# PaGE 5.1 Technical Manual

Grant, Gregory R.[*] Liu, Junmin [*]
Stoeckert, Christian J. Jr. [*]

September 2, 2004

## Contents

[*] Penn Center for Bioinformatics (PCBI), Computational Biology and Informatics Laboratory (CBIL), University of Pennsylvania

# 1 Introduction

This report gives the full details of the PaGE 5.1 algorithm. For the PaGE 5.1 perl implementation user guide please see this document.

`http://www.cbil.upenn.edu/PaGE/doc/perl/PaGE_5.1_documentation.html`

PaGE is a tool for analyzing microarray gene expression data. It can be used to find differentially expressed genes between two conditions, and to generate patterns across several conditions. PaGE was originally introduced by Manduchi et al. (1999), and though the algorithm has changed significantly, the general approach of generating discrete patterns using an FDR based confidence measure has remained unchanged.

# 2 Differential Expression

We assume that there are two well defined experimental conditions, and that each gene has a measurable expression intensity that follows some (unknown) distribution in each condition. Differential expression of a gene means that these distributions are different between the two conditions. The distributions can differ in every possible way, but the statistics we use are designed to exploit primarily a difference in the means (e.g. the $t$-statistic). Even so, the hypotheses being tested are of equality of distributions. This is a necessary consequence of using the permutation methods that we do.

The data are assumed to consist of multiple quantified microarray experiments in each condition. What can be concluded from any differential expression analysis depends on how well the data represent random samples from the conditions of interest.

We begin by considering just two experimental conditions, called condition 0 and condition 1. Condition 0 will be referred to as the *reference* condition. Up-regulation of a gene will mean the gene's mean intensity is higher in condition 1 as compared to condition 0, and down-regulation will mean the gene's mean intensity is lower in condition 1 versus condition 0.

# 3 Study Design

There are three ways to compare two conditions using microarrays. The most straightforward is to use 1-channel[1] arrays and to hybridize some number of replicate arrays for each condition. This gives a number of intensities for each gene, in each condition.

A second strategy, known as the *reference design*, uses 2-channel data, where the experimental conditions are hybridized in one channel and a common reference is hybridized to the other channel. This common reference is identical for all arrays. In this case the data consist of ratios (or log ratios), where the

[1] We consider Affymetrix arrays as 1-channel

denominator is always the gene's intensity in the reference sample, and the numerator is one or the other experimental condition. In all of the above cases we shall refer to the design as a 2-sample design, because it produces two sets of intensities for each row, corresponding to the two conditions.

The third possibility is the *direct comparison* design. 2-channel arrays are used, with condition 0 being hybridized to one channel and condition 1 hybridized to the other. Since we do not separate the channels, but instead work directly with the ratios, this requires an analysis which is somewhat different from the 1-channel or reference design.

A 2-sample design can also be "paired". Suppose for example that two anatomical regions are compared in each of $m$ animals. It might be that the expression of gene $G$ in the first region is always twice as high as the expression in the second, but the exact values depend strongly on the animal. In this case it is often better to run a paired analysis. This approach considers the data as ratios of paired experiments (or differences if the data are log transformed), analagous to the direct comparison design in which the experiments are naturally paired. Therefore this can increase the power of the results if there is a strong effect in the data. If there is not a strong paired effect, then it will likely decrease the power to run a paired analysis. These cases are discussed in detail in Section 10.

# 4 The Data

Suppose first that we have 1-channel data, or 2-channel reference design data. We essentially forget, for the moment, that we are dealing with ratios in the case of a reference design, and proceed with the same analysis for both cases. In the case of Affymetrix data we assume the probe set intensities have been turned into summary values by some method.[2] So the data in these cases consist of some number $m$ of replicates arrays in the group 0 and some number $n$ of replicates in group 1. Suppose there are $g$ rows of data. We put the data in a matrix as in (1) below.

$$\begin{array}{c|cccc|cccc}
 & C_1 & C_2 & \cdots & C_m & D_1 & D_2 & \cdots & D_n \\
\hline
G_1 & c_{11} & c_{12} & \cdots & c_{1m} & d_{11} & d_{12} & \cdots & d_{1n} \\
G_2 & c_{21} & c_{22} & \cdots & c_{2m} & d_{21} & d_{22} & \cdots & d_{2n} \\
G_3 & c_{31} & c_{32} & \cdots & c_{3m} & d_{31} & d_{32} & \cdots & d_{3n} \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
G_g & c_{g1} & c_{g2} & \cdots & c_{gm} & d_{g1} & d_{g2} & \cdots & d_{gn}
\end{array} \tag{1}$$

The columns in (1) correspond to arrays and rows correspond to genes. The columns labelled with the $C_i$'s correspond to arrays from condition 0, and the columns labelled with the $D_i$'s correspond to the arrays from condition 1.

We denote a row of the data matrix by $r$. If $p$ is a permutation of the columns (see next section), we denote the correspondingly permuted row by $r_p$.

[2] e.g. MAS 5.0 (Affymetrix (2003)), RMA (Irizarry *et al.* (2003)), Probe Profiler (Corimbia Inc.), etc.

# 5 Permutations

In the two-sample case where the data are in the form of the data matrix (1), a permutation is any rearrangement of the $m + n$ columns. This can also be thought of as choosing some number $k$ of columns from group 0, and then choosing the same number of columns from group 1, and switching them. We then obtain two new groups of columns from the first $m$ and the last $n$ columns of the permuted data matrix. What is important in a permutation is which columns end up in which group, and not the order that they happen to be listed in left to right. Therefore there are a total of $\binom{m+n}{n}$ possible permutations in the two-sample case.

In the case of a direct comparison design, where each row of data consists of a set of $n$ ratios, a permutation consists of taking $k$ of the columns, for some $k$, and and taking the reciprocals of those $k$ elements in each row.[3] The set of all permutations consists of doing this for all possible subsets of $k$ columns, for all $k \leq n$. So for example if a row is $(x_1, x_2, x_3, x_4, x_5)$ and $k = 3$ with columns 1, 2, and 4 being changed, then the permuted row becomes $(1/x_1, 1/x_2, x_3, 1/x_4, x_5)$.

There are as many distinct permutations as there are subsets of the $n$ columns, therefore there are $2^n$ permutations in the direct comparison case.

# 6 Missing Values

PaGE does not impute missing values, instead it leaves them missing. The reason for this is that imputed missing values can be fairly unreliable when there are only a few replicates, while when there are many replicates a few missing values do not impact greatly on the results. Thus there is no great need to try to impute.

PaGE will consider a row with missing values as having fewer replicates for that row. Of course if there is less than two values in any condition then that row must be ignored. The same is true when considering permutations of the data, a permutation that leads to a group having less than two replicates is ignored in the permutation distributions.

The program's behavior with respect to missing values can be controlled by the user through the "min presence" parameter. This allows the user to choose the minimum number of values that must be present in the condition in order to not be ignored. The parameter can be set separately for each group, or one global setting can be used for all groups. The default value is two replicates per group.

# 7 The Statistics

We denote by $S$ any two-class statistic, and think of it as a function which maps rows of the data matrix $r$ to real numbers. Suppose there is some center point $c$ such that $S > c$ indicates up-regulation and $S < c$ indicates down-regulation. The statistics we have in mind are:

[3] If the data are log transformed, then take the negative instead of the reciprocal.

- The modified $t$-statistic

$$((c_1, c_2, \ldots, c_m), (d_1, d_2, \ldots, d_n)) \longrightarrow \frac{\mu_1 - \mu_0}{\alpha + D} \qquad (2)$$

where $\mu_0$ is the mean of $(c_1, c_2, \ldots, c_m)$, $\mu_1$ is the mean of $(d_1, d_2, \ldots, d_n)$,

$$D = \sqrt{\frac{\sigma_0^2(m-1) + \sigma_1^2(n-1)}{m+n-2}},$$

where $\sigma_0^2$ is the sample variance of $(c_1, c_2, \ldots, c_m)$

$$\frac{1}{m-1}\sum_{j=1}^{m}(c_j - \mu_0)^2,$$

and $\sigma_1^2$ is the sample variance of $(d_1, d_2, \ldots, d_n)$

$$\frac{1}{n-1}\sum_{j=1}^{n}(d_j - \mu_1)^2.$$

In this case the center $c$ equals 0. When $\alpha = 0$ this is the standard two-sample $t$-statistic. As we will see later, results can be extremely sensitive to the value of $\alpha$ when comparing $t$-statistics across many genes, particularly when there are relatively few replicates. Therefore we refer to $\alpha$ as the *$t$-statistic tuning parameter*.[4] The effect of this parameter is discussed in detail in Section 14 below.

- The second statistic is the ratio of the means in the two conditions

$$((c_1, c_2, \ldots, c_m), (d_1, d_2, \ldots, d_n)) \longrightarrow \frac{\mu_0}{\mu_1}$$

where $\mu_0$ is the mean of $(c_1, c_2, \ldots, c_m)$ and $\mu_1$ is the mean of $(d_1, d_2, \ldots, d_n)$. In this case the center $c = 1$.

- If $n = m$ and the data are paired, then the geometric mean of the paired ratios

$$((c_1, c_2, \ldots, c_n), (d_1, d_2, \ldots, d_n)) \longrightarrow \sqrt[n]{\frac{c_1}{d_1}\frac{c_2}{d_2}\cdots\frac{c_n}{d_n}}$$

The center $c = 1$.

These don't all make sense in all cases. For example if there are negative intensities in the unlogged data then the ratio statistics is not very sensible. If $m \neq n$ then the geometric mean of the paired ratios is not well defined.

We could apply the $t$-statistic to the unlogged, or to the logged, data, or to any transformation of the data. Therefore, even with fixed $\alpha$, the $t$-statistic

[4] This is the same as the so-called $t$-statistic "fudge factor" introduced by Tusher *et al.* (2001).

is not one statistic but a family of statistics. This will be discussed further in Section 15.

et $M$ be the data matrix, and let $M_p$ be the data matrix whose rows have been permuted by the permutation $p$. Let $g$ be the number of rows of $M$. We think of the statistic $S$ as a mapping which takes $M$, or $M_p$, to a vector of real numbers of length $g$. We denote this mapping also by $S$. We denote the value of $S$ on row $r$ by $S_r$.

# 8 The FDR

We will focus on upregulation in condition 1 versus condition 0. The case of downregulation follows by switching the roles of conditions 0 and 1. We will also assume that larger values of $S$ are more significant. This is the case for all of the statistics above. The case where smaller values are more significant follows by switching the direction of the inequalities.

For any row $r$ we take the null hypothesis $H_r^0$ to be that the distribution for row $r$ in condition 0 is identical to the distribution for row $r$ in condition 1. Suppose that for $g_0$ of the rows the null hypothesis is true. Let $g_1 = g - g_0$. For each real number $k > c$, let $\mathcal{G}_k$ be the set of rows $r$ of $M$ such that $S_r \geq k$. $\mathcal{G}_k$ is the set of predictions if we use $k$ as the "cutoff" for the statistic. Let $R_k$ be the size of $\mathcal{G}_k$. Let $V_k$ be the number of rows in $\mathcal{G}_k$ for which the null hypothesis is true.

With this set-up, we will have made $V_k$ false predictions out of $R_k$ total predictions. Provided $R_k > 0$, we call the ratio $V_k/R_k$ the *false discovery proportion* of this set of predictions.

Our approach towards differential expression analysis of microarrays is to control this proportion in some way. This is the common approach to the multiple testing problem in microarray differential expression analysis, so we will not argue its merits here. There are many ways to define what it means to control this proportion, and our FDR definition differs from the original one of Benjamini and Hochberg (1995), as well as that of Storey (2002) and Storey and Tibshirani (2003). We define the *false discovery rate* (FDR) of the procedure itself, as

$$\begin{cases} E(V_k)/R_k, & R_k > 0 \\ 0, & R_k = 0. \end{cases} \tag{3}$$

In contrast the original definition of Benjamini and Hochberg is

$$\begin{cases} E(V_k/R_k), & R_k > 0 \\ 0, & R_k = 0. \end{cases} \tag{4}$$

The advantage of (4) is that it takes into account the correlation between $V_k$ and $R_k$. However the advantage of (3) is that it can be more realistically estimated via permutation distributions. [5]

[5] Indeed to estimate (4) one needs to know something about the random properties of $V/R$. If we permute the columns of the data matrix (1), we can obtain some kind of approximation to an observation of $V$ under the complete null hypothesis, but this tells us nothing about $V/R$ under the true distribution of the data. The bootstrap distribution obtained by sampling with

The goal is to find a value of $k$ so that (3) is acceptablly low. Sometimes one is willing to tolerate a relatively high FDR such as .5, other times a low FDR such as .05 is desired. PaGE searches for the least conservative (i.e. smallest) value of the cutoff $k$ for which this FDR can be achieved, by estimating the FDR over the range of values of $k$ and choosing the smallest $k$ which achieves the desired FDR. The range of values of $k$ goes from the center $c$ to the observed maximum of the statistic over the unpermuted data. PaGE, by default, divides this range into 1,000 equally spaced values of $k$, called "bins".

# 9 FDR Estimation

For each permutation $p$ of the data matrix we obtain a value $V_k^p$ which equals the number of rows whose permutation statistic $S_{r_p} \geq k$. Thus we obtain a permutation distribution $D_k$ of $V_k$ under the complete null hypothesis (that is, when all null hypotheses are true). Note that the distribution $D_k$ depends on the joint distribution of the $S_r$, $1 \leq r \leq g$, which is maintained by permuting the data matrix in columns.

Let $\tilde{\mu}_k$ be the mean of $D_k$. There are two problems in using $\tilde{\mu}_k$ as an estimate of $E(V_k)$. First off, since it is calculated under the complete null hypothesis, it is at best a measure of how many hypotheses would be falsely rejected if they were all true. So, assuming that some hypotheses are false, $\tilde{\mu}_k$ would be an overestimate. Second, unless all hypotheses are true, the false hypotheses can cause the distribution of $V_k$ to be different from what it would be if we could consider only the true hypotheses in defining $D_k$. Since we do not know which hypotheses are true; we must allow the false hypotheses contribute to the counts involved in $D_k$. Typically permutation distributions are used to derive $p$-values, for which there is substantial theory, however, here we are interested in actually estimating $E(V_k)$ from the permutation distribution, and this requires some justification.[6]

Regarding the second issue, we argue that use of $\tilde{\mu}_k$ is conservative. Suppose that null hypotheses $r$ is false. Then for permutations $p$ which switch only one or a few columns between conditions, the false hypothesis $r$ will tend to have large values of the statistic $S_r$, and will therefore tend to contribute to the count of $V_k^p$ more than it would if $H_r^0$ were true. Similarly, downregulated genes will tend to overcontribute to the count $V_k^p$ for those permutations which switch most or all of the columns between the groups. Therefore the estimate $\tilde{\mu}_k$ will tend to be larger than the true value of $E(V_k)$. The more hypotheses that are false, the more conservative $\tilde{\mu}_k$ will be.

Turning to the first issue, we assume that $\tilde{\mu}_k$ is an overestimate of $E(V_k)$, as described above. Therefore

$$R_k - \tilde{\mu}_k \tag{5}$$

replacement from the two groups separately, will give us an approximation to the distribution of $R$, but again tells us nothing about $V/R$. To obtain information about $V/R$ most authors have had to make strong assumptions about the data.

[6] Note that similar "permutation estimates" are utilized in the SAM theory of Storey and Tibshirani (2003), however without justification as to why we should expect them to be reasonable estimates.

is an underestimate of the number of true positives (the rows in $\mathcal{G}_k$ for which the null hypotheses is false). Therefore $g - (R_k - \tilde{\mu}_k)$ is an overestimate of the total number of true hypotheses. Originally $\tilde{\mu}_k$ was calculated as an estimate to $V_k$ assuming all hypotheses are true. If we recalculate assuming there are $g - (R_k - \tilde{\mu}_k)$ true hypotheses, then we obtain

$$\tilde{\mu}_k(1) = \frac{\tilde{\mu}_k}{g} \left( g - (R_k - \tilde{\mu}_k) \right).$$

Since $g - (R_k - \tilde{\mu}_k)$ is an overestimate of the number of true hypotheses, $\tilde{\mu}_k(1)$ is still an overestimate of $E(V_k)$, however it is a better estimate than $\tilde{\mu}_k$. Using the same logic, we calculate

$$\tilde{\mu}_k(2) = \frac{\tilde{\mu}_k(1)}{g} \left( g - (R_k - \tilde{\mu}_k(1)) \right),$$

and in general

$$\tilde{\mu}_k(i+1) = \frac{\tilde{\mu}_k(i)}{g} \left( g - (R_k - \tilde{\mu}_k(i)) \right).$$

This sequence quickly converges, and PaGE takes $\tilde{\mu}_k(6)$ as its final estimate for $V_k$, which we denote by $\tilde{V}_k$.

We take as estimate of the FDR

$$\mathrm{FDR}_k = \tilde{V}_k / R_k.$$

It is useful to also define the quantity $\mathrm{CONF}_k = 1 - \mathrm{FDR}_k$. $\mathrm{CONF}_k$ is as estimate of the confidence the probability that any gene taken at random from $\mathcal{G}_k$ is a true positive.

We assign confidences not just to the $\mathcal{G}_k$, but to the rows of the data matrix themselves, by

$$\mathrm{CONF}_r = \min_{\substack{k \text{ such} \\ \text{that } r \in \mathcal{G}_k}} \mathrm{CONF}_k,$$

where $r$ is a row of the data matrix. In this way, we have confidence of at least $\gamma$ that a row $r$ with $\mathrm{CONF}_r = \gamma$ represents a truly differentially expressed gene.

# 10 The case of direct comparison and paired designs

In the case of direct comparison data, which consist of a data matrix with $n$ columns of ratios (possibly log ratios), the statistics are either the one sample $t$-statistic or the geometric mean of the ratios. The one sample $t$-statistic is applied to the logarithm of the data. If row $r$ is given by ratios $(b_1, b_2, \ldots, b_n)$, let $x_i = \log(b_i)$.[7] The statistic is then given by

$$(x_1, x_2, \ldots, x_n) \longrightarrow \frac{\mu}{\alpha + \sigma} \tag{6}$$

[7] In the PaGE 5.0 code natural log is used.

where $\mu$ is the mean of $(x_1, x_2, \ldots, x_n)$, and $\sigma^2$ is the sample variance

$$\frac{1}{n-1}\sum_{j=1}^{n}(x_j-\mu)^2.$$

As with the two-sample $t$-statistic, the tuning parameter $\alpha$ is added to the denominator. This parameter will be discussed further in Section 14 below.

In the case of paired designs, $m = n$. If the data are logged then paired differences are formed $b_i = d_i - c_i$ using the notation of (1). The data are now reduced to the case of direct comparison data. If the data are not logged then paired ratios are formed $b_i = d_i/c_i$.

Using the statistics described above and the one-sample permutations described in Section 5, the theory is identical to that described in Section 9, making the appropriate changes of notation.

## 11 The special case of unlogged negative intensities

In some cases data can contain negative intensities even before log transformation. For example with Affymetrix data the MAS 4.0 or earlier algorithms could produce negative values, as well the Probe Profiler algorithm (Corimbia Inc.) produces negative values. For some analysis it does not matter, for example when applying the $t$-statistic to the unlogged data there is no issue with there being negative intensities. However, in any case where it does matter (for example when using one of the ratio statistics), PaGE will alert the user that they are trying to apply a method that does not make sense when there are negative intensities. The analysis can continue if the user is willing to shift all intensities by some positive quantity so that the resulting data no longer contain negatives. When PaGE alerts the user it will suggest a moderate shift to perform in order to continue. Note that the results can depend on this transformation, too large a shift can decrease power dramatically, so shifting should be done with some attention to this fact. But in some cases there is no other way to perform an analysis, so this option can be useful.

## 12 Levels

Now that each gene has been assigned a confidence, they are next assigned “levels”. The user chooses a confidence $\gamma$. A cutoff $C$ for the statistic is determined by setting it to the minimum $k$ for which $\mathrm{CONF}_k > \gamma$, if such a $C$ exists. The set of all rows $r$ such that $S_r > C$ then gives the least conservative set of predictions which achieve a confidence of at least $\gamma$. Depending on the data, there may not be any such value of $C$ that achieves confidence $\gamma$, in which case $C$ is set to be $\infty$.

Depending on whether the statistic is on an additive scale (such as the $t$ statistic), or on a multiplicative scale (such as the geometric mean), the levels are created differently. Assume first the additive case. In this case if the statistic

$S_r$ is less than $C$, then row $r$ is given level zero. If $C \leq S_r < 2C$, then it is given level one. If $2C \leq S_r < 3C$ then it is given level two. In general if

$$nC \leq S_r < (n+1)C$$

then it is given level $n$.

Higher levels are rows with higher confidence than lower levels, however, the confidence associated with each row still only refers to the confidence of being differentially expressed, and not the confidence of being in any particular level. Levels are used for display purposes, particularly to generate patterns across several conditions, as discussed below.

If the statistic is on a multiplicative scale, then row $r$ is given level $n$ if

$$C^n \leq S_r < C^{n+1}.$$

The parameter $\gamma$ is referred to as the *level confidence.* As one raises the level confidence, fewer levels are produced and the genes assigned to the levels have higher confidence. One can raise or lower the level confidence as desired to find the best granularity for their needs. One will usually want to adjust the level confidence to a moderate level, and what that level is will depend on the dataset. After the gene confidences have been calculated, PaGE presents the user with a table which gives a summary breakdown of how many genes were found, but up- and down-regulated, for a range of confidences. The user can then choose one that suits their needs. See Section 15 for more on this.

## 13 Multiple conditions and patterns

PaGE can be used just as a differential expression analysis tool comparing two conditions. But another very useful feature of PaGE is its ability to compare multiple conditions simultaneously. For example one might have a time or developmental series, or a series of risk classes for tumor types, etc. Data often come in this multiclass form. We assume the conditions are labelled 0, 1, ..., $M$. PaGE compares each condition 1, 2, ..., $M$ to condition 0, generating patterns of length $M$ from the levels as described in the previous section. These patterns aid in locating general behavior of genes across the conditions. For example, if the conditions represent a time-series, then a pattern such as $(0, 1, 3, 3, 7)$ would indicate a gene whose expression was steadily rising. A pattern $(0, 0, -3, -3, -3)$ could be a gene which has just shut off in the third condition and remained off. Condition 0 is referred to as the *reference* condition. (Note: don't confuse the reference condition with the reference channel of a reference design study.) Position $i$ in the pattern represents a differential expression call between condition $i$ and condition 0. We have found that organizing the data into integer based patterns allows for convenient perusal of the results.

Level confidences can be set separately for each position, or one global level confidence can be used. As the level confidence is raised, fewer patterns are produced, as it is lowered more patterns are produced.

Examples of multi-class analyses are given in the HTML documentation.

## 14 The $t$-statistic tuning parameter

When using the $t$-statistic, the power of the method outlined above can be dramatically effected by the value of the tuning parameter $\alpha$ in (2) and (6). This is particularly true when there are only a few replicates per condition. The reason for this is that, since there are so many genes, there are many null genes whose variances in both groups are small just by chance. The $t$-statistics blow up for those null genes with vanishingly small variance, and since it is not those genes we want to find, and the algorithm is forced to be more conservative in its predictions to avoid picking them up. On the other hand, when $\alpha$ is too large, then for the non-null genes, the $t$-statistic's denominator dominates more for those genes with small mean difference and small variance, than those with large mean difference and large variance. The former set gets more and more lost in the noise as $\alpha$ goes up. As a result differentially expressed genes with small mean difference $\mu_1 - \mu_0$ and small variances in both groups tend to have larger confidence with smaller $\alpha$, while genes with a high mean difference and large variances have larger confidence with larger $\alpha$. Therefore, what value to set $\alpha$ to depends on the nature of the differentially expressed genes as well as the non-differentially expressed genes.

There is no obvious formula that can be applied to the data matrix to determine the value of $\alpha$ which maximizes the power. However, since the confidence is the same regardless of $\alpha$, a power criteria to determine $\alpha$ would be desirable. In other words we want the value of $\alpha$ which maximizes the power. Maximum power generally occurs for a moderate value of $\alpha$. See for example the data in Table 1, Table 2, and Table 3. PaGE therefore tries a range of values of $\alpha$, from very small to very large, and then chooses the one which gives the greatest number of results. This is the default value of $\alpha$. Other values of $\alpha$ can, however, find genes that the default value misses. Therefore it is important that the user have control over this parameter.

To illustrate this we generated several simulated datasets. First is a dataset with 5000 "genes", 300 of which are differentially expressed. Differentially expressed genes have varying mean differences. The first 25 rows had mean difference 1.2 between the two conditions. The next 25 rows had mean difference 1.3. The next had 1.4, etc. Specifically the 12 blocks of 25 has mean differences 1.2, 1.3, 1.4, 1.5, 2, 2.2, 2.4, 2.6, 4, 4.3, 4.6, 4.9, respectively. These are rows numbered 0-299. For each row, the intensities in each condition are given by beta distributions. The variances of those beta distributions increases over each block of 25 from very low to very high. So row 0 has $\mu_1 - \mu_0 = 1.2$ and very low variances, row 1 has $\mu_1 - \mu_0 = 1.2$ but slightly higher variances, up to rwo 24 which has $\mu_1 - \mu_0 = 1.2$ and high variances. Row 25 then has $\mu_1 - \mu_0 = 1.3$ with low variances, row 26 has $\mu_1 - \mu_0 = 1.3$ with slightly higher variances. Etc. The null genes were generated by 4700 randomly chosen beta distributions with randomly chosen parameters. The full dataset can be downloaded here:

```
http://www.cbil.upenn.edu/PaGE/doc/files/testdata_300diff_of_5000.txt
```

Using the first three columns of each condition we ran PaGE with 13 different values of $\alpha$ and also ran SAM. The web page `http://www.cbil.upenn.edu/PaGE/doc/files/3reps_conf.8_example.html`

has the results for the genes found at .8 confidence or higher. Columns represent different runs. The final column gives the SAM results. In the top rows the actual values of $R - V$ (the number of true positives), $R$ (the total number of predictions), and the confidence are given for each run. An "`X`" means that that gene was found in that run (the 300 differentially expressed genes are listed, numbered 0-299).

SAM produces results very close to the setting $\alpha = .1$. The power is maximized around $\alpha = 2.5$. At this value 166 genes are reported by PaGE, as opposed to SAM's 145. But the overlap between the two sets is only 128 genes. SAM finds 17 that PaGE does not and PaGE finds 38 that SAM does not.

It can also be seen from this example that lower values of $\alpha$ pick up the genes with low fold change and low variance, while higher values picked up the genes with high fold change and high variance.

If the non-differentially expressed genes tend to be bimodal, with small variance in each mode, then with a small $\alpha$ the null genes will add a lot of noise to the system. At the extreme, if a gene is bimodal with values 1 or 2, each with 50% probability, sampling 3 replicates per condition, and we choose $\alpha = 0$, then one time in 32 the $t$-statistic will be $\infty$. Raising $\alpha$ decreases these extreme values of the $t$-statistic so that genes with significant differences in means and moderate variance can stand out.

The SAM (Tusher *et al.* (2001)) algorithm uses an approach which depends on a smoothing criteria to determine a value of $\alpha$ to use. The rationale is that the $t$-statistic distributions should be identical for all (null) genes, so they impose a unifority criteria for the $t$-statistics to determine $\alpha$. They do not present theory however which connects their crietria with increased power, and in fact this method can go severely wrong with regards to the power of the results. Perhaps the problem is that we do not want the non-null genes to have the same distribution as the null genes, however their smoothing criteria is applied to all genes.

To demonstrate this we generated a simulated data set of 1000 genes, with 100 differentially expressed, and with the null genes each having a bimodal distribution. The data set can be obtained here:

`http://www.cbil.upenn.edu/PaGE/doc/files/bimodalnulltest.txt`

The $q$-values are equal to one minus the confidence. The lowest $q$-values produced by SAM on this data set are .5 (14 genes). The full set of $q$-values can be obtained here:

`http://www.cbil.upenn.edu/PaGE/doc/files/bimodalnulltest_SAM_qvals.txt`

In contrast, PaGE reports 17 genes at confidence greater than .8, all but one of which are true positives. The complete list of PaGE confidence values can be obtained here:

`http://www.cbil.upenn.edu/PaGE/doc/files/bimodalnulltest_PAGE_conf.txt`

Therefore for this data set SAM produces very poor results.

| $\alpha$ | num predicted |
|---:|---:|
| .01 | 0 |
| .1 | 9 |
| .18 | 12 |
| .2 | 15 |
| .3 | 15 |
| .32 | 11 |
| .33 | 10 |
| .34 | 10 |
| .35 | 0 |
| .4 | 0 |

Table 1: The effect of the $t$-statistic tuning parameter. Three replicates, one-class simulated data 1000 rows, with 50 differentially expressed. Confidence = .5. Mean of $\sigma$ is 0.227.

This example also illustrates, again, how different genes can be found with different values of $\alpha$, so there is not necessarily a single value that encompasses all types of differential expression.

In the example above on the page

`http://www.cbil.upenn.edu/PaGE/doc/files/3reps_conf.8_example.html`

the confidence is, as expected, close to the desired .8, regardless of the choice of $\alpha$. One might therefore consider taking the union of the results for several values of $\alpha$ in order to increase the power of the results. The problem with such an approach is that the sets of genes tend to overlap more on the true positives than on the false positives. Therefore taking the union of the sets tends to decrease the confidence of the results. For example taking the union of $\alpha$ = .0001, 3.5, and 20 in the above example increases the number of true positives to 175, but decreases the confidence to 0.738, which might be more confidence than it is worth giving up to find the extra 15 genes.

Therefore, it depends on the purposes of the investigator how they will want to deal with this issue. Often the moderate default value will be sufficient. When there are only a small number of differentially expressed genes, then one might want to adjust $\alpha$ to try to find them.

# 15 The choice of level confidence, statistic, transformation, and other parameter settings

### Finding an appropriate level confidence

The main parameter that the user will want to adjust is the level confidence. Every dataset is particular, so it is difficult to guess ahead of time what will be the best level confidence. Therefore PaGE allows you to set this parameter after the confidences have been calculated and the program displays a summary of the number of genes found over a range of confidences. If one has very few genes

| $\alpha$ | num up | num down |
|---:|---:|---:|
| .001 | 52 | 3 |
| .01 | 85 | 16 |
| .05 | 137 | 34 |
| .1 | 205 | 81 |
| .15 | 223 | 138 |
| .2 | 232 | 149 |
| .3 | 253 | 158 |
| .4 | 221 | 163 |
| .5 | 198 | 180 |
| .75 | 140 | 181 |
| 1 | 109 | 166 |
| 2 | 53 | 119 |

Table 2: The effect of the $t$-statistic tuning parameter. Six replicates, two-class mouse pancreas data. Confidence = .8. Mean of $D$ is 0.157.

| $\alpha$ | num up | num down |
|---:|---:|---:|
| .001 | 100 | 5 |
| .01 | 98 | 18 |
| .02 | 106 | 3 |
| .025 | 239 | 143 |
| .03 | 215 | 129 |
| .05 | 278 | 256 |
| .07 | 243 | 228 |
| .1 | 231 | 249 |
| .2 | 207 | 274 |
| .3 | 157 | 167 |

Table 3: The effect of the $t$-statistic tuning parameter. Eight replicates, one-class mouse pig heart valve data. Confidence = .8. Mean of $\sigma$ is 0.144.

differentially expressed, or if the data are very noisy, then a relatively low level confidence might be necessary to find them. Keep in mind that an FDR is very different from a $p$-value, so that while a $p$-value of .5 is practically useless, an FDR of .5 might be very useful. A set of predictions with FDR .5 has one out of every two genes being true positives. If one started with an array where only one out of every 100 genes were true positives, then this represents a dramatic enrichment for the true positives. Therefore it is not unreasonable to lower the level confidence as low as .5.

Conversely, if there is a large number of differentially expressed genes, on the order of thousands, then the user will generally want to set the the level confidence higher to see just the most confident predictions. One in this case might wish to raise the level confidence as high as .95, or even .99 in extreme cases.

### Using the ratio of means versus the $t$-statistic

For most datasets the user will probably want to start with the $t$-statistic option. If the $t$-statistic does not return many results, one should try adjusting $\alpha$ or using the other statistics. Even if there are many genes found, however, different statistics can pick up different kinds of differential expression. To illustrate this we generated a simulated dataset with two conditions, 100 "genes," and four replicates per condition. Two of the genes are differentially expressed. The 98 non-differentially expressed genes have moderate intensity: (beta distribution with mean 50, spread 35). Gene 0 is differentially expressed in the low intensity range (means of approximately 4 and 9 in the two conditions respectively). Gene 1 is differentially expressed in the high intensity range (means of approximately 400 and 450 in the two conditions respectively). Data available at

`http://www.cbil.upenn.edu/PaGE/doc/files/mean_vs_tstat_testdata.txt`

Table 4 shows the results using the ratio of means statistic (left) and the $t$-statistic (right). Using the ratio of means the low intensity differentially expressed gene (gene 0) is much more significant than the high intensity differentially expressed gene (gene 1). Conversely, using the $t$-statistic the high intensity differentially expressed gene is much more significant than the low intensity differentially expressed gene.

Ultimately there is no "best" statistic. A statistic can often be optimized for a single test, and there is substantial statistical theory about how to do this. But a statistic cannot typically be optimized for thousands of genes at once. Unfortunately there are no push-button solutions to this problem, each dataset is particular and must be treated as a special case, but by starting with the defaults the user can usually quickly hone in on reasonable parameter settings to suit their needs.

### Using logged versus unlogged data

The caveats of the previous section about relying on one statistic apply also to the different possible data transformations one can perform. Perhaps the most common is the log transformation. PaGE offers the option of performing this

| ID | Conf. ratio of means | Conf. $t$-statistic |
|---|---|---|
| 0 | 0.943 | 0.344 |
| 1 | 0.343 | 0.985 |
| max others | 0.424 | 0.367 |

Table 4: Comparison of results using the ratio of means statistic versus the $t$-statistic. Data consists of a simulated dataset of 100 genes. Gene with ID 0 is low intensity differentially expressed. Genes with ID 1 is high intensity differentially expressed. Genes with ID 2-99 are medium intensity non-differentially expressed. The bottom row shows the maximum confidence achieved by all other genes. Each method picks up one of the two differentially expressed genes at high confidence.

| ID | Conf. logged | Conf. unlogged |
|---|---|---|
| 0 | 0.673 | 0.344 |
| 1 | 0.343 | 0.985 |
| max others | 0.376 | 0.367 |

Table 5: Comparison of results using the logged versus unlogged data with the $t$-statistic. Same data as in Table 4. The bottom row shows the maximum confidence achieved by all other genes. Using the unlogged data was much better at finding gene 1, while using the logged data performed better on Gene 0 and completely lost gene 1 in the noise.

transformation when one is using the $t$-statistic. Using the same test datasets as above, Table 5 shows what happens to the confidence of gene 1 when the data are logged versus unlogged. The confidence of Gene 0 goes down while the confidence of Gene 1 goes up.

Thus one cannot trust either approach to perform better for all genes simultaneously. This happened because applying logs to the data, and then applying the $t$-statistic, which focuses on differences, is similar to first taking ratios, and then taking logs. For gene 1 in the above example, whose intensities are in the high range of the spectrum, it has a relatively small ratio, compared to the null genes whose intensities are in a lower range of the spectrum.

So when looking for a sparse set of genes among many, one may have to try several variations on the options. When there are many genes differentially expressed, then one will probably have their hands full regardless of what method they use.

## 16 PaGE output

PaGE output is designed to simplify interpretation of results. PaGE generates output in text and HTML format. At the top is a report of all of the parameter settings used in that particular run. The cutoff is reported which achieves the desired level confidence. For up-regulation this is called the *upper cutratio* and for down-regulation is is called the *lower cutratio*. If the $t$-statistic is used, then the $t$-statistic tuning parameter is reported. The levels, or patterns in the case of multiclass data, are then given.

The reest of the report contains the gene lists, which are organized into levels, or patterns. Levels, or patterns, are listed in dictionary type ordering. Each pattern is followed by the list of genes whose expression levels follow that pattern. The row identifier may be given as a link to further information regarding that array element, such as a link to GenBank. Following the gene identifier is the list of confidences for each position in the pattern. Genes are sorted by descending confidence within a pattern. When confidences are equal they are sorted by decreasing value of the statistic. The next column gives the averages of the intensities for that ID in the respective groups. Note that if a paired analysis is done, this number might be misleading, as it is not the paired mean being reported. These means are just given just as an aid in perusing the data. The next column gives the value of the statistic on the unpermuted data. The next column gives a name or description, if the user has provided a file which maps IDs to names or desriptions. On the far right of the row is a link that allows the user to eyeball the full set of intensities for that row of data. The means are also given. If a pattern has too many genes in it, it will be written to a separate page and a link will be included in the main page.

# References

Affymetrix Statistical Algorithms Reference Guide (2003).
**http://www. affymetrix.com/support/technical/manuals.affy**

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Proc. R. Statist. Soc. Series B*, **57**, 289–300.

Corimbia Inc. Probe Profiler Software.
`http://www.corimbia.com/ProbeProfiler.htm`

Irizarry, R.A., B.M. Bolstad, F. Collin, L.M. Cope, B. Hobbs, and T. Speed (2003). Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research* **31**, e15.

Manduchi E., G.R. Grant, S.E. McKenzie, G.C. Overton, S. Surrey, and C.J. Stoeckert (2000). Generation of patterns from gene expression data by assigning confidence to differentially expressed genes. *Bioinformatics* **16**, 685–698.

Saeed A.I., Sharov V., White J., Li.J., Liang W., Bhagabati N., Braisted J., Klapa M., Currier T., Thiagarajan M., Sturn A., Snuffin M., Rezantsev A., Popov D., Ryltsov A., Kostukovich E., Borisovsky I., Liu Z., Vinsavich A., Trush V., Quackenbush J. TM4: a free, open-source system for microarray data management and analysis. *Biotechniques.* 2003 Feb;34(2):374-8.

Storey, J.D. (2002). A direct approach to false discovery rates. *J. Roy. Statist. Soc.* **84**, 479–498.

Storey, J.D. and R. Tibshirani (2003). SAM thresholding and false discovery rates for discovering differential gene expression in DNA microarrays. pp. 272–290 in *The Analysis of Gene Expression Data*, ed. by G. Parmigiani, E.S. Garrett, R.A. Irizarry, and S. Zeger. Springer, New York.

Tusher, V.G., R. Tibshirani and G. Chu (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Nat. Acad. Sci.* **98**, 5116–5121.