SAM
"Significance Analysis of Microarrays"
Users guide and technical document

Gil Chu [*] Michael Seo [†] Jun Li [‡] Balasubramanian Narasimhan [§]
Robert Tibshirani [¶] Virginia Tusher [‖]

Acknowledgments: We would like to thank the R core team for permission to use the R statistical system and Charlie Tibshirani for writing the error checking code for version 2.1. We also thank Trevor Hastie for helpful suggestions.

Contents

[*]Department of Biochemistry, Stanford University, Stanford CA 94305. Email: chu@cmgm.stanford.edu.
[†]Department of Statistics, Stanford University, Stanford CA 94305. Email: swj8874@gmail.com.
[‡]Department of Statistics, Stanford University, Stanford CA 94305. Email: jun@stat.stanford.edu.
[§]Department of Statistics and Department of Health Research & Policy, Stanford University, Stanford CA 94305. Email: naras@stat.stanford.edu.
[¶]Department of Health Research & Policy and Department of Statistics, Stanford University, Stanford CA 94305. Email: tibs@stanford.edu.
[‖]Department of Biochemistry, Stanford University, Stanford CA 94305. Email: goss@cmgm.stanford.edu.

List of Figures

List of Tables

1 Important Announcement

To foster communication between SAM users and make new announcements, a new Yahoo group has been established. See `http://groups.yahoo.com/group/sam-software`.

2 Summary of Changes

The following are changes since the initial release of SAM 1.0.

2.1 Changes in SAM 5.0

We built a new web application using Shiny. We have written instructions on how to use this new application in section 7 and removed instructions for the previous version. Slight changes have been made when calculating the estimated miss rates, false discovery rate, and q-values. We now use exact delta values that is specified, rather than an estimated delta value that is closest to it.

Other results we get from running SAM remains the same as the previous version. Also, SAM no longer accpets .xls file. Please convert the data into .xlsx before running SAM.

2.2 Changes in SAM 4.0

We have added a new method called `SAMSeq`, for testing differential expression from RNAseq data. For this option, there is a new button on the opening SAM screen. The background details are given in [5]. The main change in SAM to handle sequence data is in the construction of the SAM score. This construction uses resampling and the nonparametric statistics such as the Wilcoxon for the two-class case. SAM 4.0 uses the samr package v2.0.

2.3 Changes in SAM 3.03

This version calls the R package samr v1.26. An inconsistency was found in the way that fold change was computed for logged data. The means in each group were computed on the unlogged data rather than the logged data. This is now fixed. Note that if the user mistakenly clicks the *unlogged* button when the data is actually on a log scale, the resulting fold changes might turn out to be negative!

2.4 Changes in SAM 3.02

This version calls the R package samr v1.25, in which two bugs were fixed. The standard deviation in the denominator was not being computed correctly for the quantitative option, and if there were < 500 genes in the dataset. the input value of the exchangeability factor was not being used.

2.5 Changes in SAM 3.01

A bug in the survival analysis code was fixed and this manual was updated with more information on the interpretation of time series analysis.

2.6 Changes in SAM 3.0

SAM now has facilities for Gene Set Analysis [2], a variation on the Gene Set Enrichment Analysis technique of [7]. Details are in section 11.

2.7 Changes in SAM 2.23

Numerous small bug fixes, including parsing of the first row in time course experiments and prevention of overflow in the plot for large datasets.

The method for estimating the tail strength standard error was changed. The existing method produced estimates that were generally too small.

2.8 Changes in SAM 2.21

- SAM now reports the overall tail strength for the dataset on the SAM plot. See Taylor and Tibshirani (1995)- http://www-stat.stanford.edu/ tibs/ftp/tail.pdf for details.
- A bug in the plotting routine that bombed when there were more than 32,000 points (a limitation of Excel) was fixed.
- Some better error reporting was added.

2.9 Changes in SAM 2.20

- A new facility for assessment of sample sizes has been added!
- For time course data, SAM now uses the internal standard error of the slope or signed area from each time course (thanks to Kate Rubins for the suggestion).
- As a result, for unpaired time course data, it is now allowable to have only one time course in one or more classes. SAM computes the gene scores but warns the user than the SAM plot and FDRs will be unreliable. [Since there is not enough data to carry out permutations]. Similarly for paired and one class time course data.

2.10 Changes in SAM 2.10

- Added more thorough error checking of the input data response row (1)
- Sped up the computation of the significant gene list, and made local FDR computation optional in the controller window. The default is now false, which speeds up the gene list computation.
- Fixed some small bugs

2.11 Changes in SAM 2.01

Version 2.01 corrects several problems since release 2.0. We believe it is much improved as a result.

- Fixed One-sample case bug where a large number of samples resulted in large storage allocations

- Fixed small problem with gene list and qvalues, when only 1 gene called significant
- Fixed problem with q-value, when only positive (or negative) genes are significant
- Fixed the validation checks when data is in multiple sheets

2.12 Changes in SAM 2.0

This is a major new release of SAM. The numerical computations are now done using the R package `samr` version 1.0. In addition there are many new features:

- Facilities for two class, one class and paired *time course* data
- Non-parametric tests- wilcoxon and rank regression
- Pattern discovery via eigengenes
- Local false discovery rates, and miss rates
- A faster, more accurate imputation engine
- Changes were made in estimation π_0 for the `multiclass` option, and in the score for the `quantitative` section. See section 12 for details.

Due to changes in the internals of SAM, results using SAM 2.x will be close to, but not exactly those obtained with earlier versions of SAM.

2.13 Changes in SAM 1.21

Two bugs were fixed.

- A bug relating to what SAM perceives as a large number of permutations was fixed. The default was very naive.
- A bug in adding the imputed data sheets for multiple sheets was fixed.

2.14 Changes in SAM 1.20

- SAM can now handle a large number of samples. Input data can span several sheets (contiguous or non-contiguous). An example file, named `twoclassbig.xls` included with the distribution.
- A bug in the calculation of FDR for paired data, with a fold change specified, was fixed.

Versions 1.16–1.19 were skipped.

2.15 Changes in SAM 1.15

Bugfix release. A bug that caused SAM to bomb during the calculation of $\hat{\pi_0}$ was fixed.

2.16 Changes in SAM 1.13

Bug fix release. A bug was fixed in the calculations for Censored Survival data. Everyone is advised to upgrade to this version.

2.17 Changes in SAM 1.12

This is mostly a bug fix release. Users of SAM 1.10 should immediately upgrade to this release.

- Bug fix: An error in the calculation of the fold-change was fixed. The criterion for applying fold-change to significant genes was also corrected. We thank alert users for catching this.
- By popular request, a new column called **Fold Change** has been added to the significant genes list. This applies only to Two-class and Paired responses. Where the fold change cannot be calculated, it is flagged with an `NA` for "Not Applicable."

2.18 Changes in SAM 1.10

- Bug fixes: a serious bug in the imputation was fixed. The bug caused some data to be imputed with the value 65535. A symptom of this bug was that the plot would have a strange appearance due to the scaling.
- A new facility for block permutations has been added, to handle different experimental conditions such as array batches. See section 5.3.
- In cases where the total number of possible permutations is small, the full set of permutations is used rather than a random sampling.
- The "threshold" now is replaced by a "fold change" criterion, and now handles logged (base 2) and unlogged data appropriately. The fold change applies only to two-class or paired data.
- We have added a new output column to the significant gene list: the "q-value": for each gene, this is the lowest False Discovery Rate at which that gene is called significant. It is like the well-known p-value, but adapted to multiple-testing situations. Q-values were invented by John Storey [6].
- The reported False Discovery Rates are now lower and more accurate than in Version 1.0. They are scaled by a factor $0 \leq \hat{\pi}_0 \leq 1$, that is now displayed on all output. See Section 12 and reference [6].

- Significant gene ids are now linked directly to the Stanford SOURCE web database. Several options for search are provided. Default is by gene name.
- For two-class and paired data, one must now specify whether data is in log-scale or not.
- Stricter checks on response variable values are now performed.
- Several efficiency issues have been addressed.
- The web version of SAM is no longer under development. Hence we have removed it from this manual. The old version still works for the time being, and the version 1.0 manual contains documents it.

Due to changes in the internals of SAM, results using SAM 1.10 will be close to, but not exactly those obtained with SAM 1.0.

We have also updated the FAQ with the latest information. See section 13.

3 Introduction

SAM (Significance Analysis of Microarrays) is a statistical technique for finding significant genes in a set of microarray experiments. It was proposed by Tusher, Tibshirani and Chu [9]. The software was written by Michael Seo, Balasubramanian Narasimhan and Robert Tibshirani.

The input to SAM is gene expression measurements from a set of microarray experiments, as well as a response variable from each experiment. The response variable may be a grouping like *untreated, treated* (either unpaired or paired), a multiclass grouping (like breast cancer, lymphoma, colon cancer), a quantitative variable (like blood pressure) or a possibly censored survival time. SAM computes a statistic d_i for each gene i, measuring the strength of the relationship between gene expression and the response variable. It uses repeated permutations of the data to determine if the expression of any genes are significantly related to the response. The cutoff for significance is determined by a tuning parameter **delta**, chosen by the user based on the false positive rate. One can also choose a **fold change** parameter, to ensure that called genes change at least a pre-specified amount. See section 12.

4 Examples

If you download the SAMR package from CRAN, some examples are stored inside `inst` folder. We will upload examples in a separate place for convenience soon. These examples are meant to familiarize the users with the format in which SAM expects the data.

We briefly describe the examples below.

Two Class An example of two class, unpaired data.

Two Class (Missing) An example of two class, unpaired data, with missing data.

Two Class (Blocked) An example of two class, unpaired data, with experimental blocks defined.

Two Class (Big) An example of two class, unpaired data with multiple sheets

Two Class (Unpaired Timecourse) An example of two class unpaired timecourse data

Two Class (Paired Timecourse) An example of two class paired timecourse data

Paired An example of paired data.

One Class An example of oneclass data.

One Class (Timecourse) An example of one class timecourse data

Multi Class An example of multiclass response.

Survival An example of censored survival data. Note the format of the labels in the first row!

Quantitative An example of quantitative data.

Pattern Discovery An example of data for pattern discovery

Two Class Sequence An example of two class, unpaired data from RNA-seq experiments.

Paired Sequence An example of two class, paired datafrom RNA-seq experiments.

Instructions on using SAM on these examples is discussed in section 7.

5 Data Formats

The data should be put in an Excel spreadsheet. The first row of the spreadsheet has information about the response measurement; all remaining rows have gene expression data, one row per gene. The columns represent the different experimental samples.

- The first line of the file contains the response measurements, one per column, starting at column 3. This is further described below in section 5.1.
- The remaining lines contain gene expression measurements one line per gene. We describe the format below.

Column 1 This should contain the gene name, for the user's reference.

Column 2 This should contain the gene ID, for the user's reference.

Remaining Columns These should contain the expression measurements as numbers. Missing expression measurements should be noted as either blank or non-numeric values.

For sequencing data, the values are counts and hence must be non-negative.

5.1 Response Format

Table 1 shows the formats of the response for various data types. A look at the example files is also informative.

Response type	Coding
Quantitative	Real number eg 27.4 or -45.34
Two class (unpaired)	Integer 1, 2
Multiclass	Integer 1, 2, 3, ...
Paired	Integer -1, 1, -2, 2, etc. eg - means Before treatment, + means after treatment -1 is paired with 1, -2 is paired with 2, etc.
Survival data	(Time, status) pair like (50,1) or (120,0) First number is survival time, second is status (1=died, 0=censored)
One class	Integer, every entry equal to 1
Time course, two class (unpaired)	(1 or 2)Time(t)[Start or End]
Time course, two class (paired)	(-1 or 1 or -2 or 2 etc)Time(t)[Start or End]
Time course, one class	1Time(t)[Start or End]
Pattern discovery	eigengenek, where k is one of 1,2,... number of arrays

Table 1: Response Formats

A *Quantitative* response is real-valued, such as blood pressure. *Two class (unpaired)* groups are two sets of measurements, in which the experiment units are all different in the two groups. (i.e. control and treatment groups with samples from different patients). With a *Multiclass* response, there are more than two groups, each containing different experimental units. This is a generalization of the *unpaired* setup to more than 2 groups. *Paired* groups are two sets of measurements in which the same experimental unit is measured in each group (i.e. samples from the same patient, measured before and after a treatment. *Survival* data consists of a time until an event (such as death or relapse), possibly censored. In the *One class* problem, we are testing whether the mean gene expression differs from zero. For example, each measurement might be the log(red/green)

ratio from two labelled samples hybridized to a cDNA chip, with green denoting before treatment and red, after treatment. Here the response measurement is redundant and is set equal to all 1s.

A *Time course* response means that each experimental unit is measured at more than one time point. The experimental units themselves can fall into a two class, one class, or a two-class paired design. SAM summarizes each time course by a slope or a signed area, and then treats the summarized data in the same way as it treats a two class, one class, or a two-class paired design. In *Pattern discovery*, no explicit response parameter is specified. Instead, the user specifies the eigengene number, eg 1,2, etc. SAM then computes that eigengene (principal component) of the expression data, and treats that eigengene as if it were a quantitative response. It looks for genes that are highly correlated with that eigengene and also reports the eigengene itself. The only difference with a quantitative response is the way in which permutations are generated (details later).

5.2 Example Input Data file for an unpaired problem

The response variable is $1 = untreated$, $2 = treated$. The columns are gene name, gene id, followed by the expression values.

The first row contains the response values.

		1	1	2	2	1	1	2	2
GENE1	GENEID101	7.64	-0.50	-1.95	10.12	-10.77	-4.47	-7.65	7.58
GENE2	GENEID102	38.10	4.86	7.87	-13.59	-9.79	-13.46	-8.91	-5.07
GENE3	GENEID103	21.15	5.96	3.20	-4.74	-3.70	-12.35	-10.17	0.63
GENE4	GENEID104	187.21	-23.81	16.76	14.10	-99.76	-89.11	-10.92	5.52

Table 2: Example Dataset for an unpaired problem

Note that there are two blank cells at the beginning of line 1. The gene expression measurements can have an arbitrary number of decimal places.

5.3 Block Permutations

Responses labels can be specified to be in blocks by adding the suffix *BlockN*, where N is an integer, to the response labels. Suppose for example that in the two-class data of section 2, samples 1,3,5,7 came from one batch of microarrays, and samples 2,4,6,8 came from another batch. We call these batches "blocks." Then we might not want to mix up the batches in our permutations of the data, in order to control for the array differences. That is, we'd like to allow permutations of the samples within the set 1,3,5,7 and within the set 2,4,6,8, but not across the two sets. We indicate the blocks (batches) as follows:

		1Block1	1Block2	2Block1	2Block2	1Block1	1Block2	2Block1	2Block2
GENE1	GENEID101	7.64	-0.50	-1.95	10.12	-10.77	-4.47	-7.65	7.58
GENE2	GENEID102	38.10	4.86	7.87	-13.59	-9.79	-13.46	-8.91	-5.07
GENE3	GENEID103	21.15	5.96	3.20	-4.74	-3.70	-12.35	-10.17	0.63
GENE4	GENEID104	187.21	-23.81	16.76	14.10	-99.76	-89.11	-10.92	5.52

Table 3: Example Dataset for a Blocked unpaired problem

For example, "1Block1" means treatment 1, block (or batch) 1. "1Block2" means treatment 1, block (or batch) 2. In this example, there are $4! = 24$ permutations within block 1, and $4! = 24$ permutations within Block 2. Hence the total number of possible permutations is $24 \cdot 24 = 576$. If the block information is not indicated in line 1, all permutations of the 8 samples would be allowed. There are $8! = 40320$ such permutations.

Please note that block permutations cannot be specified with Paired response as there is an implicit blocking already in force.

5.4 Time course data

Response labels can be specified to be in time course by adding the suffix *TimeT*, where T is a real number, to the response labels. Suppose for example that we have experimental units in each of two classes, and each unit is measured at two or more time points. Here is a typical response line:

1Time1Start	1Time2	1Time3End	1Time1Start	1Time2.5	1Time3.4End	2Time0.5Start	2Time1.2	2Time2.75	2Time3.7End

Table 4: Example Dataset for a unpaired two class time course problem

The first experimental unit is in class 1, and was measured at times 1, 2, and 3. The second experimental unit is in class 1, and was measured at times 1, 2.5, and 3.4. The third experimental unit is in class 2, and was measured at times 0.5, 1.2, 2.75, and 3.7. Note that the times can be any real numbers, and the number of times can be different for each experimental unit (but must be at least 2). The "Start" and "End" suffixes indicate the first and last arrays for a given experimental unit. For a paired data, the format is the same. The leading class label is -1 or 1, or -2 or 2, as in the paired data response format. For oneclass time courses, the leading class label is a 1.

5.5 Normalization of experiments

Different experimental platforms require different normalizations. Therefore, *the user is required to normalize the data from the different experiments (columns) before running SAM*. However, for

convenience SAM v2.0 now offers normalization via simple median centering of the arrays.

For cDNA data, centering the columns of the expression matrix (that is, making the columns median equal to zero) is often sufficient. For oligonucleotide data, a stronger calibration may be necessary: for example, a linear normalization of the data for each experiment versus the row-wise average for all experiments.

6 Handling Missing Data

SAM imputes missing values via a K-Nearest Neighbor algorithm normalization. Full details may be found in [4] and [8]. The user specifies the number of neighbors k (default=10). Here is how it works:

1. For each gene i having at least one missing value:
   (a) Let S_i be the samples for which gene i has no missing values.
   (b) Find the k nearest neighbors to gene i, using only samples S_i to compute the Euclidean distance. When computing the Euclidean distances, other genes may have missing values for some of the samples S_i; the distance is averaged over the non-missing entries in each comparison.
   (c) Impute the missing sample values in gene i, using the averages of the non-missing entries for the corresponding sample from the k nearest neighbors.
2. If a gene still has missing values after the above steps, impute the missing values using the average (non-missing) expression for that gene.

If the number of genes is large, the near-neighbor computations above can take too long. To overcome this, we combine the K-Nearest Neighbor imputation algorithm with a **Recursive Two-Means Clustering** procedure:

1. If number of genes p is greater than p_{max} (default 1500):
   (a) Run a two-means clustering algorithm in gene space, to divide the genes into two more homogeneous groups. The distance calculations use averages over non-missing entries, as do the mean calculations.
   (b) Form two smaller expression arrays, using the two subsets of genes found in (a). For each of these, recursively repeat step 1.
2. If p is less than p_{max}, impute the missing genes using K-Nearest-Neighbor averaging.

7 Running SAM

Download the samr package in R. Load in samr library and type in `runSAM()` to run SAM. Once the SAM interface is up, upload an .xlsx data file by clicking on the `Choose File` button. Note .xls file will not work any more. The data has to follow the format specified in 5. Then, you need to select the type of response variable, data type (microarray vs. sequencing), analysis type (individual genes vs. gene sets), and if desired, any of the values of the default parameters. Each of the response types require different selections and some buttons will appear and disappear based on the response type. Figure 1 shows different selections for the *quantitative* option. You need to provide a .gmt file to perform gene set analysis.

If you press the `Run` button, a result screen as in figure 2 will appear. Under the `SAM Plot` tab, a SAM plot will show up. Positive significant genes are labelled in red and negative significant genes are in green. The `Delta Table` tab lists the number of significant genes and the false positive rate for a number of values of Δ. The `Sample Size` tab gives information on FDR and power for various sample sizes.

Once the `Run` button is clicked, another panel shows up as shown in the figure 3. These parameters can change the values in plots and tables. For instance, you can change the Δ parameter and examine the effect on the false positive rate. If you want a more stringent criterion, you can try setting a non-zero `Minimum fold change` parameter (see section 12 for details). Changes in these parameters is reflected right away in the plots and tables and do not require you to press the `Run` button again.

To save the results in excel, you need to specify where you want to save and what you want to name the file and press the `Save` button. The default is the current directory and a file name called *result*. It takes a few seconds to save plots and tables in an excel format. Note that if there is already an excel file with the same name, the previous file is replaced with a new file. If you have any missing data in your data, a new worksheet named `Imputed Data` containing the imputed dataset is added to the workbook. This data can be used in subsequent analyses to save time. If there is no missing data, this worksheet is not added.

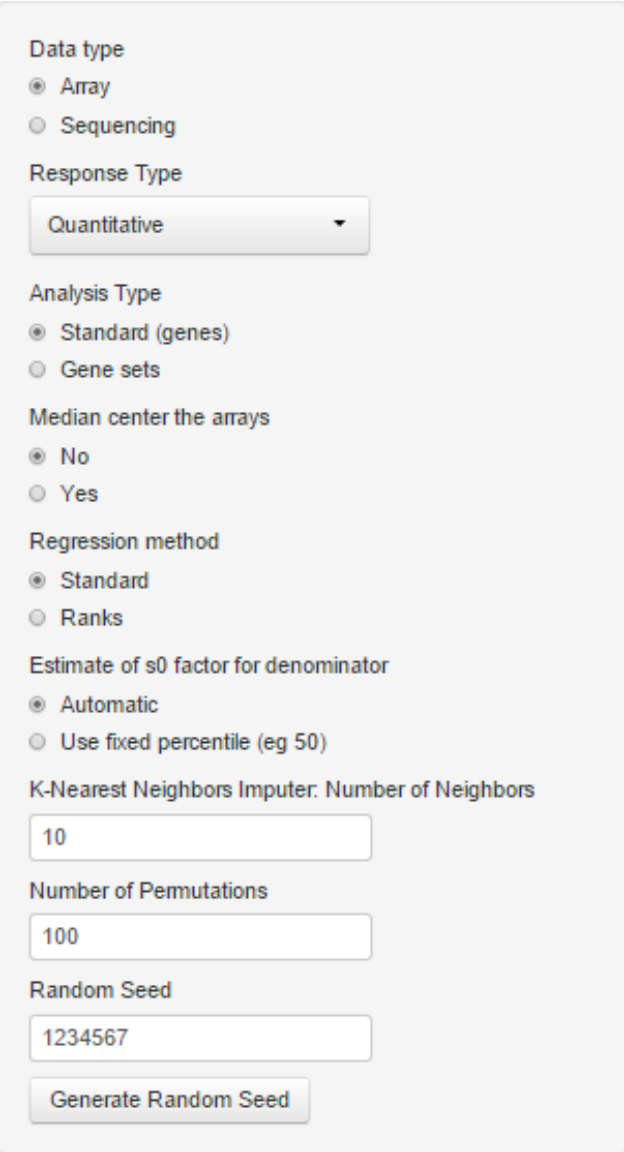

Figure 1: Selecting type of analysis to run

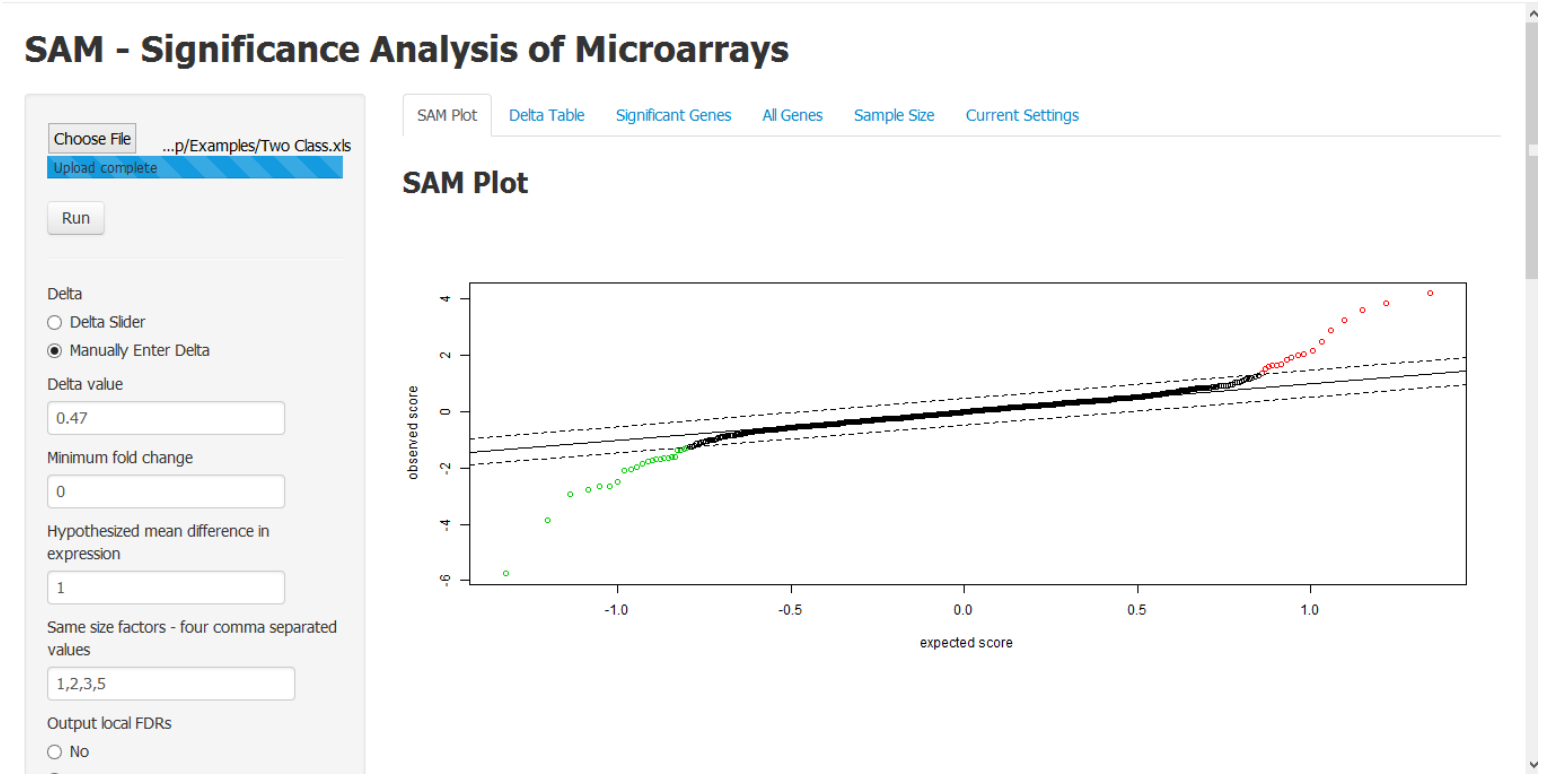

Figure 2: SAM result

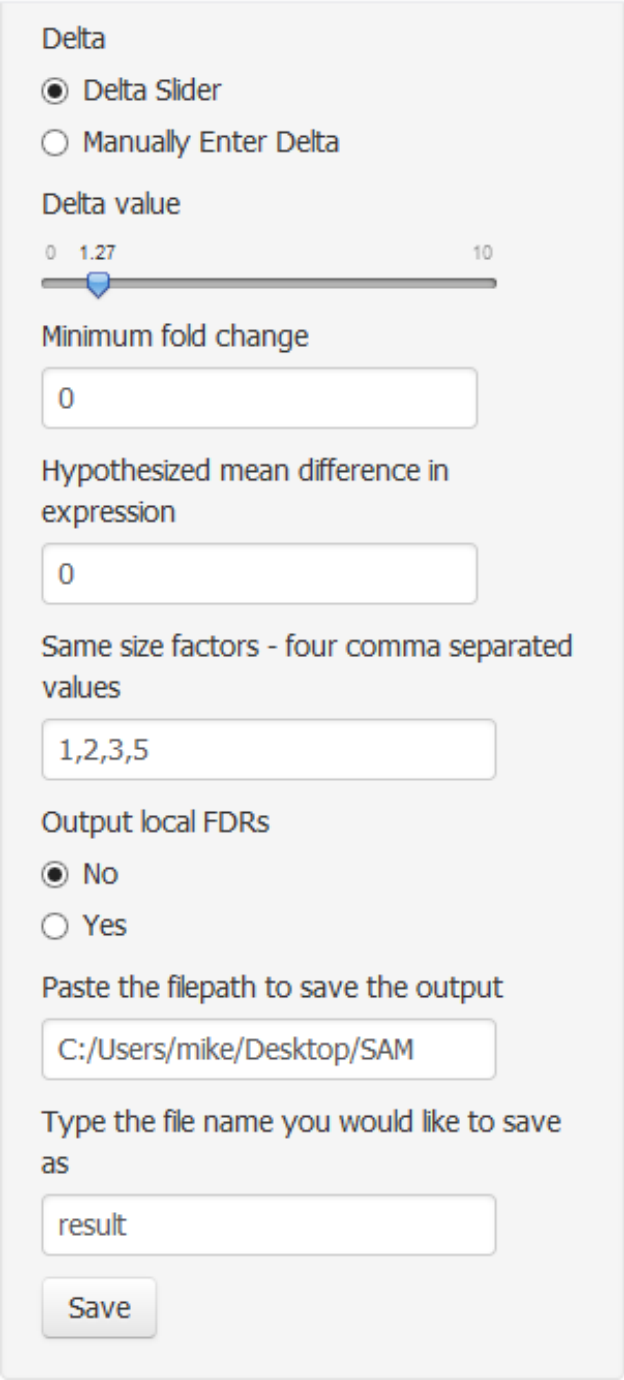

Figure 3: SAM result control panel

7.1 Format of the Significant gene list

For reference, SAM numbers the original genes, in their original order, as 1,2,3, etc. In the output, this is the **Row number**. The output for the list of significant genes has the following format:

Row number The row in the data sheet.

Gene name The gene name specified in the first column selected data rectangle. This is for the user's reference.

Gene id The gene id specified in the second column selected data rectangle. This is for the user's reference.

SAM score(d) The T-statistic value.

Numerator The numerator of the T-statistic.

Denominator($s + s_0$) The denominator of the T-statistic.

q-value This is the lowest False Discovery Rate at which the gene is called significant based on the work of John Storey [6] who invented q-values. It is like the familiar "p-value", adapted to the analysis of a large number of genes. The q-value measures how significant the gene is: as $d_i > 0$ increases, the corresponding q-value decreases.

Local FDR This is the false discovery rate for genes with scores d that fall in a window around the score for the given gene. This is in contrast to the usual FDR, which is the false discovery rate for a list of genes, whose scores exceed a given threshold. For example, if we set Δ to a certain value, we might get upper and lower score cutpoints of ± 3, yielding 100 genes with an FDR of 10%. While the local FDR for genes with scores near ± 3 is probably $> 10\%$, the local FDR for genes with the largest scores (say ± 6), might be close to zero. Local false discovery rates are discussed in [3] and [1].

NOTE: In our experience, the local FDR is inherently more difficult to estimate than the usual (global) FDR. Hence, the usual FDR is a more reliable measure of the accuracy of the gene list. In particular, we use a window of at least 50 genes to estimate the local FDR at each point. This means that for the most extreme genes, the window will consist mostly of genes that are less significant than the target gene. Thus, the reported local FDR will be too large for these genes, and larger than the global FDR. The local FDR is most accurately estimated for genes near the middle of the distribution.

For *multiclass* data, the *contrast* for each gene in each class, is also shown. This is the standardized mean difference between the gene's expression in that class, versus its overall mean expression, The 2.5 and 97.5 percentiles of this quantity over permutations is shown for reference.

Thus for a gene that is significant overall, one can determine which class difference(s) caused it to be significant.

The numerator, denominator and q-value are further explained in the technical section below. The list is divided into positive and negative genes, having positive or negative score d_i. Positive score means positive correlation with the response variable: e.g. for group response 1,2, positive score means expression is higher for group 2 than group 1.

For a *survival time response*, SAM computes the Cox score test for each gene. Thus a *positive score* (red genes in the SAM plot) means that *higher expression correlates with higher risk, i.e. shorter survival*. The reverse is true for negative scores (green genes): a negative score means higher expression correlates with lower risk, i.e. longer survival.

[We had this wrong in some earlier versions of this manual]!

7.2 The Miss rate table

In any testing problem, it is important to consider not only false positive rates (i.e. FDRs) but also false negative rates. For this purpose, a *miss rate* table is also printed. It gives the estimated false negative rate for genes that do not make the list of significant genes. For example, suppose we set Delta to a certain value, giving upper and lower score cutpoints of ± 3 and yielding 100 significant genes with an FDR of 10%. The miss rate table might tell us that the miss rate for scores in the range (2.5, 3) is 40%. That means that 40% of the genes with scores in that range, are false negatives, i.e. are actually differentially expressed.

8 Interpretation of SAM output

The three panels of figure 4 shows the SAM plots for three different datasets. There are 1000 genes in each of the datasets, and 8 samples, 4 each in control and treatment conditions. We carried out SAM analysis using the unpaired (2 class) option. The corresponding false positive tables are shown in table 5.

In dataset (A), there are number of genes above the band in the upper right and below the band in the bottom left. Looking at table 5, we chose $\Delta = .5$. producing about 65 significant genes and about 5.9 false positives on average. The choice of Δ is up to the user, depending how many false positives he/she is comfortable with. The SAM plots can be asymmetric. There can be significant genes in the top right, but not bottom left, or vice-versa.

In dataset (B), there may be no significant genes. With $\Delta = .5$ (shown in the plot), there are 2 called genes but about 1.3 false positive genes on average.

In dataset (C), there are many significant genes. If $\Delta = 0.3$, then nearly 800 genes are called significant and there are only about 23 false positives on the average. This data was generated as

$$x_{ij} = z_{ij} + \mu_{ij} \tag{8.1}$$

for gene $i = 1, 2, \ldots 1000$, sample $j = 1, 2, \ldots 8$. The first four samples are from group 1, the second four from group 2, Here $z_{ij} \sim N(0, 1)$ (standard normal), $\mu_{ij} = 0$ for $j \leq 4$, $\mu_{ij} = \theta_i \sim N(0, 4)$ for $j > 4$. Hence all genes have a true change θ_i in expression from group 2 vs group 1, although it may be small. In the interpretation of the SAM results, one should also look at the score d_i, which is the standardized change in expression. A value of $d_i = 0.5$ (say) may be called statistically significant in example (C), but is it biologically significant? That is up the scientist. Another way to address this issue: set a non-zero `fold change` for calling genes. With a moderate fold change (say 2), far fewer genes will be called in this example.

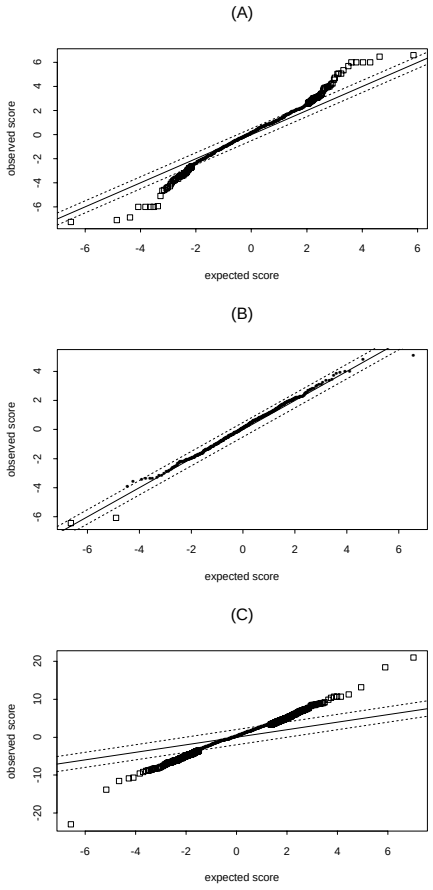

Figure 4: SAM results for 3 different datasets

Avg nb of FPs (A) Nb of significant genes

Δ	#false pos	# called	FDR
0.3	11.7	100	0.117
0.4	9.3	76	0.122
0.5	5.9	65	0.091
0.6	4.4	39	0.113
0.7	3.5	33	0.106
0.8	2.1	29	0.072
0.9	1.6	17	0.094
1.0	1.3	16	0.081

(B)

Δ	#false pos	# called	FDR
0.3	4.8	2	2.40
0.4	1.8	2	0.90
0.5	1.3	2	0.65
0.6	0.6	2	0.30
0.7	0.3	2	0.15
0.8	0.2	0	Inf
0.9	0.2	0	Inf
1.0	0.2	0	Inf

(C)

Δ	#false pos	# called	FDR
0.3	23.4	894	0.026
0.4	10.6	840	0.013
0.5	5.0	818	0.006
0.6	3.1	780	0.004
0.7	1.9	741	0.003
0.8	1.6	708	0.002
0.9	1.4	674	0.002
1.0	0.9	636	0.001

Table 5: SAM false positive results for 3 scenarios

9 Time series data- interpretation of results

For *time course* data in two groups (unpaired or paired) or in one group, you can choose to summarize each time course by a *slope* (least squares slope of expression vs time), or a *signed area*. SAM then treats the summarized data in the same way as it treats a two class, a one class, or a two-class paired design. The slope is useful for finding genes with a consistent increase or decrease over time. The signed area is useful for finding genes that rise and then level off or come back down to their baseline.

For example, for two class unpaired data, if *slope* is chosen, SAM summarizes each time series by a slope. Then the slopes are compared across the two groups. Thus a positive SAM score d_i means that the slopes are larger (on average) in group 2 than in group 1; the opposite is true for a negative d_i. A positive SAM score could mean that the slopes are positive in both groups, but larger in group 2, or they could both be negative but less negative in group 2, or finally they could be negative in group 1 and positive in group 2.

If *signed area* is chosen, the time course profile is shifted so that it is zero at the first time point. Then the area under the time course curve is computed, counting positive area above the line and negative below the line. Then SAM compares the areas across the groups. For example, a positive SAM score d_i in the two group case means that the signed area is larger in group 2 than it is in group 1; the opposite is true for a negative d_i.

10 More options and ideas

- For *one class time course* data, you can also use the *quantitative* option to find genes that match a given pattern. For example, we can generate expression for 1000 genes over 9 time points. The last 900 genes can be just standard Gaussian noise. The first 100 genes go down for the first 3 time points, level off for the next 3, and then increase again for the final 3 time points. The slope of the decrease for the first 3 time points varies from -0.5 to -1.5 for different genes; similarly the increase for the last 3 time points ranges from 0.5 to 1.5.

  To try to find these 100 genes, we set the response row to -3,-2,-1,0,0,0,1,2,3 and then choose the *quantitative* option. This did not do a good job of finding the first 100 genes. The reason is that varying slopes throws off the regression (i.e the correlation measure). However if we select the `rank regression` option, SAM uses the ranks for the response and gene expression values. Now SAM does a good job of isolating the top 100 genes

- Suppose in the above example we had no idea of the predominant patterns in our set of genes. Then we can use the *pattern discovery option*. This is illustrated in the `pattern discovery` worksheet. In the response row we indicate which eigengene we want to find. Usually we would start with 1, and then later try 2, 3, etc. until the FDRs get too high. SAM then computes the requested eigengene, finds the genes that have high correlation with

it, and also prints out the estimated eigengene in the `significant genes` output sheet. The user should make a scatterplot of the eigengene in Excel and study its shape. In the `pattern discovery` worksheet example, SAM finds the generating pattern described above and does a fairly good job of isolating the important genes.

- SAM normally estimates the *exchangeability factor* s_0 by an automatic method described in section (12.1). This estimate is expressed as a percentile of the standard deviation values of all the genes. The role of s_0 is to prevent genes whose expression is near zero (and hence unreliable) from having large scores d_i (such a gene might have $d_i \approx 0/0$. However occasionally one might want to set s_0 manually, and this option is offered in SAM. For example, if you want to get the *standard* Cox scores for an entire gene set for some other purpose, you can set the s_0 percentile to -1 (forcing $s_0 = 0$) and then click `All genes` tab. [Note that setting the s_0 percentile to 0, sets s_0 to the minimum gene standard deviation, which is probably > 0.] You can also try playing with s_0 and seeing how the FDR changes

11 Gene set analysis

SAM now has facilities for Gene Set Analysis [2], a variation on the Gene Set Enrichment Analysis technique of [7]. The idea is to make inferences not about individual genes, but pre-defined sets of genes. The gene set analysis (GSA) method is also implemented in the R package GSA, available from CRAN

The gene set analysis (GSA) method differs from Gene Set Enrichment Analysis in the following ways:

- GSA uses the "maxmean" statistic: this is the mean of the positive or negative part of gene scores d_i in the gene set, whichever is large in absolute value. [In detail: take all of the gene scores d_i in the gene set, and set all of the negative ones to zero. Then take the average of the positive scores and the zeros, giving a positive part average avpos Do the same for the negative side, setting the positive scores to zero, giving the negative part average avneg. Finally the score for the gene set is avpos if $|\mathrm{avpos}| > |\mathrm{avneg}|$, and otherwise it is avneg.]

- Efron and Tibshirani shows that this is often more powerful than the modified Kolmogorov-Smirnov statistic used in GSEA.

- GSA also uses a somewhat different null distribution for estimation of false discovery rates: it does "restandardization" of the genes (rows), in addition to the permutation of columns done in GSEA. This means that a gene set must be unusual BOTH as compared to gene sets of the same size sampled at random from the set of genes represented by the gene set, and as compared to itself, when the outcome labels are permuted.

To do a gene set analysis in SAM, click the `Gene sets` option for Analysis Type and browse for the gene set file (.gmt file) that you want to use. You may use any of the gene set files available at

`http://www-stat/stanford.edu/~tibs/GSA`

or one that you construct yourself.

A .gmt file is a tab-delimited text file, with one row per gene set. The gene set name is in column 1 and the gene set description is in column 2 (this is for info purposes only; just fill the column with whatever you like). The remaining entries are the symbols for each of the genes in that gene set.

The entries in the .gmt file must use the same coding as that of column 2 of your expression spreadsheet.

Further points:

- There are boxes where you can specify `minimum and maximum gene set sizes`. Gene sets outside of these ranges are ingored.
- When you run Gene Set Analysis, a message might appear, saying that there was too little overlap between your gene names and those in the .gmt file. This probably means that you have not used the same coding for both, or that you have the gene names in the expression sheet in the wrong column (they should be in column 2)
- When you run a Gene Set Analysis, the SAM plot looks different from the usual plot. Because the gene sets are usually of different sizes, the gene set scores cannot be directly compared. Hence we convert each score to a p-value, using separate permutation distributions for each gene set to estimate FDRs. The Gene Set Analysis plot shows the FDR for each p-value cutoff, both for positive and negative gene sets. The slider sets the FDR cutoff that defines upper and lower p-value cutoffs, and the resulting number of significant gene sets are shown in the top left part of the panel.
- A "Negative" gene set is one in which lower expression of most genes in the gene set correlates with higher values of the phenotype y. For example, two classes coded 1,2, lower expression correlates with class 2. For survival data, lower expression correlates with higher risk, i.e shorter survival (Be careful, this can be confusing!)
- A "Positive" gene set is one in which higher expression of most genes in the gene set correlates with higher values of the phenotype y.
- Under `Significant Gene Set` tab, you get a list of positive and negative sets, and you can also type the number of each gene set to see the individual genes and their scores.

- `Gene set collection` tab gives general info about the overlap between your list of genes and the gene set collection.
- 100 or 200 permutations are OK for initial exploratory analysis, but to get accurate estimates of FDR, we recommend *at least 1000 permutations*.
- Gene set analysis is only available for response types `Two class unpaired`, `Two class paired`, `Survival`, `Multiclass` and `Quantitative`.

12 Technical details of the SAM procedure

The data is x_{ij}, $i = 1, 2, \ldots p$ genes, $j = 1, 2, \ldots n$ samples, and response data y_j, $j = 1, 2, \ldots n$ (y_j may be a vector).

Here is the generic SAM procedure for array data. For sequencing data, the definition of the score d_i is different- see Section 12.5.

1. Compute a statistic

$$d_i = \frac{r_i}{s_i + s_0};\ i = 1, 2, \ldots p \tag{12.1}$$

   r_i is a score, s_i is a standard deviation, and s_0 is an exchangeability factor. Details of these quantities are given later in this note.

2. Compute order statistics $d_{(1)} \leq d_{(2)} \cdots \leq d_{(p)}$
3. Take B sets of permutations of the response values y_j. For each permutation b compute statistics d_i^{*b} and corresponding order statistics $d_{(1)}^{*b} \leq d_{(2)}^{*b} \cdots \leq d_{(p)}^{*b}$.
4. From the set of B permutations, estimate the expected order statistics by $\bar{d}_{(i)} = (1/B) \sum_b d_{(i)}^{*b}$ for $i = 1, 2, \ldots p$.
5. Plot the $d_{(i)}$ values versus the $\bar{d}_{(i)}$.
6. For a fixed threshold Δ, starting at the origin, and moving up to the right find the first $i = i_1$ such that $d_{(i)} - \bar{d}_{(i)} > \Delta$. All genes past i_1 are called "significant positive". Similarly, start at origin, move down to the left and find the first $i = i_2$ such that $\bar{d}_{(i)} - d_{(i)} > \Delta$. All genes past i_2 are called "significant negative". For each Δ define the upper cut-point $\mathrm{cut}_{up}(\Delta)$ as the smallest d_i among the significant positive genes, and similarly define the lower cut-point $\mathrm{cut}_{low}(\Delta)$.

7. For a grid of Δ values, compute the total number of significant genes (from the previous step), and the median number of falsely called genes, by computing the median number of values among each of the B sets of $d^{*b}_{(i)}$, $i = 1, 2, \ldots p$, that fall above $\mathrm{cut}_{up}(\Delta)$ or below $\mathrm{cut}_{low}(\Delta)$. Similarly for the 90th percentile of falsely called genes.

8. Estimate π_0, the proportion of true null (unaffected) genes in the data set, as follows:

   (a) Compute $q25, q75 = 25\%$ and 75% points of the permuted d values (if $p = \#$ genes, $B = \#$ permutations, there are pB such d values).

   (b) Compute $\hat{\pi}_0 = \#\{d_i \in (q25, q75)\}/(.5p)$ (the d_i are the values for the original dataset: there are p such values.)

   (c) Let $\hat{\pi}_0 = \min(\hat{\pi}_0, 1)$ (i.e., truncate at 1). This estimate of π_0 is analogous to setting $\lambda = 0.5$ in the $\hat{\pi}_0$ proposed in [6]. For *multiclass* data, the scores are all positive, so we use the 0th and 50th percentiles of the permuted values [NOTE: this was corrected in version 2.0].

9. The median and 90th percentile of the number of falsely called genes from step 6, are multiplied by $\hat{\pi}_0$.

10. User then picks a Δ and the significant genes are listed.

11. The False Discovery Rate (FDR) is computed as [median (or 90th percentile) of the number of falsely called genes] divided by [the number of genes called significant].

12. **Fold change**. Suppose $\bar{x}_{i1}$ and $\bar{x}_{i2}$ are the average expression levels of a gene i under each of two conditions. These averages refer to raw (unlogged) data. Then if a nonzero fold change t is also specified, then a positive gene must also satisfy $|\bar{x}_{i2}/\bar{x}_{i1}| \geq t$ in order to be called significant and a negative gene must also satisfy $|\bar{x}_{i1}/\bar{x}_{i2}| \leq 1/t$ to be called significant. When a fold change is specified, genes with either $\bar{x}_{i1} \leq 0$ or $\bar{x}_{i2} \leq 0$ (or both) are automatically left off the significant gene list, as their fold change cannot be unambiguously determined. When such fold changes are reported in output, they are indicated by `NA`.

13. **The q-value** of a gene is the false discovery rate for the gene list that includes that gene and all genes that are more significant. It is computed by finding the smallest value of $\hat{\Delta}$ for which the gene is called significant, and then is the FDR corresponding to $\hat{\Delta}$.

14. The **local FDR** for a gene is the false discovery rate for genes having a similar score d_i as that gene. It is estimated by taking a symmetric window of 0.5% of the genes on each side of the target gene, and estimating the FDR in that window. If 1.0% times the total number of genes in the dataset is less than 50, then the percentage is increased so that the number of genes is 50.

12.1 Computation of s_0

1. Let s^α be the α percentile of the s_i values. Let $d_i^\alpha = r_i/(s_i + s^\alpha)$.
2. Compute the 100 quantiles of the s_i values, denoted by $q_1 < q_2 \ldots < q_{100}$.
3. For $\alpha \in (0, .05, .10 \ldots 1.0)$
   (a) Compute $v_j = \mathrm{mad}(d_i^\alpha | s_i \in [q_j, q_{j+1}))$, $j = 1, 2, \ldots n$, where mad is the median absolute deviation from the median, divided by .64
   (b) Compute $\mathrm{cv}(\alpha)$= coefficient of variation of the v_j values
4. Choose $\hat{\alpha} = \mathrm{argmin}[\mathrm{cv}(\alpha)]$. Finally compute $\hat{s}_0 = s^{\hat{\alpha}}$. s_0 is henceforth fixed at the value $\hat{s}_0$.

For *Wilcoxon option*, *rank regression* and *pattern discovery*, the s_0 percentile is set at 5%. We found that this offered better performance than automatic estimation of s_0 in these cases.

12.2 Details of r_i and s_i for different response types.

Quantitative response r_i is the linear regression coefficient of gene i on the outcome:

$$r_i = \frac{\sum_j y_j(x_{ij} - \bar{x}_i)}{\sum_j (y_j - \bar{y}_j)^2} \tag{12.2}$$

where $\bar{x}_i = \sum_j x_{ij}/n$ and s_i is the standard error of r_i:

$$s_i = \frac{\hat{\sigma}_i}{[\sum_j (y_j - \bar{y}_i)^2]^{1/2}}, \tag{12.3}$$

and $\hat{\sigma}_i$ is the square root of residual error:

$$\begin{aligned} \hat{\sigma}_i &= \Big[\frac{\sum_j (x_{ij} - \hat{x}_{ij})^2}{n-2}\Big]^{1/2} \\ \hat{x}_{ij} &= \hat{\beta}_{i0} + r_i y_j \\ \hat{\beta}_{i0} &= \bar{x}_j - r_i \bar{y}_j \end{aligned} \tag{12.4}$$

If *rank regression is selected*, y_i and each gene x_{ij} are first converted to ranks.

Two class, unpaired data $y_j = 1$ or 2. Let $C_k = \{j : y_j = k\}$ for $k = 1, 2$. Let $n_k = \#$ of observations in C_k. Let $\bar{x}_{i1} = \sum_{j \in C_1} x_{ij}/n_1, \bar{x}_{i2} = \sum_{j \in C_2} x_{ij}/n_2$.

$$r_i = \bar{x}_{i2} - \bar{x}_{i1}$$

$$s_i = [(1/n_1 + 1/n_2)\{\sum_{j\in C_1}(x_{ij} - \bar{x}_{i1})^2 + \sum_{j\in C_2}(x_{ij} - \bar{x}_{i2})^2\}/(n_1 + n_2 - 2)]^{1/2}$$

If instead the *Wilcoxon* statistic is selected, The Mann-Whitney (two sample Wilcoxon) statistic is computed.

NOTE: this was changed in version 2.0; in previous versions we used the regression of the outcome on gene i. The current version is more consistent with the treatment of other data types.

Censored survival data $y_j = (t_j, \Delta_j)$. t_j is time, $\Delta_j = 1$ if observation is a death, 0 if censored. Let D be the indices of the K unique death times $z_1, z_2, \ldots z_K$. Let $R_1, R_2, \ldots R_K$ be the indices of the observations at risk at these unique death times, that is $R_k = \{i : t_i \geq z_k\}$). Let $m_k = \#$ in R_k. Let d_k be the number of deaths at time z_k and $x^*_{ik} = \sum_{t_j=z_k} x_{ij}$ and $\bar{x}_{ik} = \sum_{j\in R_k} x_{ij}/m_k$.

$$\begin{aligned} r_i &= \sum_{k=1}^{K}[x^*_{ik} - d_k\bar{x}_{ik}] \\ s_i &= [\sum_{k=1}^{K}(d_k/m_k)\sum_{j\in R_k}(x_{ij} - \bar{x}_{ik})^2]^{1/2} \end{aligned} \tag{12.5}$$

NOTE: A *positive score* (red genes in the SAM plot) means that *higher expression correlates with higher risk, i.e. shorter survival*. The reverse is true for negative scores (green genes): a negative score means higher expression correlates with lower risk, i.e. longer survival.

[We had this wrong in some earlier versions of this manual]!

Multiclass response $y_j \in \{1, 2, \ldots K\}$. Let C_k= indices of observations in class k, $n_k = \#$ in C_k, $\bar{x}_{ik} = \sum_{j\in C_k} x_{ij}/n_k$, $\bar{x}_i = \sum_j x_{ij}/n$.

$$r_i = [\{\sum n_k/\prod n_k\}\sum_{k=1}^{K} n_k(\bar{x}_{ik} - \bar{x}_i)^2]^{1/2} \tag{12.6}$$

$$s_i = [\frac{1}{\sum(n_k - 1)} \cdot (\sum \frac{1}{n_k})\sum_{k=1}^{K}\sum_{j\in C_k}(x_{ij} - \bar{x}_{ik})^2]^{1/2} \tag{12.7}$$

$$\tag{12.8}$$

Paired data $y_j \in \{-1, 1, -2, 2 \ldots -K, K\}$. Observation $-k$ is paired with observation k. Let

$j(d)$ be index of the observation having $y_j = d$.

$$z_{ik} = x_{ij(k)} - x_{ij(-k)} \tag{12.9}$$

$$r_i = \sum_k z_{ik}/K \tag{12.10}$$

$$s_i = [\sum_k (z_{ik} - r_i)^2/\{K(K-1)\}]^{1/2} \tag{12.11}$$

One class data $y_j = 1 \forall j$.

$$r_i = \bar{x}_i = \sum_j x_{ij}/n$$

$$s_i = \{\sum_j (x_{ij} - \bar{x}_i)^2/(n(n-1))\}^{1/2} \tag{12.12}$$

12.3 Details of Permutation Schemes

For *unpaired*, *quantitative*, *multiclass* and *survival* data we do simple permutations of the n values y_j. For *paired data*, random exchanges are performed within each $-k, k$ pair. For *one class* data, the set of the expression values for each experiment are multiplied by $+1$ or -1, with equal probability. If blocks are specified, the permutations are restricted to be within blocks, as described earlier. For *pattern discovery*, the elements within each row (gene) are permuted separately. This gives a new data matrix, whose eigenvectors are then computed.

12.4 Assessment of sample sizes

Assessment of sample sizes for microarray data is a tricky exercise. What assumptions should one make, and what quantities should be provided as output?

Some packages (e.g. the R package `ssize`) assume that the genes are independent and use the Bonferroni inequality to set the type I error. Since genes in microarray experiments are far from independent, this approach seems to be too conservative. They also report the power for each gene. But how does one interpret this in the context of thousands of genes.

In our approach we start with the output from a SAM analysis for a set of pilot data. From this we estimate the standard deviation of each gene, and the overall null distribution of the genes. Then for a given hypothesized mean difference, we estimate the false discovery rate (FDR) and false negative rate (FNR) of a list of genes. Since the calculation is based on the SAM scores from permutations of the data, the correlation in the genes is accounted for. By working with the scores rather than the raw data, we avoid the difficult task of simulating new data from a population having a complicated (and unknown) correlation structure.Table 6 summarizes the outcomes of p hypothesis tests of a set of p genes.

Table 6: *Possible outcomes from p hypothesis tests of a set of genes. The rows represent the true state of the population and the columns are the result from a data-based decision rule.*

	Called Not Significant	Called Significant	Total
Null	U	V	p_0
Non-null	T	S	p_1
Total	$p-R$	R	p

Now $\mathrm{FDR} = V/R$ and $\mathrm{FNR} = T/(p-R)$, $\mathrm{power} = S/p_1$ and type I error $= V/p_0$. For simplicity, we assume that the number of genes called significant (R) is the same as the number of non-null genes in the population (p_1). This implies that $1 - \mathrm{power} = \mathrm{FDR}$ and type I error=FNR. Hence conveniently, the FDR can be interpreted as one minus the per gene power, and similarly for the FNR.

Here are the details of the calculation for the two-class unpaired case. (Below we indicate changes necessary for other data types). If n_1 and n_2 are the sample sizes in each group, The SAM score is

$$d_i = \frac{\bar{x}_{i2} - \bar{x}_{i1}}{s_i}$$

where

$$s_i = [(1/n_1 + 1/n_2)\{\sum_{j \in C_1}(x_{ij} - \bar{x}_{i1})^2 + \sum_{j \in C_2}(x_{ij} - \bar{x}_{i2})^2\}/(n_1 + n_2 - 2)]^{1/2}$$

If non-zero, the exchangeability constant s_0 is also included in the denominator (i.e the denominator is $s_i + s_0$.) If σ_i is the within-group standard deviation for gene i (assumed to be the same in each group), then ${s_i}^2$ estimates

$$\mathrm{var}(\bar{x}_{i2} - \bar{x}_{i1}) = \sigma_i^2(1/n_1 + 1/n_2)$$

(we assume that the proportion of samples in groups 1 and 2 remains the same as we vary the sample size). Hence a shift of δ units in one gene for each sample in group 2 causes an average increase in the SAM score d_i of $\delta/(\sigma_i\sqrt{1/n_1 + 1/n_2})$. Here is the calculation in detail:

1. Estimate the null distribution of the SAM scores, and the per gene standard deviation σ_i. from the set of SAM permutations.

2. For k (the number of truly changed genes) running from 10 to $p/2$, do the following:

   - Sample a set of p scores from the permutation distribution of the scores

- Add $\delta/(\sigma_i\sqrt{1/n_1 + 1/n_2})$ in class 2 to a randomly chosen set of k of these scores.
- Find the cutpoint c equal to the k largest score in absolute value
- Estimate the FDR and FNR of the rule $|d_i| > c$. This is straightforward since we know which genes are truly non-null (they are the ones that were incremented by δ)

3. Repeat Step 2 twenty times and report the median result for each k.

SAM does the above calculation for sample sizes $n, 2n, 3n$ and $5n$ (assuming the input sample size factors are $1, 2, 3, 5$) and reports the results both graphically and in tables on the SAM output sheet. This gives the user information on how the FDR and FNR will improve if the sample size were to be increased.

The user specifies the **hypothesized mean difference** δ and **sample size factors** s_1, s_2, s_3, s_4 (default 1,2,3,5). SAM then tries sample sizes $s_1 n, s_2 n, s_3 n, s_4 n$

To get an idea of what values of the mean difference δ are appropriate or reasonable, the user can look at the significant gene list from the SAM analysis. The `Numerator` column is the mean difference for each gene.

In SAM version 2.1, sample size assessment is offered only for unpaired, paired, oneclass and survival data types. For paired data, we take $n_1 = n_2 = n/2$ (remember n is the total sample size). and all of the above recipe is the same. For one class data, $\text{var} = \sigma_i^2/n$. For survival data with r_i equal to the numerator of the Cox score statistic, we assume that $\text{var} r_i = \sigma_i^2/n$ and we interpret δ relative to r_i. That is for example, if in our pilot data the genes that we call significant have $|r_i| > 100$ (roughly), we might set $\delta = 100$ in our sample size assessment.

Here is an example. We generated some two-class data: 1000 genes and 20 samples, 10 samples in each of classes. Each measurement was standard Gaussian (i.e. there was no difference between the groups in the pilot data). We ran SAM (two class unpaired, logged) and entered a mean difference of $\log_2 2 = 1.0$. Thus we are hypothesizing a difference of 2 fold for class 1 versus class 2, assuming that the data are on a log base 2 scale. The results are shown in Figure 5. Remember that the quantity on the horizontal axis— `number of genes`— refers to both the hypothesized number of truly non-null genes, and the number of genes called significant.

We see that, depending on the number of genes truly changed at 2-fold, the sample size should be increased to 60 or 100, in order to get the FDR down to 10 or 5%. The false negative rate is consistently low throughout.

12.5 Sequencing data

Data from RNA-seq experiments come in the form of counts for each gene or probe. They are non-negative and can be very skewed (some large values). In addition, the sequencing depth for each sample is typically different, creating bias in the counts for that sample. Hence one cannot simply apply methods designed for microarray data to RNA-seq data.

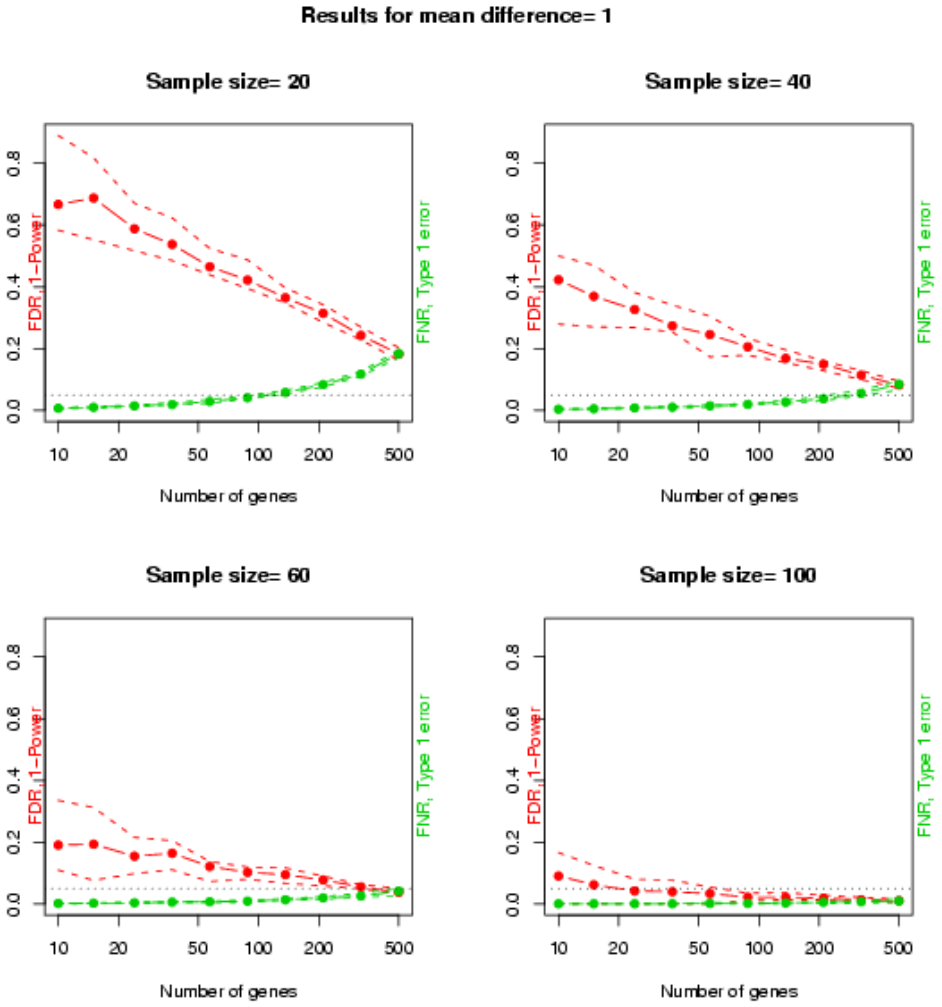

Figure 5: Sample size assessment plot

Some approaches to RNA-seq data use the poisson or negative binomial distributions to model the counts. While this is useful, we have found that at times it is not very robust or reliable [5]. Hence we have developed a non-parametric approach to this problem, that involves a) estimating the sequencing depths b) resampling from the data using these estimated depths, c) computing a non-parametric summary measure (such as the Mann-Whitney-Wilcoxon test) on each resampled dataset and d) averaging the summary measures over the resamples.

From a macroscopic point of view, this process simply replaces the SAM score d_i with a new score equal to the average summary measure. Then the rest of the SAM procedure, as outlined above, is the same. We call this procedure "SAMSeq". When the `sequencing` option is selected, SAM carries out this procedure and also outputs the estimating sequencing depths. Full details may be found in [5].

13 Frequently Asked Questions

1. SAM generates an error when I run it on my dataset. What should I do?

   Most often, errors are due to improper data formats.

   - Please make sure that your data is formatted exactly as described in section 5. Particular attention needs to be paid to the format of the response in the first row as described in section 5.1.
   - Please make sure that the response type you chose in the SAM panel shown in figure 1 matches the format of your response.

     In our testing, about 95% of the problems have been due to the wrong response format.
   - Is there a gene with only one or zero *non-missing* value? If so, the imputation will fail.

2. Why does the random number seed stay the same? Can you not generate a new seed automatically?

   The random number seed allows one to reproduce an analysis. By default, it is set to 1234567. However, if one uses the default seed for every analysis, then the *same sequence of permutations* are generated. This is not always desirable. It would appear that generating a seed randomly using the clock or some such mechanism without bothering the user for input might be better. Not necessarily. If reproducibility is important, then asking the user to set the seed is preferable so that any analysis can be rerun to confirm results. We have come down on the side of reproducibility. The user always has a choice of requesting a randomly generated seed based on the clock by clicking on the **Generate Random Seed** button. Please also note that the random number generator seed used in any analysis is always listed in the output to ensure reproducibility of results.

3. This document does not answer my questions. Where should I look?

   As we get asked new questions, we update this list of frequently asked questions with answers. Please visit the url `http://www-stat.stanford.edu/~tibs/SAM` where you may find further information.

4. Where can I go for help if I just cannot get SAM to work?

   We are very interested in making SAM work for all users. However, before reporting problems or bugs, we'd really like you to make sure that the problem is really with SAM. The following checklist should help.

   - If the problem is with SAM usage, please make sure that you have formatted your data exactly as mentioned in the SAM manual.
   - If you are having problem on a particular type of data, please make sure that you have formatted the response labels appropriately and have chosen the correct applicable data type.

   If you still cannot get SAM to work, send email to `sam-bug@stat.stanford.edu` with complete details including

   (a) The error message

   (b) The system you are using

   (c) The version of R you are using

   (d) The dataset you used that generated the error.

References

[1] B. Efron and R. Tibshirani. Microarrays, empirical bayes methods, and false discovery rates. *Genetic Epidemiology*, 1:70–86, 2002.

[2] B. Efron and R. Tibshirani. On testing the significance of sets of genes. *Annals of applied statistics*, 2007.

[3] B. Efron, R. Tibshirani, J. Storey, and V. Tusher. Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, pages 1151–1160, 2001.

[4] T. Hastie, O. Alter, G. Sherlock, M. Eisen, R. Tibshirani, D. Botstein, and P. Brown. Imputation of missing values in dna microarrays. Technical report, 1999. Working draft.

[5] Jun Li and Robert Tibshirani. Finding consistent patterns: a nonparametric approach for identifying differential expression in rna-seq data. *To appear: Statistical Methods in Medical Research*, 2011.

[6] J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B.*, 64:479–498, 2002.

[7] A. Subramanian, V. K. Tamayo, P. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102:15545–15550, 2005.

[8] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 16:520–525.

[9] V. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to transcriptional responses to ionizing radiation. *Proc. Natl. Acad. Sci. USA.*, 98:5116–5121, 2001.