after removing extremely high and low counts data. UQUA calculates scaling factors based on per-lane upper-quartile (75th percentile) of all the gene counts excluding those that have zero counts for all lanes.

After normalization, it calculates the log-ratio (*M*) and the absolute value (*D*) of difference. Let x_g^i be the mean or median of the expression of gene *i* of all replicates in an experimental condition *g* (*g* =1 or 2). The statistics *M* and *D* for gene *i* are defined as:

$$M^i = \log_2\left(\frac{x_1^i}{x_2^i}\right) \tag{8}$$

$$D^i = \left| x_1^i - x_2^i \right| \tag{9}$$

These statistics collect the information on fold-change (*M*) as well as the absolute difference (*D*), which compensates the unstable behavior of *M* at low expression values. The probability of a gene being differentially expressed is the probability that both |*M*| and *D* are greater than the noise |*M**| and *D**, $p(\left|M^{*}\right| < \left|M^{i}\right|, \quad D^{*} < D^{i})$. *M** and *D** probability distributions are empirically computed by comparing gene expression counts between each pair of replicates within the same condition. The odds of differential expression to non-differential expression are calculated as:

$$\frac{p(\left|M^{*}\right| < \left|M^{i}\right|, \quad D^{*} < D^{i})}{1 - p(\left|M^{*}\right| < \left|M^{i}\right|, \quad D^{*} < D^{i})} \tag{10}$$

For example, if the odds value is 4:1, the probability of differential expression is equivalent to 0.8. We call this probability as P_{NOI} in this thesis.

When no replicates are available, NOISeq simulates replicates based on a multinomial distribution for read counts with parameters *n* (the number of replicates to be simulated), *pnr* (the number of the total reads for each replicate to be simulated expressed in a percentage of the total reads of the available sample), and *v* (the variability in the total read numbers of the simulated samples).

The authors compared NOISeq with several other methods including DESeq, edgeR, baySeq, and Fisher's Exact Test (FET). For both simulated count data and real datasets, they showed that NOISeq performed comparable to or better than other methods. While NOISeq found slightly fewer truly differentially expressed genes compared to other methods, the sensitivity of discovering differentially expressed genes by NOISeq was less dependent on the sequencing depth. The sensitivity of other methods increased with increasing sequencing depth resulting discovering more true positives. However, this was at the cost of having significantly more false positives compared to NOISeq.

1.5 Objectives of This Study

The simulated count data used in all previously mentioned comparative studies except for the study by Glaus et al. (2012) were directly simulated from a Poisson or a negative binomial model. However, in the real RNA-Seq analysis, some short reads can be mapped to more than one gene. Such non-uniquely mapped short reads are usually discarded and are not counted. This practice could affect the relationship between expected gene expression levels and the actual read counts obtained. One objective of this study is to include the effect of uncertainty in short-read mapping, therefore simulation was done following the RNA-Seq sequencing process and data analysis procedure step-by-step.

Another focus of this study different from others is that we compared the consistency and performance (sensitivity, FDR, precision and recall) of methods when no replicates were available. Although it is important and highly recommended to have biological replicates, in practice, the majority of RNA-Seq experiments include no or very few replicates. By testing the method performance with and without replicates, we examined if these methods could recover any meaningful results even when no replicates were included.

Based on multiple simulated and real datasets, we compared the performance of a parametric, DESeq, and a non-parametric, NOISeq, differential gene-calling methods.

Finally, instead of choosing one method to analyze all kinds of data, we attempted to develop strategies regarding how to better apply these methods or combine their results based on the characteristics of the data.

2 Materials and Methods

2.1 RNA-Seq Simulation

2.1.1 Overall process of the simulation

Simulated read-count data have been often used for testing performance of differential gene-calling methods (e.g., Auer and Doerge, 2011; Glaus, et al., 2012; Hardcastle and Kelly, 2010; Kvam, et al., 2012). In these studies, except for Glaus et al. (2012), count data were simulated from a defined distribution given the expected expression level of each gene in each condition. However, such simulation studies cannot incorporate uncertainties generated during the actual RNA-Seq studies. For example, it is a common practice in RNA-Seq analysis to only consider short reads that can be uniquely mapped to the reference sequences and discard the rest that maps multiple genomic locations or multiple transcripts. This practice may affect the relationship between the expected expression level and the actual count data obtained, which may cause the count data not following the defined distribution. In order to examine how discarding non-uniquely mapped reads affects the results of RNA-Seq analysis, instead of only simulating the count data, we simulated the entire RNA-Seq process step by step. The overall simulation process is summarized in Figure 1. Each process is described next.

(1) We used the consolidated set of protein-coding sequences (CDS) gathered from the mouse genome as our transcriptome (see Supplementary Material section S3 for the description of the consolidated mouse CDS set). It included 26,017 transcripts after excluding alternative splicing forms. Note that the terms “gene”, “transcript”, and

"CDS" are used interchangeably in this thesis. They all mean the sequences that are used for generating short reads and used as the references to be mapped by short reads subsequently.

(2) Each gene was randomly assigned an expression level from a Gamma distribution (described in section 2.1.2).

(3) Short reads with their length of 36bp were generated from each gene starting at random positions on that gene. Note that no errors (sequencing errors) were introduced in this process. The number of short reads generated for each gene was set to be proportional to the expression level and the length of the gene (described in section 2.1.3). The script that generated the starting positions of short reads can be found in Supplementary Material section S1.

(4) Short reads generated were mapped back to the mouse reference sequences (our consolidated CDS dataset) by using SOAP2 (Li, et al., 2009) allowing 2 mismatches. We chose SOAP2 for its good performance reported in Ruffalo et al. (2011) as described in section1.2. Following the common practice, we only considered short reads that can be uniquely mapped back to the reference sequences using the option "–r 0". In this setting, short reads that were mapped more than one location were discarded.

(5) These steps were repeated for each replicate of each experimental condition.

(6) Finally, the number of short reads mapped to each gene was used as the count input for differential gene expression analysis.

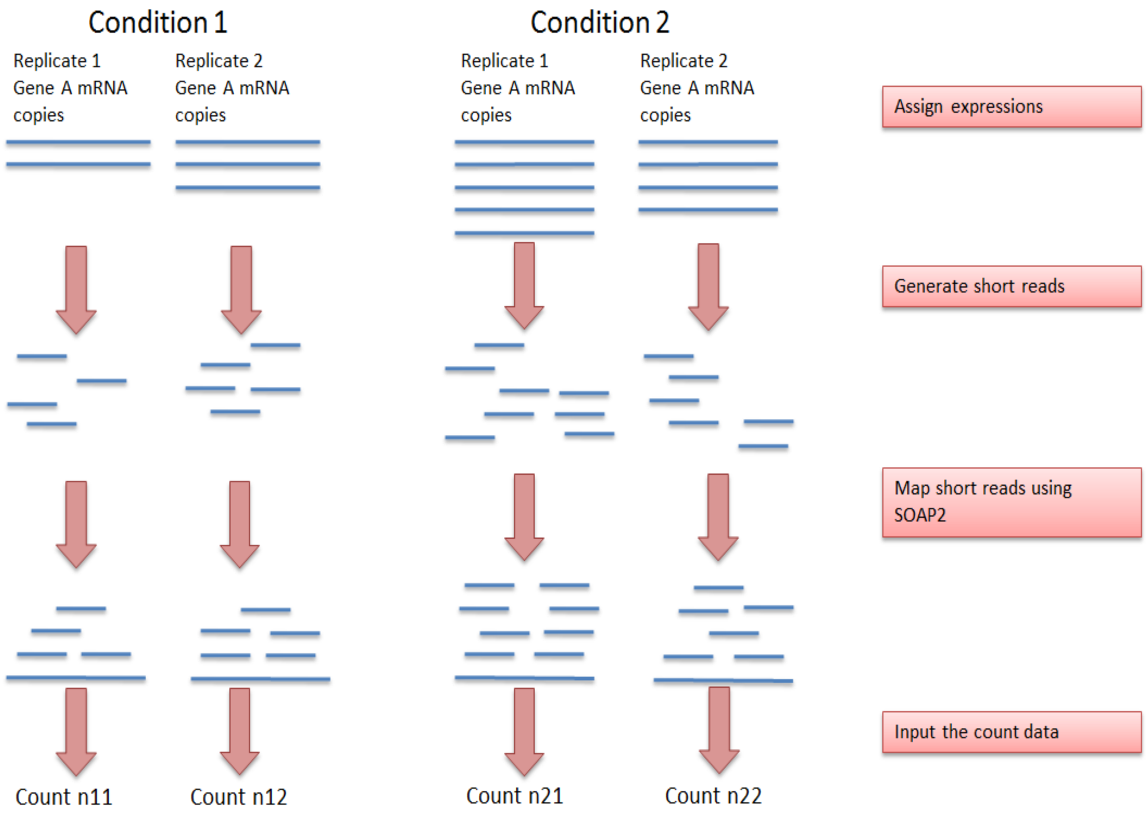
Condition 1
Replicate 1 Gene A mRNA copies
Replicate 2 Gene A mRNA copies
Condition 2
Replicate 1 Gene A mRNA copies
Replicate 2 Gene A mRNA copies
Assign expressions
Generate short reads
Map short reads using SOAP2
Input the count data
Count n11
Count n12
Count n21
Count n22

Figure 1: Workflow of data simulation.

2.1.2 Modeling gene expressions

The expression level of each gene at the control condition was assigned randomly from a Gamma distribution with the shape parameter 0.15 and the scale parameter 1160. The parameters were chosen to produce a distribution that was similar to the distributions of our real RNA-Seq datasets (described in sections 2.4 and 2.5). Figure 2 shows the distribution of gene expression (read count) of a real mouse dataset. Figure 3 shows the distribution of our simulated gene expression (read count) generated based on the above Gamma distribution.

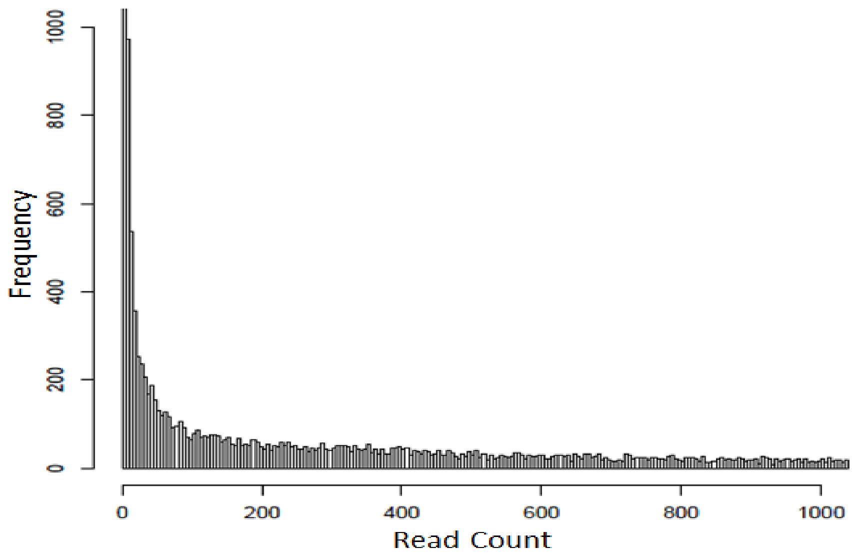

Figure 2: Distribution of the read counts obtained from the real mouse RNA-Seq dataset.

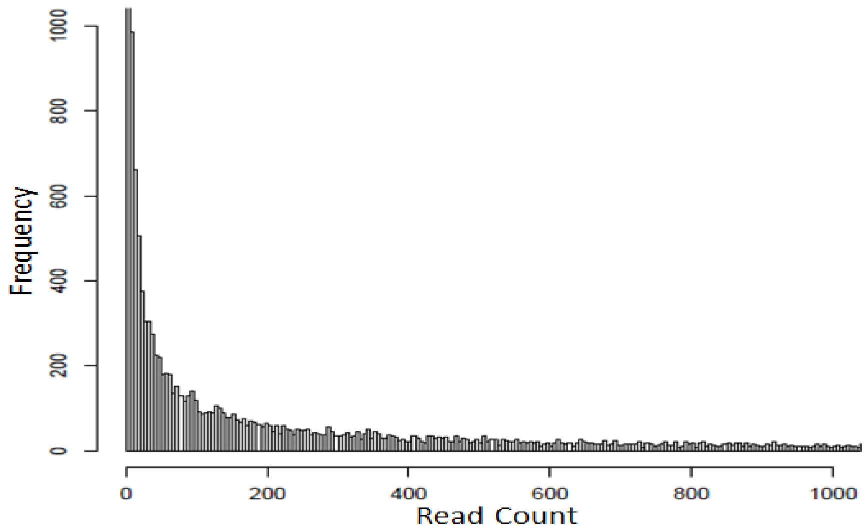

Figure 3: Distribution of the read count generated in our simulated RNA-Seq dataset.

Our simulation strategy is summarized in Table 1. Assuming that most genes are not differentially expressed, we assigned 10% of genes (type A) to be “over-expressed” and another 10% (type B) to be “under-expressed” in the experimental condition. For both over- and under-expressed genes, the fold-changes were chosen randomly from 1.1 - 5.0. The remaining 80% of the genes (type C) were considered to have no difference in expected expression levels between the control and experimental conditions. Since RNA-Seq experiments often include only a few replicates, we included only two replicates in each of the experimental (“Exp1” and "Exp2") and control (“Ctr1” and "Ctr2") conditions.

Another set of data was also simulated for testing experiments without replicates (one replicate in experimental condition and one replicate in control condition).

Table 1: Simulation strategy used in this study.

Gene types	Number of Genes (26,017)[a]	Gene expression levels[b]			
		Exp1	**Exp2**	**Ctr1**	**Ctr2**
A	2,602	Over	Over	Normal	Normal
B	2,602	Under	Under	Normal	Normal
C	20,183	Normal	Normal	Normal	Normal

[a] Total number of genes are shown in parentheses.

[b] “Over”: over-expressed, “Under”: under-expressed, and “Normal”: no-differential expression.

2.1.3 Modeling technical and biological variations

The biological variation between replicates within each condition group was modeled by a Gamma distribution. Two datasets were generated with different levels of variations. The dataset with a *moderate* variation had 0.33 of the coefficient of variation (CV) modeling after the *Chlamydomonas reinhardtii* dataset described in section 2.4.2. The *large* variation dataset had 0.67 of CV modeling the *Acyrthosiphon pisum* dataset described also in section 2.4.2. For gene *I*, its expression level is expressed as:

$$\lambda_i \sim Gamma(k_i, \theta_i) \tag{11}$$

where k_i is the shape parameter and θ_i is the scale parameter of the Gamma distribution. The technical variation was modeled by a Poisson distribution. Thus the expression level of gene i, E_i, after considering both biological and technical variations can be expressed as:

$$E_i \sim Pois(\lambda_i) \tag{12}$$

The number of short reads generated from each gene was assumed to be proportional to the expression level and the length of the gene to mimic the real RNA-Seq process:

$$N_i = c \times E_i \times L_i \tag{13}$$

where N_i is the number of short reads generated for gene I, L_i is the length of gene i, and c is a constant to make desired amount of total reads in the experiment. We chose a value for c to generate the approximate total of 23 million short reads for each replicate.

2.2 Differential Gene Calling Methods Compared

One of our focuses in this study was to examine how different differential gene-calling methods perform when there was no replicate as it is usually the case in many RNA-Seq experiments. Based on this focus, we chose a popular parametric statistics method DESeq (version 1.2.1) and a relatively newly introduced non-parametric statistics method NOISeq (R script downloaded on Feb 21, 2012 from http://bioinfo.cipf.es/noiseq/doku.php?id=downloads). While DESeq, edgeR, and baySeq are all based on negative binomial models, DESeq can handle experimental data without replicates with a straightforward option. The recent version of Cuffdiff and BitSeq were released at the time of writing; thus they were not included in the study. As mentioned before, Auer and Doerge (2011) reported that TSPM does not perform well when the number of replicates is small. TSPM also requires replicates. NOISeq represents a completely different approach (non-parametric) and can handle experimental data without replicates.

When testing the performance of each method without replicates, for DESeq, the option “method” for variance estimation was set to “blind”. For NOISeq, we used the recommended parameter values n=5 and pnr=0.2, but for the parameter v, we used 0.2. This value was chosen since it produced best results for preliminary analysis. We used the default parameter values for DESeq and NOISeq when replicates were available. DESeq takes raw count data as input. We used RPKM as the normalization method for NOISeq input data.

2.3 Test Statistics

In our simulations, as shown in Table 1, type-A and -B genes were set to be differentially expressed (“actual positives”), and type-C genes were set to be non-differentially expressed (“actual negatives”). We compared the list of these genes with those determined to be differentially expressed by DESeq and NOISeq at various thresholds. Results were classified as follows:

- True Positive (TP): genes set to be differentially expressed and called as differentially expressed by the method,
- True Negative (TN): genes set to be non-differentially expressed and not called as differentially expressed by the method,
- False Positive (FP): genes set to be non-differentially expressed but called as differentially expressed by the method, and
- False Negative (FN): genes set to be differentially expressed but not called as differentially expressed by the method.

The performance of the methods was evaluated as follows:

$$Sensitivit\ y = \frac{TP}{TP + FN} \tag{14}$$

$$False \quad Dis\,\mathrm{cov}\,ery \quad Rate \quad (FDR) = \frac{FP}{TP + FP} \tag{15}$$

$$Precision = 1 - FDR = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = Sensitivity = \frac{TP}{TP + FN} \tag{17}$$

Equation (15) is used to calculate the empirical false discovery rate from the observed data. For simplicity, we call it as “False Discovery Rate” or FDR.

2.4 Tests using the real RNA-Seq data

We also tested DESeq and NOISeq on three sets of real RNA-Seq data. Human embryonic kidney and Ramos B cells data were published by Sultan et al. (2008). Two unpublished RNA-Seq datasets were by courtesy of our collaborators: a green alga *Chlamydomonas reinhardtii* dataset (Cerutti et al., in preparation) and a pea aphid *Acyrthosiphon pisum* dataset (Brisson et al., unpublished).

2.4.1 Sultan et al.'s dataset

Sultan et al. (2008) performed differential gene expression analysis between human embryonic kidney- and Ramos B-cell lines using both RNA-Seq and microarray experiments. We extracted the short-read count data for 13,118 genes from their RNA-Seq study considering only the hits on the exons (total read numbers ranging from around 5 to 7 millions) (data obtained from their supplemental material Table S2). Their experiments included two biological replicates. However, these data were combined in their study due to high correlation between replicates within each cell line. In their analysis, they only focused on genes that were expressed in both microarray and RNA-Seq platforms. Genes with detection score (defined as the rank of the probe bead signal relatively to the negative control bead signals divided by the number of negative controls on the chip field) greater than or equal to 0.95 in microarray experiment and at least 5 hits in merged data in both conditions in RNA-Seq experiment were considered expressed (7043 genes). Then they applied *t*-test to find differentially expressed genes. We followed their procedure and analyzed their RNA-Seq data (7043 genes) using both DESeq and NOISeq as if the data were without replicates due to the merge of the data. The accuracy of these methods were tested against the results obtained by Sultan et al. based on their microarray analysis (their supplemental material Table S4; q-value = 0.01 as the cutoff) for the same set of genes. We considered the microarray results as bases of our comparison, defining the "actual" positives and negatives. Then the precision and recall were calculated following the equations (16) and (17).

2.4.2 *Chlamydomonas* and pea aphid datasets

The *Clamydomonas* dataset compared the expression of 16,865 *C. reinhardtii* genes between the control and the nitrogen-starvation experiment (144-hour time point) (Cerutti et al., in preparation). Each condition included 2 replicates (total read numbers ranging from around 20 to 30 million). As mentioned before, this dataset had a moderate level of variation (0.33 of CV). The pea aphid (*A. pisum*) dataset compared 35,884 genes between the control and solitary conditions (8-hour time point) (Brisson et al., unpublished). For this dataset, 3 replicates were included for each condition (total read numbers ranging from around 2 to 3 million). The level of variation was twice larger than that of the *Clamydomonas* dataset (0.67 of CV).

We used these datasets to test "consistency" in the results obtained by DESeq and NOISeq between when replicates were available and not available. Differentially expressed genes were first identified using all replicates. These results were used as the "standards" for the comparative purpose. We next analyzed the RNA-Seq data assuming no replicate. Then the precision and recall were calculated following the equations (16) and (17). Since there are multiple replicates for both control and experimental conditions, we took the average statistics from all pairwise comparisons and reported the “average precision” and the “average recall”.

For example, suppose that both control and experimental conditions have two replicates: Exp1 and Exp2, and Ctr1 and Ctr2. Suppose further that using one of the methods (*e.g.*, DESeq), we find 400 differentially expressed genes considering all replicates. These 400 genes are considered to be "actual positives". If 80 differentially expressed genes are identified using the comparison of “Exp1 *vs.* Ctr1” (without considering replicate) and 20 of them are overlapped with the 400 "actual positive", the precision from this “Exp1 *vs.* Ctr1” comparison is calculated as 20/80 =0.25. The recall from this comparison is calculated as 20/400=0.05. We repeat this procedure for all other no-replicate comparisons (“Exp1 vs. Ctr2”, “Exp2 vs. Ctr1”, and “Exp2 vs. Ctr2”). Finally, we take the average of the precisions of the total four comparisons. We calculate the “average recall” in a similar fashion.

Note that these "average precision" and "average recall" were used to measure the consistency in the results obtained with and without having replicates. Since for these actual RNA-Seq data, we do not know the *true positives* and *true negatives*, these statistics are used by no means to indicate the accuracy of the methods.

3 Results and Discussion

3.1 Effects of Bias on Differential Gene Calling

3.1.1 Expression-level dependency

We first examined if differential gene calling is dependent on gene-expression levels. We used the simulated dataset with the *moderate* variation (CV=0.33) for this analysis. Using ranges of thresholds, sensitivities (equation (14)) were calculated separately for two groups of genes: over-expressed (type A in Table 1) and under-expressed (type B in Table 1) genes. For DESeq, *q*-value (FDR adjusted *p*-value) was used for the threshold. For NOISeq, P_{NOI} (the probability of a gene being differentially expressed provided by NOISeq; see section 1.4.7) was used for the threshold. When both being used as a threshold, *q*-value is roughly comparable to the probability of equivalent expression (1- P_{NOI}) (Kall, et al., 2008). As shown in Tables 2 and 3, we observed expression-level dependent results with both DESeq and NOISeq. Both methods showed slightly higher sensitivities for the over-expressed genes than for the under-expressed genes. Over-expressed genes were slightly more likely to be correctly called as differentially expressed than under-expressed genes. This is consistent with the results reported by Wu et al. (2010).

Table 2: Sensitivity of DESeq gene-calling and gene-expression levels.

	q-value threshold							
Gene group	**0.005**	**0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Over-expressed	0.21	0.26	0.36	0.44	0.51	0.57	0.63	0.66
Under-expressed	0.19	0.23	0.32	0.39	0.46	0.52	0.57	0.62

Table 3: Sensitivity of NOISeq gene-calling and gene-expression levels.

	P_{NOI} threshold							
Gene group	**0.995**	**0.99**	**0.95**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**
Over-expressed	0.00	0.00	0.00	0.05	0.17	0.28	0.38	0.47
Under-expressed	0.00	0.00	0.00	0.03	0.14	0.25	0.35	0.46

3.1.2 Gene-length dependency

As pointed out by Oshlack and Wakefield (2009), RNA-Seq differential calling shows also gene-length dependency. We examined if this length-dependency was also present in the results calculated by DESeq and NOISeq using again our simulated dataset with the *moderate* variation. We binned the results of differential gene calling according to the length of the transcript sequence (100bp increment) and calculated the proportion of the genes being called differentially expressed for each bin. As shown in Figure 4, DESeq had length-dependency where longer transcripts were more likely to be called as differentially expressed, whereas NOISeq did not exhibit such dependency. In order to examine the proportion of true positives among the genes called as differentially expressed, in Figure 5, precisions are plotted. Precision also showed a length-dependency for DESeq where precision decreased with gene length. NOISeq showed a consistently high precision regardless the length of the gene. Figures 4 and 5 indicate that for longer genes, DESeq calls more genes as differentially expressed, but their results include more false positives. In contrast, NOISeq calls a smaller number of genes as positives, but with very high accuracy, regardless of the lengths.

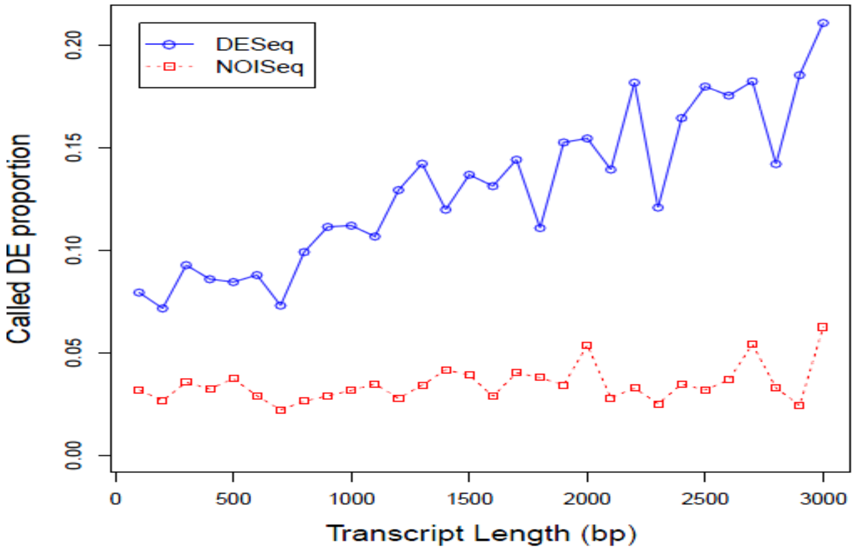

Figure 4: Gene lengths and differential gene-calling.
The transcripts are binned according to their lengths with 100bp increment. For each bin, the proportion of transcripts called differentially expressed by each method is plotted. The thresholds used are q=0.2 for DESeq and P_{NOI}=0.8 for NOISeq. The same patterns were observed when different threshold values were used.

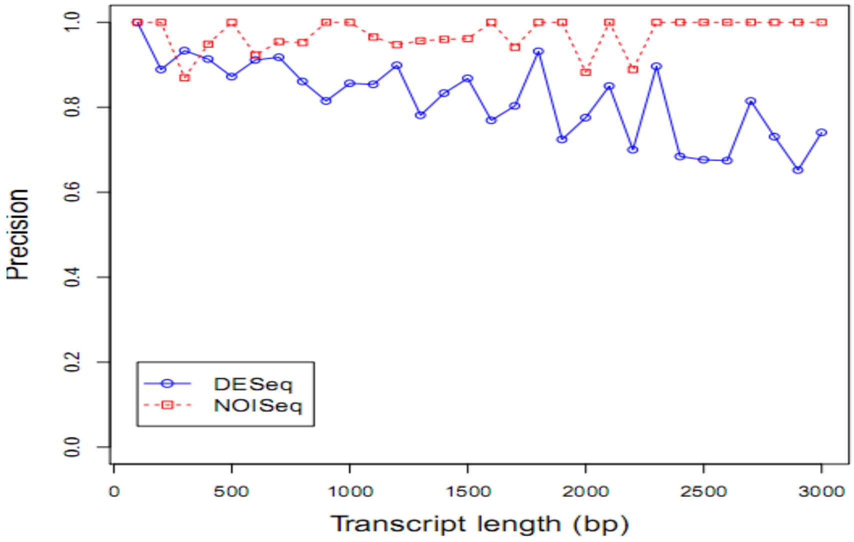

Figure 5: Gene lengths and differential gene-calling accuracy.
The transcripts are binned according to their lengths with 100bp increment. For each bin, precision of calling differentially expressed genes by each method is plotted. The thresholds used are q=0.2 for DESeq and P_{NOI}=0.8 for NOISeq. The same patterns were observed when different threshold values were used.

3.2 Gene-calling Performance and Biological Variation

As mentioned before (section 2.1.3), the biological variation can be quite large in RNA-Seq data. In order to study if and how the variation in the data affects the performance of differential gene calling, we analyzed two simulated datasets that modeled two levels of biological variation: *moderate* (CV= 0.33) and *large* (CV=0.67) datasets. The sensitivities and false discovery rates were calculated using the equations (14) and (15).

3.2.1 False discovery rate (FDR) control

DESeq calculates a *q*-value (FDR adjusted *p*-value) for each gene and uses it as the threshold to identify differentially expressed genes. If the method controls FDR well, the *q*-value threshold should equal to or greater than the observed FDRs. As shown in Figure 6, DESeq controlled FDRs more reliably when the biological variation was *moderate* compared to when the biological variation was *large*. With the *large* variation, observed FDRs were significantly larger than reported *q*-values especially for *q*-values smaller than 0.2. This result is consistent with the one obtained by Kvam et al.(2012). FDR was not controlled in their "Simulation 4" where variation was large.

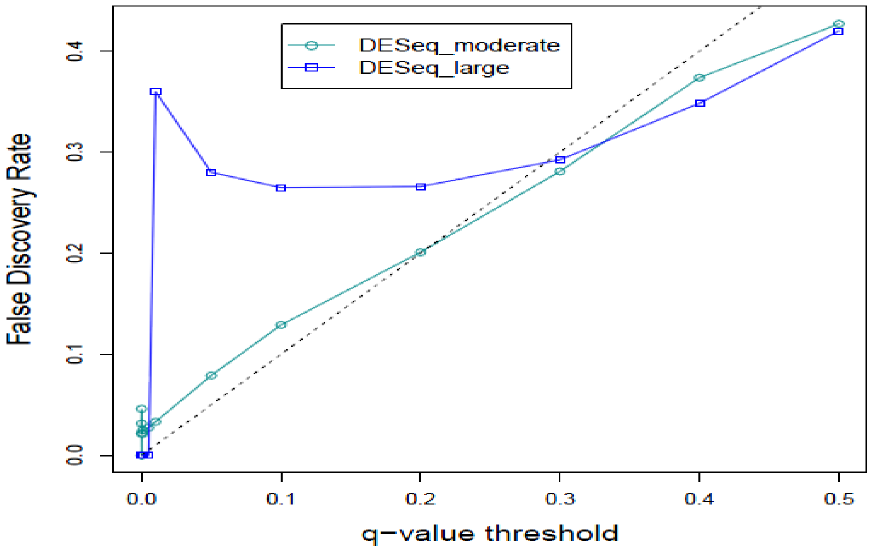

Figure 6: False discovery rate observed for DESeq at different q-value thresholds with moderate and large biological variation.
The black dashed-line is where the observed FDRs equal q-value thresholds.

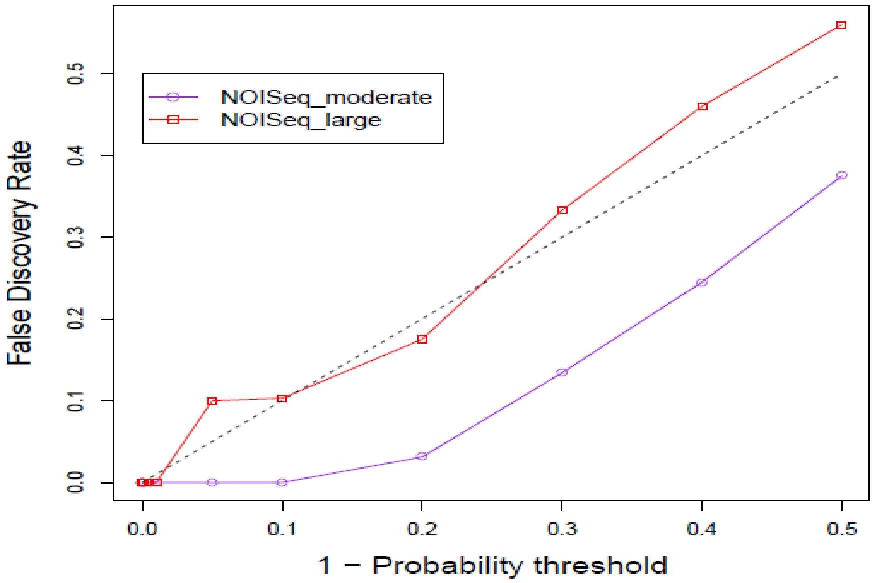

Figure 7: False discovery rate observed for NOISeq at different 1-P_{NOI} thresholds with moderate and large biological variation.
The black dashed-line is where the observed FDRs equal q-value thresholds.

NOISeq calculates the probability (*P_{NOI}*) to identify differentially expressed genes. As mentioned before, we can consider 1-P_{NOI} to be equivalent to *q*-value (Kall, et al., 2008). As shown in Figure 7, although observed FDRs were consistently larger when the biological variation was large, NOISeq roughly controlled the FDR regardless of the level of variation. In fact when the variation is moderate, the observed FDRs were much lower than *1-P_{NOI}* values. DESeq and NOISeq both had much larger false discovery rates when the biological variation was large.

3.2.2 Effect of biological variation on differential gene-calling

Next we compared the effect of biological variation on the performance of differential gene-calling by DESeq and NOISeq. As shown in Figure 8, with the *moderate* variation, DESeq performed much better (higher sensitivity with the same FDR) with the *q*-value threshold greater than 0.005 as indicated by the large gap between the curves. NOISeq performed better with the P_{NOI} threshold greater than 0.8. With the *large* variation, as shown with Figure 9, DESeq performed better with the *q*-value threshold greater than 0.3 and NOISeq performed better with the P_{NOI} threshold greater than ~0.7. Both DESeq and NOISeq showed that with the same *q*-value or P_{NOI} thresholds, FDRs were higher when the biological variation was larger.

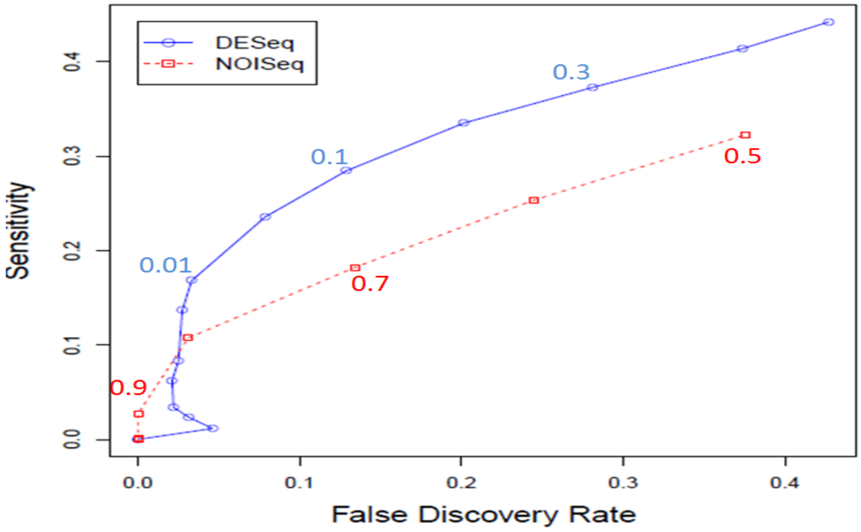

Figure 8: Gene-calling performance on the moderate dataset.
Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles) and NOISeq (red squares) for the dataset with the moderate variation. DESeq and P_{NOI} thresholds used are shown for some data points.

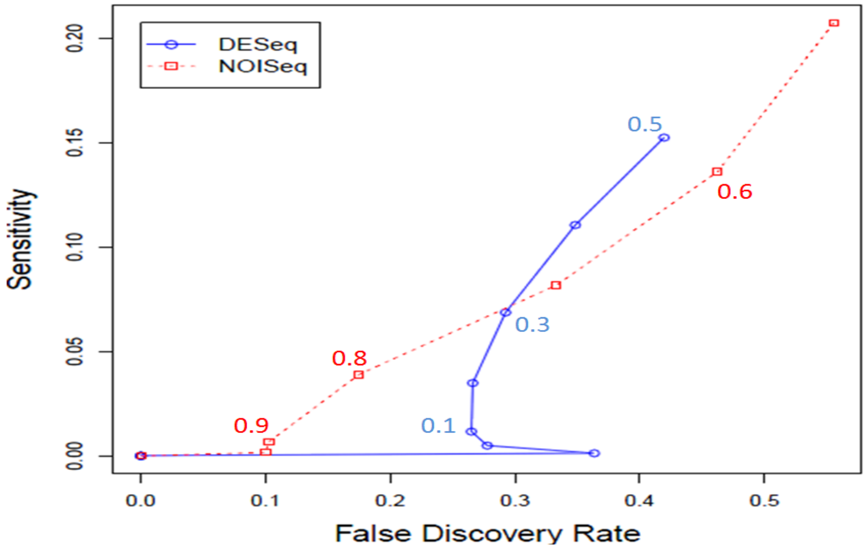

Figure 9: Gene-calling performance on the large dataset.
Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles) and NOISeq (red squares) for the dataset with the large variation. DESeq and P_{NOI} thresholds used are shown for some data points.

3.2.3 Effect of uncertain read-mapping

In our simulation process, read-mapping uncertainty was naturally incorporated. Around 90% of short reads were mapped back to the reference sequences. The 10% reads that were not mapped to the reference were the reads that were mapped to multiple locations on the reference. We suppose more reads would be discarded if sequencing errors were introduced and less reads would be discarded if reads were longer. We compared the results obtained by DESeq and NOISeq using data from two simulation processes: simulating the entire RNA-Seq process and simulating count data directly as many previous studies have done. As shown in Tables 4-7, both DESeq and NOISeq showed slightly better performance (larger or at least the same sensitivity) when count data were simulated directly compared to when simulations were done following the entire process. Uncertainty in read mapping process, therefore, affected the performance of the program, although the differences shown with our examples were small.

Table 4: Sensitivities and FDRs observed by DESeq using two simulation processes on moderate-variation data.

	q-value threshold					
	0.001[a]	0.005	0.01	0.05	0.1	0.2
Entire process	0.08 (0.02)	0.14 (0.03)	0.17 (0.03)	0.24 (0.08)	0.28 (0.13)	0.36 (0.20)
Direct count	0.09 (0.02)	0.15 (0.02)	0.18 (0.03)	0.26 (0.08)	0.31 (0.12)	0.36 (0.20)

[a] Values in parentheses are FDRs.

Table 5: Sensitivities and FDRs observed by NOISeq using two simulation processes on moderate-variation data.

	P_{NOI}-threshold					
	0.999[a]	0.995	0.99	0.95	0.9	0.8
Entire process	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.001 (0.00)	0.03 (0.00)	0.108 (0.03)
Direct count	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.001 (0.00)	0.03 (0.00)	0.114 (0.03)

[a] Values in parentheses are FDRs.

Table 6: Sensitivities and FDRs observed by DESeq using two simulation processes on large-variation data.

	***q*-value threshold**					
	0.001	0.005	0.01	0.05	0.1	0.2
Entire process	0.00 (0.00)	0.00 (0.00)	0.001 (0.36)	0.005 (0.28)	0.01 (0.27)	0.03 (0.27)
Direct count	0.00 (0.00)	0.00 (0.50)	0.001 (0.25)	0.005 (0.29)	0.01 (0.27)	0.04 (0.26)

[a] Values in parentheses are FDRs.

Table 7: Sensitivities and FDRs observed by NOISeq using two simulation processes on large-variation data.

	P_{NOI}-threshold					
	0.999	0.995	0.99	0.95	0.9	0.8
Entire process	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.002 (0.10)	0.01 (0.10)	0.039 (0.17)
Direct count	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.002 (0.08)	0.01 (0.12)	0.042 (0.16)

[a] Values in parentheses are FDRs.

3.2.4 Effect of replications on differential gene-calling

We next examined the performance of DESeq and NOISeq on the simulated datasets where no replicates were used. Compared to the results shown in Figures 8 and 9, when no replicates were available, as shown in Figures 10 and 11, the overall accuracy for both methods decreased dramatically as expected and the false discovery rates were very large at all thresholds. DESeq found hardly any truly differentially expressed genes when no replicates were available. For example, while DESeq had sensitivity about 0.25 in Figure 8 at the *q*-value threshold of 0.05, in Figure 10, sensitivity was 0 at the same threshold. NOISeq still found truly differentially expressed genes however at a cost of having many false positives. For example, NOISeq had 0.1 sensitivity and FDR = 0.03 at the $P_{NOI} = 0.8$ threshold in Figure 8, but 0.17 sensitivity and FDR = 0.23 in Figure 10 at the same threshold. DESeq was conservative in calling differentially expressed genes when no replicates were available, whereas NOISeq was much more aggressive. Similar to when replicates are available, when variation was large, both methods performed worse as shown in Figure 11. Based on the results, it is highly recommended to have replicates in RNA-Seq experiments. However, if no replicates are available, NOISeq may serve better as a starting point of analysis.

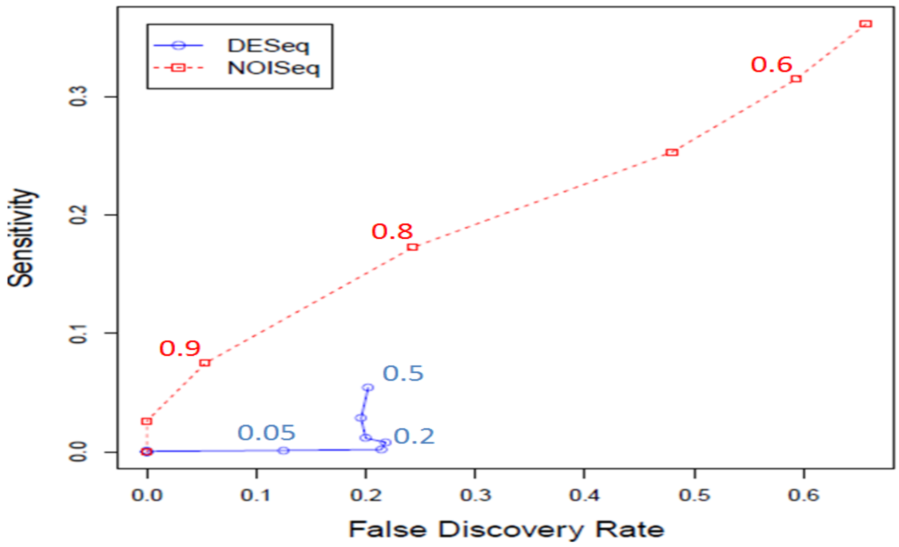

Figure 10: Gene-calling performance on the moderate-variation dataset and with no replicate.

Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles) and NOISeq (red squares) for the dataset with the moderate variation.

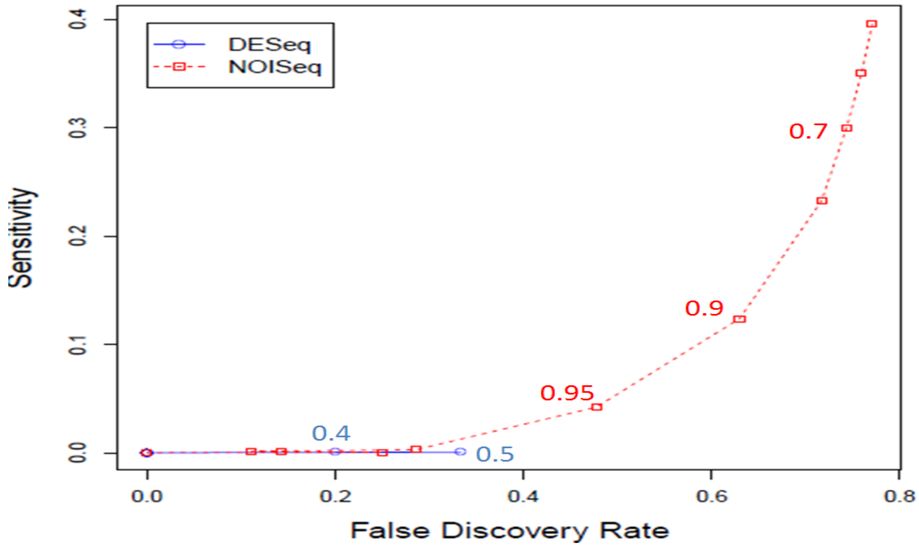

Figure 11: Gene-calling performance on the large-variation dataset and with no replicate.
Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles) and NOISeq (red squares) for the dataset with the large variation.

3.2.5 Combining results of DESeq and NOISeq to improve differential gene-calling

We next examined if we could improve the accuracy in identifying differentially expressed genes by combining the results obtained by DESeq and NOISeq. We tested ranges of combinations of thresholds (q-value for DESeq and P_{NOI} for NOISeq). At each threshold combination, positives (differentially expressed genes) were identified for those called by both methods (*i.e.*, by taking the intersection of both results). In order to find the best performing combination, we calculated the sensitivity to FDR ratio for the result at each combination. The results of the entire combination analysis are shown in Supplementary Material section S2. Results are summarized in Tables 8 and 9 below. For the *moderate*-variation data, as shown in Table 8, when using DESeq and NOISeq individually, the highest sensitivity to FDR ratios were 5.67 and 1.04, respectively.

Using the combination strategy, the combination with 0.005 *q*-value threshold for DESeq and 0.6 P_{NOI} threshold for NOISeq generated the highest sensitivity to FDR ratio, 60. This is significantly better than individual results. Figure 12 also shows that the combination strategy can improve the performance by lowering FDRs significantly. For the *large*-variation data, as shown in Table 9, combination with 0.4 *q*-value threshold for DESeq and 0.5 P_{NOI} threshold for NOISeq generated the highest sensitivity to FDR ratio among all combinations. Although using NOISeq with 0.5 P_{NOI} generated 0.38 sensitivity to FDR ratio, the FDR is too high (0.56, see Table S2) to use in practice. Therefore, we still think using the combination strategy is a better choice. However, sensitivities and ratios are all very low with such large variation. As shown in Figure 13, combination strategy can still improve the performance especially at lower FDRs. Therefore, it is possible to produce better result by combining two methods using appropriate threshold combinations: *e.g.*, (q, P_{NOI}) = (0.005, 0.6) for *moderate*-variation data, or (q, P_{NOI}) = (0.4, 0.5) for *large*-variation data. We also tried combined strategy with no-replicate data. However, combined strategy did not improve the accuracy of calling differentially expressed genes when no replicates were available.

Table 8: Combination strategy on moderate-variation data.

	Threshold combination					
***q*-value**	0.001 (4)[a]	0.005 (4.67)	0.01 (5.67)	0.05 (3)	0.1 (2.15)	0.2 (1.7)
$\boldsymbol{P_{NOI}}$	0.6 (1.04)	0.6 (1.04)	0.6 (1.04)	0.6 (1.04)	0.5 (0.84)	0.9 (-)
Sensitivity	0.075	0.12	0.147	0.192	0.247	0.028
FDR	0.003	0.002	0.003	0.024	0.045	0.007
Sensitivity/FDR	25	**60**	49	8	5.49	4

[a]Value in parenthesis is the sensitivity to FDR ratio when using the corresponding method along. ‘-‘ indicates that the sensitivity or FDR is 0.

Table 9: Combination strategy on large-variation data.

	Threshold combination					
***q*-value**	0.05 (0.02)[a]	0.1 (0.04)	0.2 (0.11)	0.3 (0.24)	0.4 (0.31)	0.5 (0.36)
***P*$_{NOI}$**	0.5 (0.38)	0.5 (0.38)	0.5 (0.38)	0.5 (0.38)	0.5 (0.38)	0.5 (0.38)
Sensitivity	0.003	0.009	0.028	0.057	0.089	0.115
FDR	0.053	0.113	0.131	0.174	0.238	0.321
Sensitivity/FDR	0.06	0.08	0.21	0.33	**0.37**	0.36

[a]Value in parenthesis is the sensitivity to FDR ratio when using the corresponding method alone.

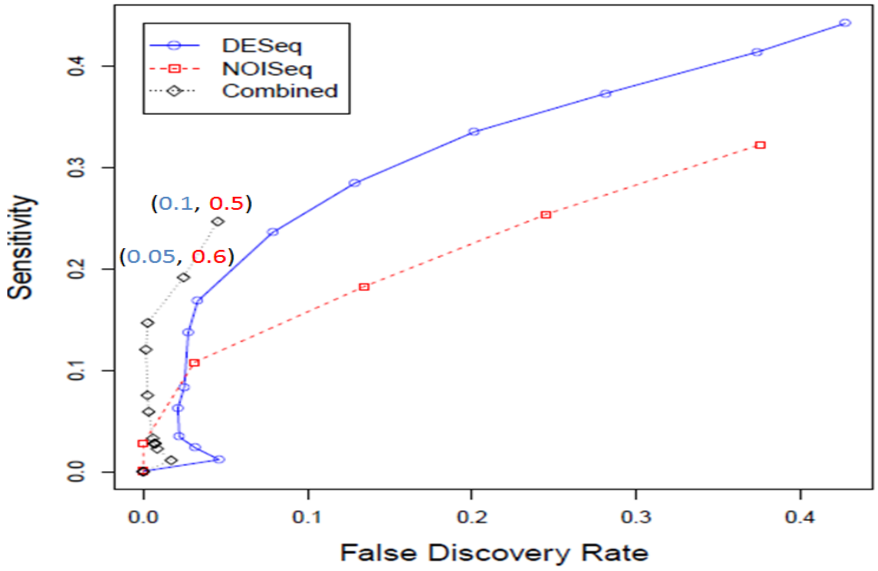

Figure 12: Performance of combination strategy on the moderate-variation dataset.

Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles), NOISeq (red squares) and Combined method (black diamond). Combinations of q-value threshold and P$_{NOI}$ threshold used are shown for some data points.

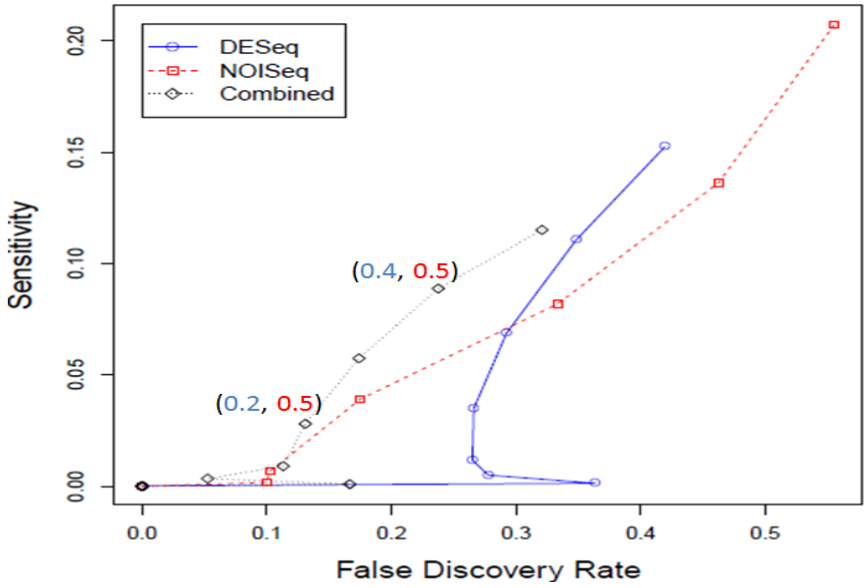

Figure 13: Performance of combination strategy on the large-variation dataset.
Sensitivities are plotted against the false discovery rates calculated from the results obtained by DESeq (blue circles), NOISeq (red squares) and Combined method (black diamond). Combinations of q-value threshold and P_{NOI} threshold used are shown for some data points.

3.3 Performance Analysis on the Real Data

3.3.1 Comparing the results between RNA-Seq and microarray analyses

We tested the performance of DESeq and NOISeq on the real RNA-Seq datasets published by Sultan et al. (2008). Since both microarray and RNA-Seq data on the same sample are available, we considered the microarray result as the reference defining "actual positives" (differentially expressed) and "actual negatives" (not differentially expressed). We also only considered the "expressed" genes (7043) following their definition. Precision and recall were calculated as described in section 2.4.1.

As shown in Tables 10 and 11, both methods had almost 80% or higher precision. However, their recall values were very low (lower than 0.03 for DESeq and lower than 0.27 for NOISeq). At the most stringent thresholds (q-value threshold = 0.001 and P_{NOI} threshold = 0.999), DESeq showed higher precision but lower recall than those of NOISeq. At less stringent thresholds ($0.005 \leq q$-value threshold ≤ 0.05 and $0.995 \geq P_{NOI}$ threshold ≥ 0.95), NOISeq showed both higher precision and recall. At more relaxed thresholds ($0.1 \leq q$-value threshold ≤ 0.2 and $0.9 \geq P_{NOI}$ threshold ≥ 0.8), NOISeq showed lower precision but much higher recall. It should be noted that although replicates in RNA-Seq data by Sultan et al. (2008) were combined, their data had a low level of variation (Pearson's correlation coefficient was 0.99 for human embryonic kidney sample and 0.98 for B-cell sample) between original replicates. From both simulated dataset and real dataset analysis, NOISeq with relaxed thresholds (0.8 - 0.95) appears to be a better choice compared to DESeq for differential gene-calling when no replicates are available.

Table 10: Performance of DESeq compared against the microarray study.

	***q*-value threshold**					
	0.001	0.005	0.01	0.05	0.1	0.2
Precision	1.00	0.81	0.79	0.83	0.87	0.86
Recall	0.001	0.005	0.007	0.015	0.02	0.03

Table 11: Performance of NOISeq compared against the microarray study.

	P_{NOI}-value threshold					
	0.999	0.995	0.99	0.95	0.9	0.8
Precision	0.91	0.85	0.86	0.84	0.82	0.76
Recall	0.003	0.007	0.01	0.06	0.12	0.27

3.3.2 Consistency analysis between with and without biological replications

Using the two sets of real RNA-Seq data, we tested the "consistency" in the results given by DESeq and NOISeq. We compared the results from DESeq and NOISeq using no replicate with those using replicates on the same datasets. Two datasets are: the *Chlamydomonas* datasets that have *moderate* variation and the pea aphid data that have *large* variation. See section 2.4.2 for more information on these datasets. The objective here is to see if DESeq and NOISeq can yield somewhat consistent/reliable results when no replicates are available. As described in section 2.4.2, we calculated the average precision and average recall. Note that in this section, these statistics are used to measure the *consistency* in differential gene-calling when no replicates are available compared against when replicates are available. They are used by no means to indicate any level of accuracy.

Table 12: Performance consistency with DESeq on the Chlamydomonas data.

	***q*-value threshold**					
	0.001	**0.005**	**0.01**	**0.05**	**0.1**	**0.2**
Average Precision	1.00	0.99	0.99	0.99	0.99	0.99
Average Recall	0.03	0.04	0.05	0.07	0.07	0.08

Table 13: Performance consistency with NOISeq on the Chlamydomonas data.

	P_{NOI} threshold					
	0.999	**0.995**	**0.99**	**0.95**	**0.9**	**0.8**
Average Precision	0.02	0.08	0.12	0.15	0.17	0.24
Average Recall	1.00	1.00	1.00	0.96	0.95	0.94

Table 14: Performance consistency with DESeq on the pea aphid data.

	q-value threshold					
	0.001	**0.005**	**0.01**	**0.05**	**0.1**	**0.2**
Average Precision	0.52	0.52	0.53	0.52	0.51	0.48
Average Recall	0.27	0.29	0.29	0.32	0.32	0.33

Table 15: Performance consistency with NOISeq on the pea aphid data.

	P_{NOI} threshold					
	0.999	**0.995**	**0.99**	**0.95**	**0.9**	**0.8**
Average Precision	0.00	0.02	0.03	0.17	0.22	0.25
Average Recall	0.00	0.33	0.33	0.38	0.39	0.43

As shown in Table 12, DESeq was found to be very conservative in finding differentially expressed genes when no replicates were available. This was indicated by very high precision and very low recall. In other words, with no replicates, while DESeq found only a small number of genes as differentially expressed, many of these identified genes were what it would have found if there were replicates. As shown in Table 13, NOISeq was found to be more aggressive in finding differentially expressed genes when no replicates were available, indicated by relatively low precision and high recall. It indicated that, without replicates, NOISeq could find almost all genes that would have been found if there were replicates. However, it also found many genes that would not have been found if replicates were available (possible false positives). When the data included much larger variation (CV=0.67), as shown with the results with the pea aphid datasets in Tables 14 and 15, as expected, results obtained from single replicates were not consistent with those obtained when replicates were available. Interestingly, precisions of NOISeq do not seem to be affected by the level of variations. Thus, regardless of the amount of variation, with NOISeq we expect to find the same proportion of false positives (inconsistently identified genes). However, the recall values were severely affected with the larger variation and it dropped to the level almost the same as found with DESeq.

These results were consistent with what we found using simulated data in section 3.2.4. When no replicate is available for both moderate- and large-variation datasets, DESeq is very conservative in finding differentially expressed genes whereas NOISeq is more aggressive but more error prone.

3.4 Suggested Guidelines of Using DESeq and NOISeq

This study clearly showed that biological variation affects significantly and differently how DESeq and NOISeq perform in differential gene-calling. Large variation will cause more false positives for both DESeq and NOISeq. We also showed that it is possible to improve the accuracy by combining the results of both methods. Based on the results we obtained in this study, following are our suggested strategies of using DESeq and NOISeq depending on the level of biological variation:

(1) If the biological variation is moderate, *e.g.*, $\mathrm{CV} \approx 0.33$, to control the FDR around 0.05, we can take advantage of combining results by taking the intersection of both methods using q=0.1 threshold for DESeq and P_{NOI}= 0.5 threshold for NOISeq.

(2) If the biological variation is large, *e.g.*, CV ≈ 0.67, it may be useful to consider a relaxed FDR control, *e.g.*, around 0.2, in order to find a good number of differentially expressed gene candidates. We can use the combined results using q = 0.4 threshold for DESeq and P_{NOI}= 0.5 threshold for NOISeq. Note that only a very small number of differentially expressed genes can be found in order to control the FDR smaller than 0.2.

It is highly recommended to have replications in RNA-Seq experiments. As our results showed, when there is no replicate, DESeq finds a very small number of differentially expressed genes. On the contrary, NOISeq finds more candidates of differentially expressed genes, which, however, include a large number of false positives. Based on what we observed with the results on real datasets, DESeq is very conservative where fewer genes are identified but its results are more consistent with the results obtained using replicates. NOISeq is more aggressive where more genes are identified but its results are less consistent with the results obtained using replicates. When the results of DESeq and NOISeq are compared to the results of Sultan et al. (2008)'s microarray analysis, NOISeq performed better. Based on our no-replicate experiments, the recommended strategy for analyzing no-replicate datasets is to use NOISeq with P_{NOI} thresholds 0.8-0.95.

4 Conclusions and Future Work

In this study, we presented a comparison between a parametric method DESeq and a non-parametric method NOISeq for differential gene-calling based on RNA-Seq data. Both DESeq and NOISeq performed much better on data with moderate biological variation than with large biological variation. They both found slightly more truly over-expressed genes than under-expressed genes. DESeq showed length-based bias where longer transcripts were called more as differentially expressed, whereas NOISeq did not show such trend.

Our results showed the importance of understanding the variation in the data. We also showed that combination strategy can be used to obtain improved differential gene-calling. If the biological variation is moderate and if we want to control FDR around 0.05, we can use the differentially expressed genes claimed by both programs with 0.1 *q*-value threshold for DESeq and 0.5 P_{NOI} threshold for NOISeq. If the biological variation is large, we need to consider a higher FDR control, *e.g.*, 0.2 or even higher, in order to find a good number of differentially expressed genes. We can use the combined results with 0.4 *q*-value threshold for DESeq and 0.5 P_{NOI} threshold for NOISeq. When no replicates are available, DESeq is conservative in finding differentially expressed genes, whereas NOISeq is aggressive. Based on the simulated data and real data analysis, we recommend using NOISeq with probability thresholds 0.8 ~ 0.95.

In order to confirm our analysis of DESeq and NOISeq, for example, qRT-PCR confirmation results for some of the identified genes would be useful. We would like to collaborate on this regard with our collaborators for further analysis of their RNA-Seq data.

More different methods with more data should be tested to see if they have consistent performance. Especially interesting ones are those very recently developed RNA-Seq differential expression calling methods, *e.g.,* new versions of Cufflinks/Cuffdiff (Trapnell, et al., 2012) and BitSeq (Glaus, et al., 2012). Combination methods should be also further explored for the most reliable results.

It would also be interesting to consider differential expression in terms of gene pathways and try to find differentially expressed gene pathways in various conditions. We can test the performance of currently available methods of analyzing gene pathways using RNA-Seq data or build our own method to analyze and gain insights on the expression patterns on the level of gene pathways.

References

Alkan, C., Kidd, J.M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J.O., Baker, C., Malig, M., Mutlu, O., Sahinalp, S.C., Gibbs, R.A. and Eichler, E.E. (2009) Personalized copy number and segmental duplication maps using next-generation sequencing, *Nature genetics*, **41**, 1061-1067.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *J Mol Biol*, **215**, 403-410.

Anders, S. and Huber, W. (2010) Differential expression analysis for sequence count data, *Genome Biol*, **11**, R106.

Anders, S. and Huber, W. (2010) Differential expression analysis for sequence count data, *Genome biology*, **11**, R106.

Auer, P.L. and Doerge, R.W. (2011) A Two-Stage Poisson Model for Testing RNA-Seq Data, *Statistical Applications in Genetics and Molecular Biology*, **10**.

Barry, W.T., Nobel, A.B. and Wright, F.A. (2005) Significance analysis of functional categories in gene expression studies: a structured permutation approach, *Bioinformatics*, **21**, 1943-1949.

Benjamini, Y. and Speed, T.P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing, *Nucleic acids research*, **40**, e72.

Bullard, J.H., Purdom, E., Hansen, K.D. and Dudoit, S. (2010) Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments, *BMC Bioinformatics*, **11**, 94.

Burrows, M. and Wheeler, D.J. (1994) A block-sorting lossless data compression algorithm, *Technical report 124. Palo Alto, CA: Digital Equipment Corporation.*

Garber, M., Grabherr, M.G., Guttman, M. and Trapnell, C. (2011) Computational methods for transcriptome annotation and quantification using RNA-seq, *Nat Methods*, **8**, 469-477.

Glaus, P., Honkela, A. and Rattray, M. (2012) Identifying differentially expressed transcripts from RNA-seq data with biological variation, *Bioinformatics*, **28**, 1721-1728.

Hach, F., Hormozdiari, F., Alkan, C., Birol, I., Eichler, E.E. and Sahinalp, S.C. (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping, *Nature methods*, **7**, 576-577.

Hansen, K.D., Brenner, S.E. and Dudoit, S. (2010) Biases in Illumina transcriptome sequencing caused by random hexamer priming, *Nucleic Acids Res*, **38**, e131.

Hardcastle, T.J. and Kelly, K.A. (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data, *BMC Bioinformatics*, **11**, 422.

Kall, L., Storey, J.D., MacCoss, M.J. and Noble, W.S. (2008) Posterior error probabilities and false discovery rates: two sides of the same coin, *J Proteome Res*, **7**, 40-44.

Kvam, V.M., Liu, P. and Si, Y. (2012) A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data, *Am J Bot*, **99**, 248-256.

Langmead, B., Trapnell, C., Pop, M. and Salzberg, S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome Biol*, **10**, R25.

Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinformatics*, **25**, 1754-1760.

Li, J., Jiang, H. and Wong, W.H. (2010) Modeling non-uniformity in short-read rates in RNA-Seq data, *Genome Biol*, **11**, R50.

Li, R., Yu, C., Li, Y., Lam, T.W., Yiu, S.M., Kristiansen, K. and Wang, J. (2009) SOAP2: an improved ultrafast tool for short read alignment, *Bioinformatics*, **25**, 1966-1967.

Mardis, E.R. (2008) The impact of next-generation sequencing technology on genetics, *Trends Genet*, **24**, 133-141.

Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L. and Wold, B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq, *Nat Methods*, **5**, 621-628.

Oshlack, A. and Wakefield, M.J. (2009) Transcript length bias in RNA-seq data confounds systems biology, *Biol Direct*, **4**, 14.

Roberts, A., Trapnell, C., Donaghey, J., Rinn, J.L. and Pachter, L. (2011) Improving RNA-Seq expression estimates by correcting for fragment bias, *Genome biology*, **12**, R22.

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**, 139-140.

Robinson, M.D. and Oshlack, A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biol*, **11**, R25.

Robinson, M.D. and Smyth, G.K. (2007) Moderated statistical tests for assessing differences in tag abundance, *Bioinformatics*, **23**, 2881-2887.

Robinson, M.D. and Smyth, G.K. (2008) Small-sample estimation of negative binomial dispersion, with applications to SAGE data, *Biostatistics*, **9**, 321-332.

Ruffalo, M., LaFramboise, T. and Koyuturk, M. (2011) Comparative analysis of algorithms for next-generation sequencing read alignment, *Bioinformatics*, **27**, 2790-2796.

Rumble, S.M., Lacroute, P., Dalca, A.V., Fiume, M., Sidow, A. and Brudno, M. (2009) SHRiMP: accurate mapping of short color-space reads, *PLoS computational biology*, **5**, e1000386.

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, *Journal of molecular biology*, **147**, 195-197.

Smyth, G.K., Michaud, J. and Scott, H.S. (2005) Use of within-array replicate spots for assessing differential expression in microarray experiments, *Bioinformatics*, **21**, 2067-2075.

Smyth, G.K. and Verbyla, A.P. (1996) A conditional likelihood approach to residual maximum likelihood estimation in generalized linear models, *Journal of the Royal Statistical Society Series B-Methodological*, **58**, 565-572.

Sultan, M., Schulz, M.H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H. and Yaspo, M.L. (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome, *Science*, **321**, 956-960.

Tarazona, S., Garcia-Alcalde, F., Dopazo, J., Ferrer, A. and Conesa, A. (2011) Differential expression in RNA-seq: a matter of depth, *Genome Res*, **21**, 2213-2223.

Trapnell, C., Pachter, L. and Salzberg, S.L. (2009) TopHat: discovering splice junctions with RNA-Seq, *Bioinformatics*, **25**, 1105-1111.

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D.R., Pimentel, H., Salzberg, S.L., Rinn, J.L. and Pachter, L. (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks, *Nat Protoc*, **7**, 562-578.

Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J. and Pachter, L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation, *Nat Biotechnol*, **28**, 511-515.

Wang, Z., Gerstein, M. and Snyder, M. (2009) RNA-Seq: a revolutionary tool for transcriptomics, *Nat Rev Genet*, **10**, 57-63.

Wu, Z., Jenkins, B.D., Rynearson, T.A., Dyhrman, S.T., Saito, M.A., Mercier, M. and Whitney, L.P. (2010) Empirical bayes analysis of sequencing-based transcriptional profiling without replicates, *BMC Bioinformatics*, **11**, 564.

Supplemental Materials

Table of Contents

List of Figures

List of Tables

S1 The R Script Used for the Simulation

We wrote the R script to generate the appropriate amount (according to the expression assigned) of starting positions of each transcript for producing short reads as shown in Figure S1. It takes the length of each transcript of a sequence file as input (input file has only the length for each transcript) and produces 4 sets of starting positions for each transcript for the entire sequence file. We then generated subsequences of length 36 from each transcript as short reads based on these starting positions. Biological and technical variations are both incorporated in the script. The simulation is described in detail in material and methods in the main text.

```
#Read in the length of each gene
m<-read.table("Ref_length.txt");
exp1<-c();
exp2<-c();
ctr1<-c();
ctr2<-c();
#Total number of genes
total<-26017;
# shape and scale parameter of gamma distribution to assign expected expression for each
gene
shape_var<-0.15;
scale_var<-1160;
#biological variation parameter
bio_var<-9;
#max fold change allowed in experimental condition
max_fold<-5;
#randomize the seed
set.seed(as.integer(as.double(Sys.time())));
#first 10% is differentially under-expressed genes
for(i in 1:2602)
{
   #fold chagne expected in experimental condition
   fold<-1/(sample(11:(max_fold*10),1)/10);
   #transcript expression level
   level<-rgamma(1,shape_var,scale=scale_var);
   #biological variation (constant coefficient of variation)
   num1<-rgamma(1,bio_var,scale=level*fold/bio_var);
   num2<-rgamma(1,bio_var,scale=level*fold/bio_var);
   num3<-rgamma(1,bio_var,scale=level/bio_var);
   num4<-rgamma(1,bio_var,scale=level/bio_var);
   #technique variation assuming Poisson
   exp1<-append(exp1,rpois(1,num1));
   exp2<-append(exp2,rpois(1,num2));
   ctr1<-append(ctr1,rpois(1,num3));
   ctr2<-append(ctr2,rpois(1,num4));
}
```

```
#this second 10% is differentially over expressed genes
for(i in 2603:5204)
{
   fold<-sample(11:(max_fold*10),1)/10;
   level<-rgamma(1,shape_var,scale=scale_var);
   num1<-rgamma(1,bio_var,scale=level*fold/bio_var);
   num2<-rgamma(1,bio_var,scale=level*fold/bio_var);
   num3<-rgamma(1,bio_var,scale=level/bio_var);
   num4<-rgamma(1,bio_var,scale=level/bio_var);
   exp1<-append(exp1,rpois(1,num1));
   exp2<-append(exp2,rpois(1,num2));
   ctr1<-append(ctr1,rpois(1,num3));
   ctr2<-append(ctr2,rpois(1,num4));
}
#rest 80% is non-differentially expressed genes
for(i in 5205:total)
{
   level<-rgamma(1,shape_var,scale=scale_var);
   num1<-rgamma(1,bio_var,scale=level/bio_var);
   num2<-rgamma(1,bio_var,scale=level/bio_var);
   num3<-rgamma(1,bio_var,scale=level/bio_var);
   num4<-rgamma(1,bio_var,scale=level/bio_var);
   exp1<-append(exp1,rpois(1,num1));
   exp2<-append(exp2,rpois(1,num2));
   ctr1<-append(ctr1,rpois(1,num3));
   ctr2<-append(ctr2,rpois(1,num4));
}

#Now generating short reads from mouse transcripts
#assumed short reads length
read_length<-36;
#initialize the length of gene that are mappable to short reads
effe_length=0;
sink("exp1.txt");
for(i in 1:total)
{
   if(m[i,1]<36)
   {
      effe_length=0;
   }
   else
   {
      effe_length=m[i,1]-read_length;
   }
   cat(sample(0:effe_length,as.integer(effe_length/300*exp1[i]),replace=TRUE));
   cat("\n");
}
sink();
sink("exp2.txt");
for(i in 1:total)
{
   if(m[i,1]<36)
   {
      effe_length=0;
   }
   else
   {
      effe_length=m[i,1]-read_length;
   }
   cat(sample(0:effe_length,as.integer(effe_length/300*exp2[i]),replace=TRUE));
   cat("\n");
}
sink();
sink("ctr1.txt");
for(i in 1:total)
{
   if(m[i,1]<36)
   {
      effe_length=0;
   }
   else
   {
      effe_length=m[i,1]-read_length;
   }
   cat(sample(0:effe_length,as.integer(effe_length/300*ctr1[i]),replace=TRUE));
   cat("\n");
}
sink();
sink("ctr2.txt");
for(i in 1:total)
{
   if(m[i,1]<36)
   {
      effe_length=0;
   }
   else
   {
      effe_length=m[i,1]-read_length;
   }
   cat(sample(0:effe_length,as.integer(effe_length/300*ctr2[i]),replace=TRUE));
   cat("\n");
}
sink();
#When testing on no-replicate experiments, just use one short reads file
#from exp and one from ctr condition.
```

Figure S1: R script for generating starting positions for short reads

S2 Analysis of Combining DESeq and NOISeq Results with Various Thresholds

As we described in section 3.2.5, we examined many possible combinations of thresholds to see if we can improve differential gene-calling performance by combining the results from DESeq and NOISeq (taking the intersection results). Supplemental Tables S1 and S2 showed the sensitivities, FDRs and sensitivity to FDR ratios from the combined results with all possible pairs of thresholds: DESeq q-value thresholds ($1*10^{-20}$, $1*10^{-10}$, 0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4 and 0.5) and NOISeq P_{NOI} thresholds ($1-1*10^{-20}$, $1-1*10^{-10}$, 0.99999, 0.99995, 0.9999, 0.9995, 0.999, 0.995, 0.99, 0.95, 0.9, 0.8, 0.7, 0.6 and 0.5).

Table S1: Analysis of combination strategy with moderate variation data.[a]

			NOISeq P_{NOI} threshold														
			1-1*10^{-20}	1-1*10^{-10}	0.99999	0.99995	0.9999	0.9995	0.999	0.995	0.99	0.95	0.9	0.8	0.7	0.6	0.5
		Single[b]	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0, -	0.03, 0, -	0.11, 0.03, **3.67**	0.18, 0.13, 1.38	0.25, 0.24, 1.04	0.32, 0.38, 0.84
DESeq q-value threshold	1*10^{-20}	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	1*10^{-10}	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.00001	0.01, 0.05, 0.2	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.006, 0.029, 0.21	0.009, 0.021, 0.43	0.009, 0.02, 0.45	0.011, 0.018, 0.61	0.011, 0.017, 0.65
	0.00005	0.02, 0.03 , 0.67	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.011, 0.017, 0.65	0.017, 0.011, 1.55	0.019, 0.01, 1.9	0.022, 0.009, 2.44	0.022, 0.008, 2.75
	0.0001	0.03, 0.02, 1.5	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.015, 0.012, 1.25	0.025, 0.008, 3.12	0.029, 0.007, 4.14	0.032, 0.006, 5.33	0.033, 0.006, 5.5
	0.0005	0.06, 0.02, 3	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.023, 0.008, 2.88	0.042, 0.005, 8.4	0.051, 0.004, 12.75	0.057, 0.003, 19	0.059, 0.003, 19.67
	0.001	0.08, 0.02, 4	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.026, 0.007, 3.71	0.052, 0.004, 13	0.065, 0.003, 21.67	0.075, 0.003, 25	0.079, 0.005, 15.8
	0.005	0.14, 0.03, 4.67	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 4	0.079, 0.002, 39.5	0.103, 0.002, 51.5	0.12, 0.002, 60	0.129, 0.006, 21.5

	0.01	0.17, 0.03, **5.67**	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 4	0.09, 0.004, 22.5	0.125, 0.003, 41.67	0.147, 0.003, 49	0.158, 0.006, 26.33
	0.05	0.24, 0.08, 3	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 4	0.105, 0.016, 6.56	0.156, 0.023, 6.78	0.192, 0.024, 8	0.213, 0.027, 7.89
	0.1	0.28, 0.13, 2.15	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 4	0.107, 0.03, 3.57	0.17, 0.04, 4.25	0.216, 0.042, 5.14	0.247, 0.045, 5.49
	0.2	0.34, 0.20, 1.7	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 5	0.108, 0.033, 3.27	0.18, 0.083, 2.17	0.236, 0.093, 2.54	0.277, 0.098, 2.83
	0.3	0.37, 0.28, 1.32	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 6	0.108, 0.033, 3.27	0.182, 0.113, 1.61	0.246, 0.146, 1.68	0.294, 0.163, 1.8
	0.4	0.41, 0.37, 1.11	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 7	0.108, 0.033, 3.27	0.182, 0.135, 1.35	0.251, 0.195, 1.29	0.308, 0.234, 1.32
	0.5	0.44, 0.43, 1.02	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.125, 0.01	0.028, 0.007, 8	0.108, 0.033, 3.27	0.182, 0.135, 1.35	0.253, 0.225, 1.12	0.315, 0.283, 1.11

[a] *The result is formatted as sensitivity, FDR, sensitivity to FDR ratio. The ratio is shown in bold and blue fonts when it is larger than the highest ratio obtained by either DESeq or NOISeq, which is 5.67 (DESeq at q=0.01). The highest ratio is shown in red and bold fonts.*

[b] *The column and the row shows the statistics from using the single method. The highest ratio in each method is shown in bold fonts.*

Table S2: Analysis of combination strategy with large variation data.[a]

			NOISeq P_{NOI} threshold														
			$1-1*10^{-20}$	$1-1*10^{-10}$	0.99999	0.99995	0.9999	0.9995	0.999	0.995	0.99	0.95	0.9	0.8	0.7	0.6	0.5
		Single[b]	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0, -	0.007, 0, -	0.04, 0.17, 0.24	0.08, 0.33, 0.24	0.14, 0.46, 0.30	0.21, 0.56, **0.38**
DESeq q-value threshold	$1*10^{-20}$	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	$1*10^{-10}$	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.00001	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.00005	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.0001	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.0005	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.001	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.005	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -
	0.01	0.001, 0.36, 0.003	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.001, 0.2, 0	0.001, 0.2, 0	0.001, 0.2, 0	0.001, 0.2, 0	0.001, 0.2, 0	0.001, 0.167, 0.01

	0.05	0.005, 0.28, 0.02	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.111, 0.02	0.002, 0.091, 0.02	0.002, 0.083, 0.02	0.002, 0.071, 0.03	0.003, 0.062, 0.05	0.003, 0.053, 0.06
	0.1	0.01, 0.27, 0.04	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.182, 0.01	0.004, 0.125, 0.03	0.005, 0.125, 0.04	0.007, 0.103, 0.07	0.007, 0.098, 0.07	0.009, 0.113, 0.08
	0.2	0.03, 0.27, 0.11	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.182, 0.01	0.007, 0.128, 0.05	0.016, 0.098, 0.16	0.02, 0.118, 0.17	0.023, 0.131, 0.18	0.028, 0.131, 0.21
	0.3	0.07, 0.29, 0.24	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.182, 0.01	0.007, 0.125, 0.06	0.027, 0.131, 0.21	0.04, 0.156, 0.26	0.047, 0.169, 0.28	0.057, 0.174, 0.33
	0.4	0.11, 0.35, 0.31	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.182, 0.01	0.007, 0.125, 0.06	0.037, 0.159, 0.23	0.059, 0.203, 0.29	0.073, 0.224, 0.33	0.089, 0.238, 0.37
	0.5	0.15, 0.42, **0.36**	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0, 0, -	0.002, 0.182, 0.01	0.007, 0.125, 0.06	0.039, 0.178, 0.22	0.071, 0.285, 0.25	0.094, 0.309, 0.3	0.115, 0.321, 0.36

[a] *The result is formatted as sensitivity, FDR, sensitivity to FDR ratio. The ratio is shown in bold and blue fonts when it is larger than the highest ratio obtained by DESeq (0.36 at q=0.5). However, this is not higher than the highest ratio obtained by NOISeq, which is 0.38 (NOISeq at P_{NOI}=0.5).*

[b] *The column and the row shows the statistics from using the single method. The highest ratio in each method is shown in bold fonts.*

S3 Development of the Consolidated Mouse Genome Annotation

Several versions of mouse genome annotations are available from different databases. Three prominent databases (NCBI, Ensembl, and UCSC) are listed in Table S3. In order to obtain a single and most inclusive, as complete as possible, annotated mouse genome, we decided to develop our own consolidated set of mouse protein coding sequences (CDS).

Table S3: Three currently available mouse genome annotations.[a]

Database	Website	Annotation Version	Number of transcripts
NCBI	http://www.ncbi.nlm.nih.gov/	MGSCv37	27270
Ensembl	http://uswest.ensembl.org/index.html	M37.61	54948
UCSC	http://genome.ucsc.edu/	Downloaded the "knownGene" table of "NCBI37/mm9" build on Feb 7, 2011	39481

[a] *Data are available in the file "gff3.zip" found at: http://bioinfolab.unl.edu/emlab/gpcr_mouse/ It contains "NCBI.gff3", "Ensembl.gff3", and "UCSC_info.txt", each of which corresponds to the annotation file downloaded from NCBI, Ensembl, and UCSC, respectively.*

Our consolidated set of mouse protein coding sequences (CDS) was generated through following steps:

Step 1: Based on the three annotations, we compared the location of CDS from each transcript (all coding regions on exons). If a CDS had the exact same location and exon-intron structure annotated by any two or all databases but named differently, we retained the corresponding regions of the sequence and simply concatenated their entry names to make the name of the sequence in our consolidated set. For example, if a sequence annotated in NCBI with name “NM_1234” is on chromosome 1 with the coding region from position 100 nucleotide (nt) to position 500 nt and a sequence annotated in Ensembl with name “ENSMUST5678” is also on chromosome 1 with its coding region from position 100 nt to position 500 nt and the exon-intron structure are exactly the same, two annotations obviously referred to the same CDS. In this case, we retained this CDS region of the genome and name it as “NM_1234|ENSMUST5678” in our consolidated set.

Step 2: If we cannot find an exact location match, we kept all versions of the sequence as they were annotated in each database. For example, if the previous example had “NM_1234” with the coding region 100-500 nt but “ENSMUST5678” with its coding region 100-550 nt, we kept both versions of the CDS. In this case, “NM_1234” and “ENSMUST5678” became two different entries with their corresponding CDSs in our consolidated set.

Step 3: For CDSs that were on not fully resolved scaffolds (no clear chromosomal assignment in the genome), we extracted the CDS from the scaffold and used BLAST (blastn) (Altschul, et al., 1990) to perform similarity search against the each resolved chromosome of the mouse genome. If a perfect match (100% identity and coverage) was found, we assumed this CDS was on the location of the matched chromosome, and then treated it as a CDS with the apparent position on the chromosome and apply steps 1 and 2 on it.

Step 4: We used those CDSs on unresolved scaffolds that did not have perfect matches after step 3 to perform BLAST (blastn) similarity search against all unresolved scaffolds. If a perfect match was found, we retained the CDS and concatenated their sequence names. For example, if a sequence named "NM_1111" was on "scaffold_1" and a sequence named "ENMUST2222" was on "scaffold_5", and their sequences were identical, we retained the CDS and named it as "NM_1111|ENMUST2222" in our consolidated set.

Step 5: We retained the CDS regions of the rest of sequences as they are annotated in 3 databases.

In essence, we extracted the CDS regions of all mouse transcripts from 3 sets of annotations, removed all redundant copies of CDS if it was annotated on the same chromosomal location in more than two databases, and kept all versions of CDS if it was annotated on different chromosomal locations in different databases.

We call our consolidated mouse CDS set "Con_Mouse". The database (in a FASTA format file) is available from: http://bioinfolab.unl.edu/emlab/gpcr_mouse/Ref_transcripts.fa

The example entries of the Con_Mouse database are shown in Figure S2.

```
>NM_025928.2|ENSMUST00000056370|uc008pva.1
ATGGCTGAGGTGAGCCGAGATAGCGAGGCTGCGGAAAGGGGGCCTGAGGGCTCCTCTCCGGAAGCTGTGCCA
GGGGACGCGACCATCCCCAGGGTGAAACTCCTGGACGCCATAGTAGACACTTTCCTCCAGAAGCTAGTCGCC
GACAGGAGCTACGAGAGGTTCACCACCTGCTACAAACACTTCCACCAGTTGAACCCTGAGGTGACGCAGAGG
ATCTATGACAAGTTTGTGGCTCAGTTGCAGACATCCATCCGCGAGGAAATCTCAGAAATCAAAGAGGAGGGG
AACCTAGAAGCTGTCCTGAACTCCCTGGATAAGATCATAGAAGAAGGCAGAGAGCGCGGAGAGCCAGCCTGG
CGACCCAGTGGAATCCCAGAGAAAGACCTGTGTAGTGTCATGGCACCCTACTTCCTGAAGCAACAGGATACC
CTGTGTCATCAAGTACGGAAACAGGAAGCCAAGAACCAGGAACTGGCCGACGCTGTCCTGGCCGGGCGCAGG
CAGGTGGAGGAGCTGCAGCAGCAGGTTCGGGCCCTCCAGCAAACATGGCAGGCTCTACACAGAGAGCAGAGG
GAGCTGCTGTCAGTGCTGAGGGCGCCTGAGTGA
>ENSMUST00000150158
ATGGAGGAACTGATACTGCAGGATGAGACCCTCCTGGAGACCATGCAGAGCTACATGGACGCCTCCCTTATA
TCCCTCATTGAGGATTTTGGAGAGAGCAGATTATCTCTGGAGGACCAGAATGAAATGTCGCTGCTCACAGCT
CTGACGGAGATCTTGGACAATGCAGATTCTGAGAACCTGTCCCCTTTTGACACCATTCCTGATTCAGAGCTG
CTCGTGTCCCCTCGGGAGAGCTCCTCTGTTGAGGTGCCTCTTGCAGACTCTCCATGGGACTTCTCTCCGCCT
CCTTTCTTGGAAACTTCTTCCCCTAAGCTGCCTAGCTGGAGACCCTCGAGACCAAGACCTCGATGGGGTCAG
TCCCCTCCTCCTCAGCAGCGCAGTGATGGGGAAGAGGAGGAGGAGGTCGCCGGTTTCAGTGGTCAGATGCTT
GCTGGC
>NM_025868.2|ENSMUST00000053664|ENSMUST00000111665|uc008kix.1
ATGGCTGTCCTTGCGCCTCTGATTGCTTTGGTGTACTCGGTGCCGCGGCTTTCTCGATGGCTGGCCCGACCT
TATTGCCTTCTGTCTGCTCTGCTTTCCATTGCTTTCCTCCTCGTGAGGAAACTGCCACCGATTTGCAATGGT
CTCCCCACGCAACGCGAAGATGGCAACCCGTGTGACTTTGACTGGAGAGAAGTGGAGATCCTGATGTTCCTC
AGTGCCATTGTGATGATGAAGAACCGCAGATCCATCACTGTGGAGCAACATGTAGGCAACATCTTTATGTTT
AGTAAAGTGGCCAACGCCATCCTTTTCTTCCGACTGGATATTCGAATGGGTCTGCTATACCTCACACTCTGC
ATAGTGTTCCTGATGACCTGCAAGCCCCCGCTGTACATGGGTCCTGAGTATATCAAGTACTTCAATGATAAA
ACCATTGATGAGGAGCTGGAGCGAGACAAGAGGGTCACTTGGATTGTGGAGTTCTTTGCCAACTGGTCTAAT
GATTGCCAATCCTTTGCTCCCATCTATGCGGACTTGTCCCTCAAGTACAACTGTTCAGGGCTAAATTTTGGG
AAGGTAGATGTTGGACGCTACACTGACGTTAGCACACGGTACAAAGTGAGCACATCACCCCTCACCAGACAG
CTCCCTACCCTGATTCTGTTCCAAGGCGGCAAGGAGGTCATTCGTCGGCCGCAGATTGACAAGAAAGGACGA
GCTGTCTCTTGGACCTTTTCTGAGGAGAATGTGATTCGAGAATTCAACTTGAATGAGCTATACCAACGAGCC
AAGAAGCACTCAAAGGGTGGAGACATGTCAGAAGAGAAGCCTGTGGACCCTGCTCCCACTACTGTGCCAGAT
GGGGAAAACAAGAAGGACAAATAG
```

Figure S2: Examples of our Con_Mouse database entries.

A summary statistics of our Con_Mouse database is presented in Table S4. It shows the number of sequences from each database or shared by different databases.

Table S4: Summary of our Con_Mouse database compared to other databases.[a]

Total number of CDSs in Con_Mouse	Shared in 3 databases	Shared in NCBI and Ensembl	Shared in NCBI and UCSC	Shared in Ensembl and UCSC	Unique in NCBI	Unique in Ensembl	Unique in UCSC
67981	18587	1195	917	5363	6081	24495	11343

[a] *The total number of genes is 26,017. In the simulation study, short reads are generated from only one transcript or CDS for one gene excluding alternative splicing forms.*

S4 Compilation of G-Protein Coupled Receptors from the Mouse Genome

In this section, we compiled the entire set of G-protein coupled receptor proteins from the mouse genome using our consolidated database Con_Mouse.

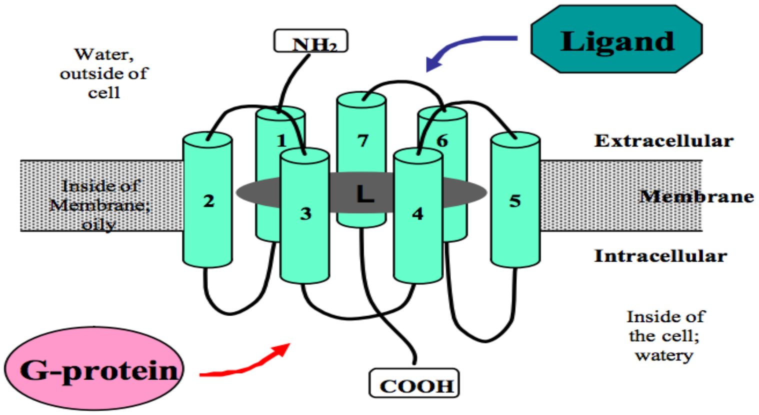

Figure S3: Seven-transmembrane receptor model.

(adapted from E. N. Moriyama, unpublished)

G-protein coupled receptors (GPCRs) are also known as seven-transmembrane receptors (7TMRs) (Figure S3). Upon binding of a ligand, the GPCR activates the G-protein (Guanine-nucleotide binding protein), which in turn activates downstream signaling pathways depending on the type of G-protein. GPCRs are important proteins that are involved in many cellular signaling processes and they are common targets of therapeutic drugs. Members of GPCRs act as receptors for many signaling molecules such as hormones, nucleotides, opiates, neurotransmitters and odorants, and GPCR families are also very divergent (Kim, et al., 2000). Due to their low sequence similarity, we took several steps to compile our list of GPCRs from our consolidated mouse CDS dataset, including BLAST (Altschul, et al., 1990), HMMER search with Pfam (Eddy, 1998; Punta, et al., 2012), and 7TMRmine (Lu, et al., 2009). All files mentioned hereafter can be found at http://bioinfolab.unl.edu/emlab/gpcr_mouse/.

BLAST approach:

First, we used mouse GPCR protein sequences that were downloaded from GPCRDB (Vroling, et al., 2011) in the file "GPCR_mouse.fa" (3296 GPCR sequences) to perform similarity search using a BLAST program tblastn against our consolidated mouse coding sequence (CDS) set. If the search result was better than 99% identity and 99% alignment coverage for both query and subject sequences, we marked this subject sequence as "G1" in the result file, "GPCR_table_RNA-seq.xlsx". This was a very strict threshold and we considered these resultant sequences (2149 sequences obtained) highly likely to be GPCRs.

Second, we went through each GPCR classification leaf group alignment from GPCRDB ("gpcr_families.txt" and alignments downloaded from http://www.gpcr.org/7tm/data/), and found "representative" sequences (209 sequences obtained). "Representative" sequence was defined as a GPCR sequence from rat ("rat_query.fa"), human ("human_query.fa"), or other vertebrate ("verte_query.fa") that was present in the leaf classification group where mouse GPCR was missing in this group. Then, we used these "representative" sequences to perform tblastn search against our Con_Mouse database. We used 90% identity and 95% coverage for both query and subject sequences as the threshold if the "representative" sequence was from rat, 70%

identity and 95% coverage as the threshold if it was from human, and 70% identity and 95% coverage as the threshold if it was from one of other vertebrates. The rationale here was that in one GPCR classification leaf group, if there were many GPCRs of other vertebrates but no mouse GPCRs, then it was possible that mouse GPCR was missing in GPCRDB in this leaf GPCR group. We marked sequences that passed the according thresholds as “G2” in the result file (30 sequences obtained).

Third, we used subject sequences from step 2 that did not pass the threshold but still had promising scores (in the file “possible_GPCR_query.fa”) to perform blastx search against the NCBI non-redundant protein database. If they returned GPCR sequences from some species as first hits, we considered them as GPCRs. The rationale here was that these sequences were more closely related to some GPCRs rather than the “representative” sequences we chose earlier. Only 3 sequences were added in this step, “XM_003086424.1”, “XM_003085563.1” and “XM_003085916.1”, and they were marked as “G3”.

Fourth, we used sequences found in previous steps (G1, G2, and G3) to perform blastn search against our Con_Mouse database again and added those sequences that showed better score than GPCRs determined in G1, G2 and G3; *i.e.,* if a sequence that had not been found as GPCR before but was found in this step to be more similar to a query from G1, G2, or G3 than another sequence determined to be in the GPCR family X, then this sequence is considered to be a GPCR candidate from the family the query is thought to belong to. The rationale here is similar to PSI-BLAST that using results from previous search to increase the sensitivity to search for more similar sequences. 308

sequences were newly identified as GPCRs in this step. We marked these sequences as "G4".

Fifth, we marked sequences as "G5" (18 sequences) if they were the best hits in the BLAST search result from "G1" step but failed to pass "G1" threshold and not found in previous steps. In general, sequences in G2 – G5 categories were considered likely to be GPCRs with more relaxed thresholds comparing to G1.

All G1 to G5 sequences are marked in the result file, GPCR_table_RNA-seq.xlsx.

HMMER with Pfam approach:

In order to find the GPCR score threshold, we first used the profile hidden Markov models (HMMs) of the GPCR families identified in Pfam v26 database (the file "pfam_list_mouse_A.txt", including 40 Pfam families) to search against all sequences in Swiss-Prot database ("2011_09" release) by profile HMM search using HMMER v3.0 program. The first threshold was determined by the highest bit score of the non-GPCR sequences (sequences not in "GPCR_mouse.fa") from Swiss-Prot in the search result (T1 column of the file "pfam_thresholds.txt"; *e.g.,* Pfam family "PF12003" had T1 score threshold of 181.9 and "PF03402" had T1 score threshold of 14.8). The second threshold was determined by the lowest bit score of the GPCR sequences (sequences in the file "GPCR_mouse.fa") in the Swiss-Prot search result even though non-GPCR sequences might have higher bit scores (T2 column of "pfam_thresholds.txt"; e.g., "PF12003" had T2 bit score threshold of 13.1 and "PF03402" had T2 bit score threshold 13). We then used the profile HMMs of the GPCR families from Pfam to search against the translated

Con_Mouse database (“Ref_proteins.fa”, translation was done using the reading frame information from “Ref_seq_table_RNA-seq.xlsx”) and based on the thresholds (T1 and T2) we just determined. Sequences from our CDS with bit scores higher than the corresponding thresholds of the Pfam family were considered GPCRs. If a sequence in our mouse database passed the relatively more relaxed T2 and/or relatively stricter threshold T1, we marked the sequence with “G” in Pfam-T2 and/or Pfam-T1 column in the result file, respectively. A total of 1125 sequences passed the T1 thresholds and 2801 sequences passed the T2 thresholds.

7TMRmine approach:

7TMRmine (Lu, et al., 2009) includes multiple GPCR sequence prediction methods. Among them, we are mostly interested in the result from SAM, GPCRHMM and SVM. Sequence alignment and modeling system (SAM) (Karplus, et al., 1998) uses profile hidden Markov models (HMMs) built from sequence alignments to predict protein family memberships. A profile HMM is a full probabilistic model based on a sequence alignment. GPCRHMM is developed by Wistrand et al. (2006). It uses profile HMMs, distinct loop length patterns, and amino acid composition differences among different regions in GPCRs for prediction. We used SAM, SAM1, and SAM2 from 7TMRmine (they use different E-value thresholds: 0.05, 4.23, and 6.52, respectively). Support vector machine (SVM) makes classification based on a hyperplane separating a remapped instance space. We used SVM-AA (using amino acid composition) and SVM-di (using dipeptide frequencies) from 7TMRmine. If a method in 7TMRmine system predicted a sequence in our translated Con_Mouse (“Ref_proteins.fa”) to be a GPCR, we marked the sequence with “G” in the according method column in the result file.

Results of mouse GPCR identification:

To summarize, 2149 sequences passed BLAST strict threshold (G1 step, denoted as “blast_s” in Figure S4), and 2508 sequences passed BLAST relaxed threshold (G2-G5 steps, “blast_r”). 1125 sequences passed T1 threshold of Pfam profile HMM search (“pfam_s”), and 2801 sequences passed T2 threshold (“pfam_r”). GPCRHMM predicted 2566 sequences as GPCRs (“gpcrhmm”). 2246 sequences passed strict threshold we set for SAM (“sam”, E-value threshold 0.05) and 2580 sequences passed relaxed threshold (“sam2”, E-value threshold 6.52). SVM predicted 5870 sequences to be GPCRs (“svm_di”).

In order to choose the most confident predictions, we looked at how prediction results overlap with each other (Figure S4). Numbers in the figure are the numbers of sequences predicted to be GPCRs using various methods.

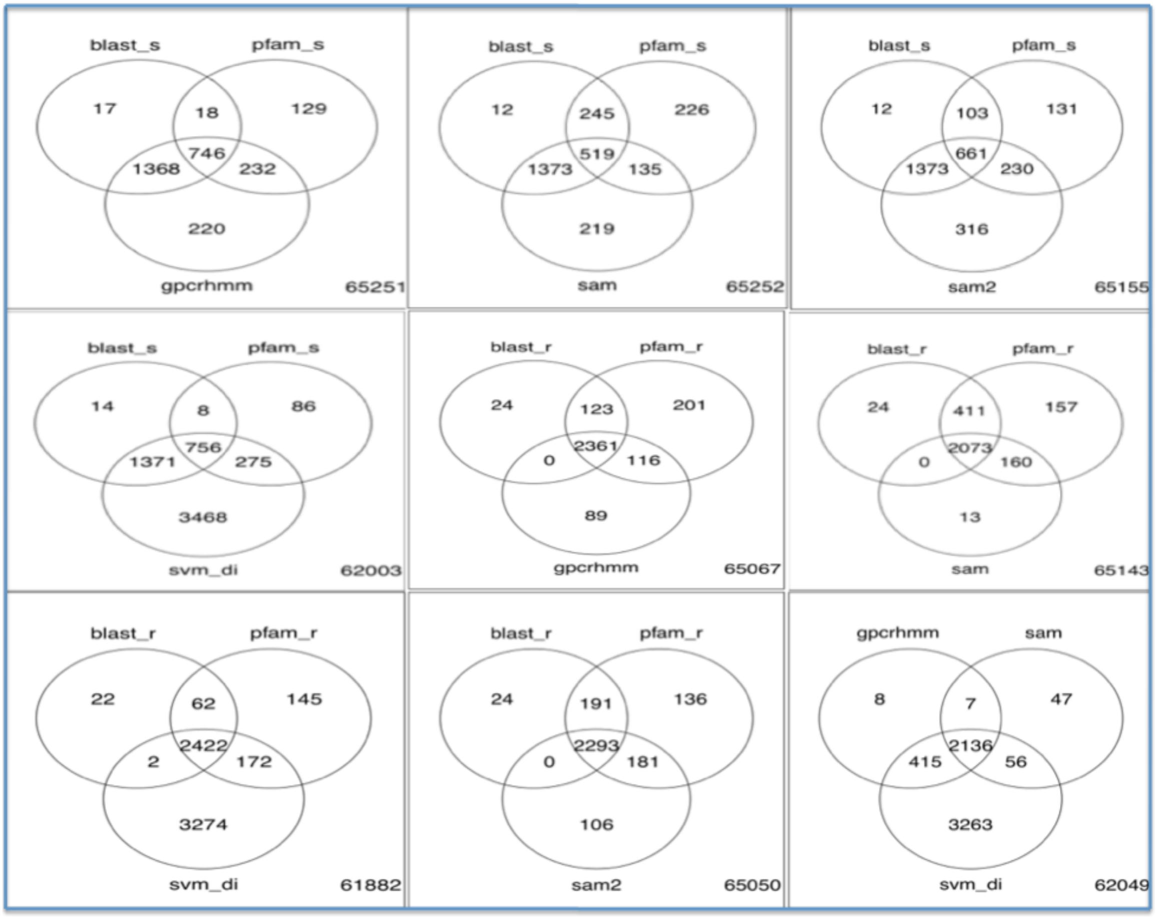

Figure S4: Number of GPCRs predicted by various methods and how the results overlap with each other
The numbers outside the circles are the numbers of sequences from Con_Mouse predicted to be not GPCR.

By comparing the resultant Venn diagrams and checking some of the actual sequences, we decided to divide our prediction file ("GPCR_table_RNA-seq.xlsx") into two parts. The first part is our most confident predictions (row 2–2759, 2758 sequences) that consists predictions from GPCRHMM, SAM, and the overlap between blast_r and pfam_r. The second part is the rest of the sequences that represent likely GPCR sequences but with less confidence (5107 sequences).

S5 Compilation of Regulators of G Protein Signaling, Nuclear Receptors, and Their Domain Distance Matrices

The objectives of this project were 1) to compile all mouse Regulators of G-protein signaling (RGS) and Nuclear Receptors (NR) and 2) to construct domain distance matrices to see how these proteins are related through domains. Both RGS and NR are medically important signal-transducing protein families, similarly important as GPCRs described in the previous section, and both have multiple domains.

Introduction

Regulators of G-protein signaling (RGS) are critical components of many cellular processes and pathways, *e.g.*, intercellular signaling and asymmetric cell division (Wilkie and Kinch, 2005). Most RGS proteins and several of their relatives are involved in G-protein GTPase-activating (GAP) activity. RGS proteins also interact with many other proteins and lipids that may cause positive or negative regulatory functions in addition to, or distinct from their GAP activities. The RGS are related by a conserved RGS domain of ~130 amino acid residues. RGS domains have been found in many species from fungi, *Dictyostelium discoideum*, and animals (Ross and Wilkie, 2000). Family and domain relationship are showing in Figure S5. Many RGS proteins have other domains or motifs coexisted on their sequences.

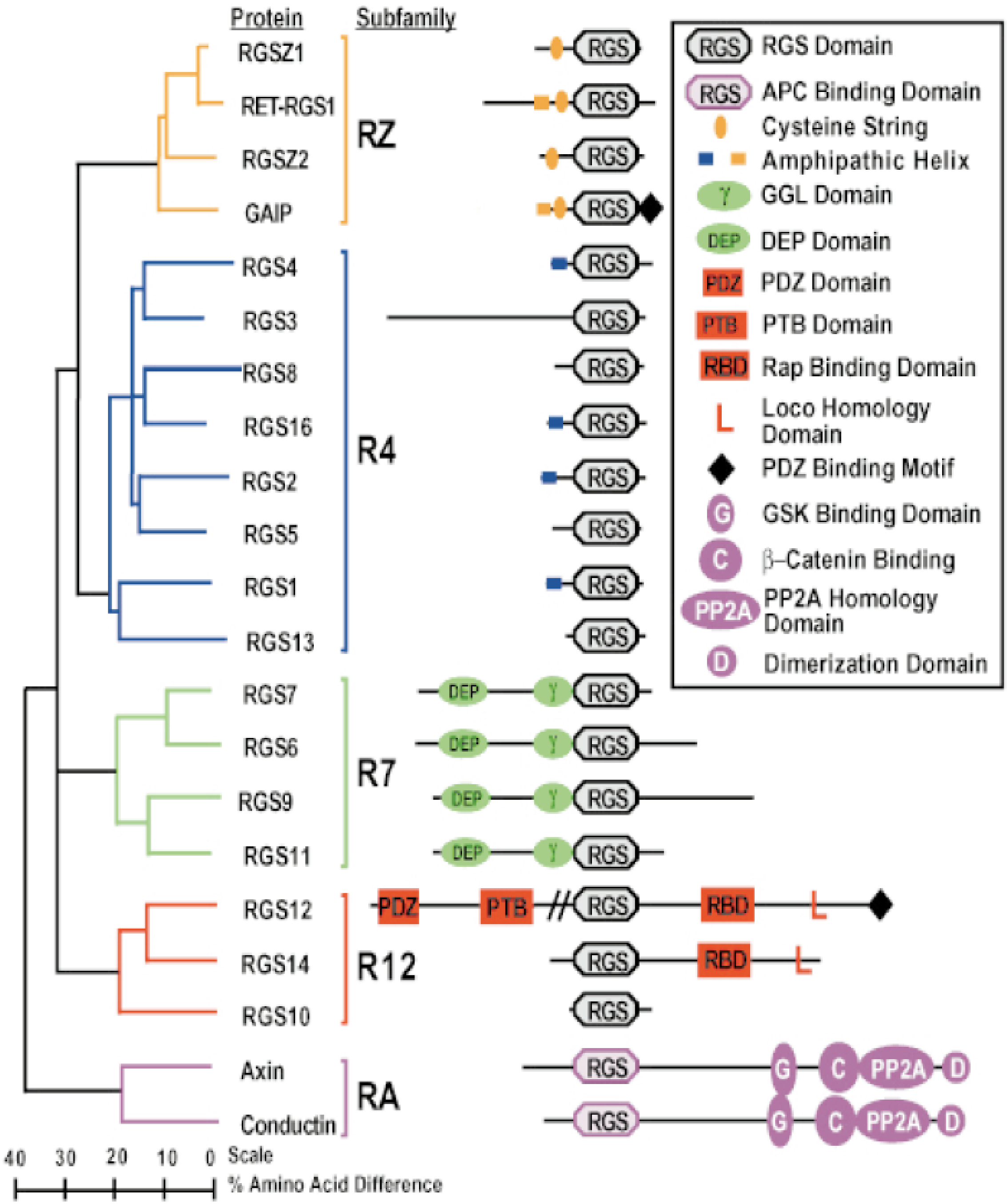

Figure S5: Mammalian RGS proteins.
Left: Phylogenetic tree showing the relationship among five subfamilies RZ, R4, R7, R12 and RA. Right: Domain organization of different RGS sequences (from Ross and Wilkie, 2000). Abbreviations: APC, adenomotous polyposis coli; GGL, Gr-like; DEP, PDZ, and PTB, protein interaction domains; PP2A, protein phosphatase 2A.

Nuclear receptors (NR) are another extremely important proteins that are involved in almost all aspects of normal human physiology and also associated with many human diseases, and thus import therapeutic targets for pharmaceuticals (Olefsky, 2001). Many plant and synthetic chemicals, *e.g.,* pesticides, industrial by-products and plastics components, have also been found to bind to nuclear receptors to trigger or disrupt their natural activities (Thornton, 2003).

Nuclear receptors are multi-domain proteins only found in metazoans that bind to regions on DNA to regulate the transcription of specific genes. Most of nuclear receptors have the same domain arrangement with DNA-binding domain (DBD) and ligand binding domain (LBD) connected by a hinge region. The DNA-binding domain is responsible for targeting the receptor to highly specific DNA sequences and the ligand-binding domain is to recognize specific hormonal and non-hormonal ligands. Usually in signal transduction pathways, nuclear receptors, upon ligand binding, form homo- or hetero-dimers and then to target specific DNA sequences to regulate the expression of the gene (Bertrand, et al., 2004). NR domain structure is presented in Figure S6.

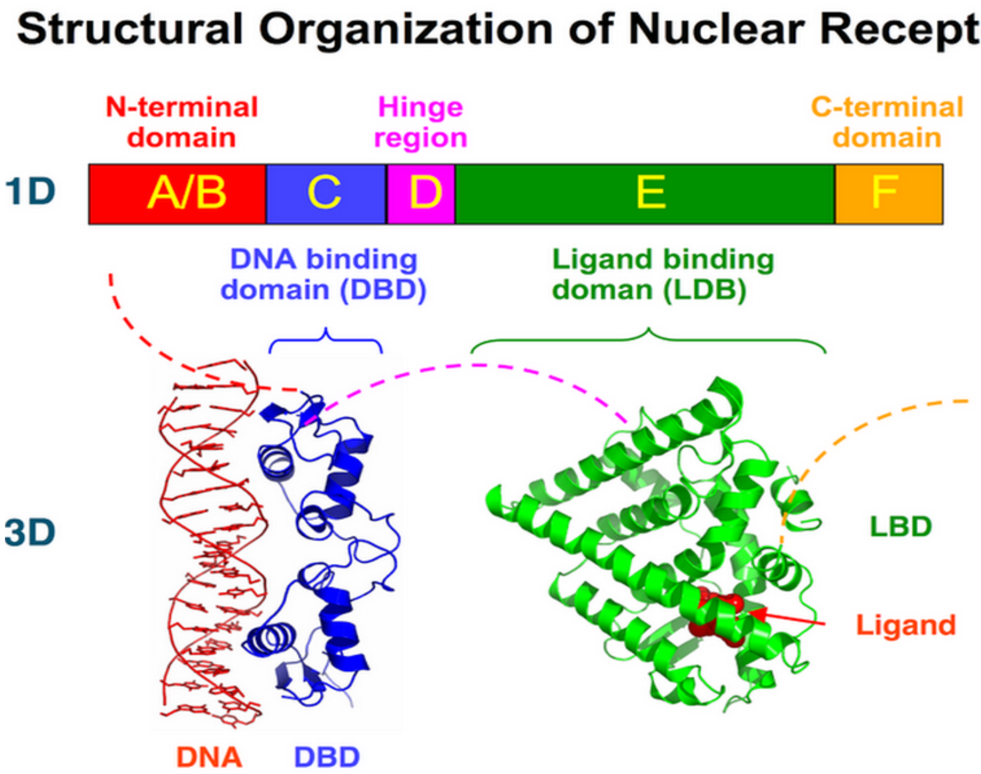

Figure S6: Structural organization of nuclear receptors.
(adapted from http://en.wikipedia.org/wiki/File:Nuclear_Receptor_Structure.png)

RGS and NR protein compilation from the mouse genome

RGS sequences were found by profile HMM search using program hmmsearch with default parameters from the HMMER v3.0 package (Eddy, 1998). Pfam (Punta, et al., 2012) v26.0 family PF00615 (RGS) and PF09128 (RGS-like) against the translated Con_Mouse database ("Ref_proteins"). We used 10 as the threshold of the domain E-value calculated by HMMER to find all potential RGS sequence candidates. E-value of 10 means that 10 expected false positive RGS sequences will be included in the result from the entire mouse protein set. We used this fairly large threshold to try to include more RGS.

Nuclear receptor (NR) sequences were compiled in the exactly same fashion except that we used Pfam family PF00104 (Ligand-Binding Domain, LBD) and PF00105 (DNA-Binding Domain, DBD) to search for NR sequences.

In total, we obtained 154 RGS proteins ("RGS.fa") and 156 NR proteins ("NR.fa").

Construction of domain distance matrices

In order to prepare "domain evolution network" similar to the idea proposed by Holloway and Beiko (2010), we identified different domains co-existing in all RGS and NR protein sequences identified and constructed domain-to-protein distance matrices. This was part of a larger collaborative project.

We used the compiled RGS sequence set to profile HMM search using HMMER program hmmsearch with default parameters against the Pfam database to find other domains on the sequences. We again used 10 as the threshold of the domain E-value calculated by HMMER to include more domains. This step was done to identify all other domains that coexist with RGS or RGS-like domain in our RGS sequences. We extracted the sequence of each domain from each RGS sequence. Each sequence of the domains found in an RGS protein was then used as the query for the similarity search using blastp against each sequence in the RGS sequence set. Each sequence in the RGS set was treated as a single database in the blastp search. The E-value was used as a distance measure between each domain sequence of an RGS sequence and each of RGS sequences. "0" or a very small E-value is expected if a domain sequence is highly similar to a region of the sequence searched. A large E-value is expected, on the other hand, when a domain from sequence “A” is used for the search against sequence “B” where this domain does not exist. If a domain was not found on a sequence within the threshold E-value of 10, we used 2870 as the maximum distance (based on the average lengths of query (100) and subject (700), we set the search space to be a constant number 70000, and the possible maximum E-value would be 2870 (based on λ=0.267 and K=0.041, using BLOSUM62)). We identified 48 domains from 154 RGS proteins, and the above process was repeated for every domain of every RGS sequence against all RGS sequences. An example of an RGS distance matrix is partially shown in Table S5.

We identified 30 domains from 156 NR proteins, and the distance matrices were compiled in the same fashion. An example of an NR distance matrix is partially shown in Table S6.

In total, we produced 154 (48 domains X 154 RGS protein sequences) and 156 (30 domains X 156 NR protein sequences) distance matrices for RGS and NR proteins, respectively. All distance matrices can be found at: http://bioinfolab.unl.edu/emlab/gpcr_mouse/

These distance matrices serve as the input for reconstructing the domain evolution network.

Table S5: An example of an RGS domain-to-protein distance matrix.[a]

Sequence Domain	NM_207213.1|ENSMUST00000041582|uc009lpu.1	NM_008488.1|uc009fqv.1	ENSMUST00000021642	ENSMUST00000097460
PF12761	3.00E-30	12	34	91
PF06246	7.00E-22	1.9	41	2870
PF08628	2.00E-62	5.6	39	52
PF00787	1.00E-63	0.46	0.86	0.026
PF02194	1.00E-92	0.81	0.099	1.6
PF00615	3.00E-67	0.24	1.00E-06	1.00E-08
PF09128	2870	2870	2870	2870
PF00621	2870	2870	2870	2870
PF00610	2870	2870	2870	2870
PF06718	2870	2870	2870	2870
PF00631	2870	2870	2870	2870
PF00169	2870	2870	2870	2870
PF00018	2870	2870	2870	2870
PF00435	2870	2870	2870	2870
PF04803	2870	2870	2870	2870
PF02284	2870	2870	2870	2870
PF08833	2870	2870	2870	2870
PF00778	2870	2870	2870	2870
PF02188	2870	2870	2870	2870
PF02196	2870	2870	2870	2870
PF11470	2870	2870	2870	2870
PF00069	2870	2870	2870	2870
PF07714	2870	2870	2870	2870
PF11333	2870	2870	2870	2870

[a] *This table shows the distance (BLAST E-value) between each domain sequence from the reference sequence (the first column) and any similar region found in each sequence.*

Table S6: Example of an NR domain-to-protein distance matrix.[a]

<table>
<tr><th>Sequence
Domain</th><th>ENSMUST00000110418</th><th>ENSMUST00000044858|uc008cay.1</th><th>NM_010936.2|ENSMUST00000023504|uc007zeq.1</th><th>NM_011934.2|ENSMUST00000021680|ENSMUST00000110207|ENSMUST00000167891|uc007oht.1|uc007ohw.1</th></tr>
<tr><td>PF12497</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF00104</td><td>5.00E-04</td><td>5.00E-108</td><td>5.00E-13</td><td>4.00E-22</td></tr>
<tr><td>PF02159</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF03489</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF00105</td><td>2.00E-21</td><td>1.00E-42</td><td>1.00E-19</td><td>3.00E-23</td></tr>
<tr><td>PF11825</td><td>5.2</td><td>6.00E-20</td><td>6.8</td><td>4.9</td></tr>
<tr><td>PF07352</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF02166</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF07371</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF02161</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF12837</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF06600</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF12577</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF12782</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF02535</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF08143</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF03408</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF12390</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF07967</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF06827</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF03854</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF10080</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF06215</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
<tr><td>PF03468</td><td>2870</td><td>2870</td><td>2870</td><td>2870</td></tr>
</table>

[a] *This table shows the distance (BLAST E-value) between each domain sequence from the reference sequence (the second column) and any similar region found in each sequence.*

References for Supplementary Materials

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *J Mol Biol*, **215**, 403-410.

Bertrand, S., Brunet, F.G., Escriva, H., Parmentier, G., Laudet, V. and Robinson-Rechavi, M. (2004) Evolutionary genomics of nuclear receptors: from twenty-five ancestral genes to derived endocrine systems, *Mol Biol Evol*, **21**, 1923-1937.

Eddy, S.R. (1998) Profile hidden Markov models, *Bioinformatics*, **14**, 755-763.

Holloway, C. and Beiko, R.G. (2010) Assembling networks of microbial genomes using linear programming, *BMC Evol Biol*, **10**, 360.

Karplus, K., Barrett, C. and Hughey, R. (1998) Hidden Markov models for detecting remote protein homologies, *Bioinformatics*, **14**, 846-856.

Kim, J., Moriyama, E.N., Warr, C.G., Clyne, P.J. and Carlson, J.R. (2000) Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties, *Bioinformatics*, **16**, 767-775.

Lu, G., Wang, Z., Jones, A.M. and Moriyama, E.N. (2009) 7TMRmine: a Web server for hierarchical mining of 7TMR proteins, *BMC Genomics*, **10**, 275.

Olefsky, J.M. (2001) Nuclear receptor minireview series, *Journal of Biological Chemistry*, **276**, 36863-36864.

Punta, M., Coggill, P.C., Eberhardt, R.Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E.L., Eddy, S.R., Bateman, A. and Finn, R.D. (2012) The Pfam protein families database, *Nucleic Acids Res*, **40**, D290-301.

Ross, E.M. and Wilkie, T.M. (2000) GTPase-activating proteins for heterotrimeric G proteins: regulators of G protein signaling (RGS) and RGS-like proteins, *Annu Rev Biochem*, **69**, 795-827.

Thornton, J.W. (2003) Nonmammalian nuclear receptors: Evolution and endocrine disruption, *Pure and Applied Chemistry*, **75**, 1827-1839.

Vroling, B., Sanders, M., Baakman, C., Borrmann, A., Verhoeven, S., Klomp, J., Oliveira, L., de Vlieg, J. and Vriend, G. (2011) GPCRDB: information system for G protein-coupled receptors, *Nucleic Acids Res*, **39**, D309-319.

Wilkie, T.M. and Kinch, L. (2005) New roles for Galpha and RGS proteins: communication continues despite pulling sisters apart, *Curr Biol*, **15**, R843-854.

Wistrand, M., Kall, L. and Sonnhammer, E.L. (2006) A general model of G protein-coupled receptor sequences and its application to detect remote homologs, *Protein Sci*, **15**, 509-521.