BMC Bioinformatics

Research article



# A Bayesian approach to efficient differential allocation for resampling-based significance testing

Shane T Jensen[1,2], Sameer Soi[2] and Li-San Wang*[1,2,3,4]

Address: [1]Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA, 19104 USA, [2]Genomics and Computational Biology Program, University of Pennsylvania, Philadelphia, PA, 19104 USA, [3]Penn Center for Bioinformatics, University of Pennsylvania, Philadelphia, PA, 19104 USA and [4]Department of Pathology and Laboratory Medicine, University of Pennsylvania, Philadelphia, PA, 19104 USA

Email: Shane T Jensen - stjensen@wharton.upenn.edu; Sameer Soi - ssoi@mail.med.upenn.edu; Li-San Wang* - lswang@mail.med.upenn.edu

* Corresponding author

Published: 28 June 2009

*BMC Bioinformatics* 2009, **10**:198 doi:10.1186/1471-2105-10-198

Received: 23 December 2008
Accepted: 28 June 2009

This article is available from: http://www.biomedcentral.com/1471-2105/10/198



## Abstract

**Background:** Large-scale statistical analyses have become hallmarks of post-genomic era biological research due to advances in high-throughput assays and the integration of large biological databases. One accompanying issue is the simultaneous estimation of p-values for a large number of hypothesis tests. In many applications, a parametric assumption in the null distribution such as normality may be unreasonable, and resampling-based p-values are the preferred procedure for establishing statistical significance. Using resampling-based procedures for multiple testing is computationally intensive and typically requires large numbers of resamples.

**Results:** We present a new approach to more efficiently assign resamples (such as bootstrap samples or permutations) within a nonparametric multiple testing framework. We formulated a Bayesian-inspired approach to this problem, and devised an algorithm that adapts the assignment of resamples iteratively with negligible space and running time overhead. In two experimental studies, a breast cancer microarray dataset and a genome wide association study dataset for Parkinson's disease, we demonstrated that our differential allocation procedure is substantially more accurate compared to the traditional uniform resample allocation.

**Conclusion:** Our experiments demonstrate that using a more sophisticated allocation strategy can improve our inference for hypothesis testing without a drastic increase in the amount of computation on randomized data. Moreover, we gain more improvement in efficiency when the number of tests is large. R code for our algorithm and the shortcut method are available at http://people.pcbi.upenn.edu/~lswang/pub/bmc2009/.

## Background

### *Nonparametric tests in multiple hypothesis testing scenarios*

Large-scale statistical analyses have become hallmarks of post-genomic era biological research due to advances in high-throughput assays and the integration of large biological databases. As the analysis becomes larger and more complex, various kinds of computational issues arise. The context of our investigation is *multiple testing*, the simultaneous estimation of p-values for a large number of hypothesis tests. For example, in a typical control-treatment microarray experiment, the goal of the analysis may

be to identify target genes by applying the same testing procedure on each of the genes and selecting those that show the most extreme differential expression.

Most multiple testing scenarios involve the assumption of a parametric null distribution (such as the normal or *t* distribution) for each observed test statistic. However, in many applications, this parametric assumption may be unreasonable, resampling-based p-values are the preferred procedure for establishing statistical significance. For example, in the usual permutation test framework, resamples are generated by randomly permuting the treatment and control labels among the available data samples. We then calculate the test statistic for each of these resamples and calculate the p-value for each gene as the fraction of the resamples that have more extreme test statistics than the observed test statistic for that gene. Ideally, we would be able to evaluate the test statistic for every possible resample, and thus calculate the resample-based p-value exactly. However, this is usually not feasible for datasets involving many replicates, so the usual procedure is to use Monte Carlo simulation to estimate each p-value based on a large set of resamples. As an example, an option in the popular SAM microarray analysis software [1] allows the user to use permutation tests to assess the p-value without the normality assumption. A similar resampling scheme for estimating p-values can be based on the bootstrap. In [2], a nonparametric test procedure is applied to every gene to examine how well the expression profile of a gene (say over a time course) fits some preset order-restrictions; the p-value of the test is obtained using 50000 bootstrap resamples *per gene*. We refer the reader to [3,4] for the rationale and more details on bootstrapping, permutation tests, and other nonparametric tests. In this paper we collectively refer these methods as *resampling* procedures and a randomly generated sample (whether bootstrap or permutation-based) is called a *resample*.

This paper focusses on the following setting: we have $N$ units (eg. genes), and we want to conduct a hypothesis test for each gene $i$ based on observed test statistic $T_i$. We do not want to make any parametric assumptions about this test statistic, so the p-value $p_i$ for each test needs to be estimated by a resampling procedure. The additional element that is implicit in our framework is that the number of tests $N$ is large (can be as high as $10^6$ for genome-wide association studies), so we need to control for the large number of tests being performed. Many multiple testing procedures focussed on control of the *family-wise error rate* (FWER), with a popular choice being the Bonferroni correction [5]. More recently, the focus in multiple testing procedures has shifted to control of the *false discovery rate* (FDR) [6-8], which is much less conservative than FWER-control procedures. Since this current work was motivated by biological applications, we will use the terms *gene* and *unit* interchangebly, with the understanding that our methods are applicable to any multiple testing situation.

### *Typical Uniform Resampling Strategies*

Typical resampling procedures for p-value estimation use an equal number of resamples, say $B$, assigned to each of $N$ genes, for a total of $N \times B$ resamples. Even in the simple framework where each gene will be assigned the same number of resamples, there are several alternative strategies for resampling-based inference. The first issue is whether each resample should be performed by randomly permuting the treatment and control labels of an entire column (across all genes) of data values, with the alternative being that treatment and control labels are permuted within each gene independently. We refer to the first strategy as a *column-wise* procedure and the second strategy as an *gene-independent* procedure. Many recent investigations (eg. [9-11]) argue for column-wise resampling procedures in order to retain potential dependencies between genes. Other recent microarray investigations (eg. [12]) have employed gene-independent resampling procedures. Clearly, a column-wise resampling procedure allocates resamples to all genes simultaneously, which implies a uniform allocation of resamples across genes. Although this column-wise strategy is preferred in certain situations, it suffers from the same inefficiencies as any uniform allocation procedure: genes that are clearly distant from the decision threshold will receive the same number of resamples as genes that are quite near the threshold. We focus on a gene-independent resampling procedure since it provides a more flexible framework for differential allocation of resamples among genes, which is the primary motivation for our current work.

Another issue is whether or not to combine resampled test statistics across genes when estimating the p-value for each gene. Many researchers (eg. [7]) prefer a *concatenation* procedure that uses all available resampled test statistics (across all genes) to achieve a higher resolution on the resampling-based null distribution when estimating each p-value. Since all resampled test statistics (across all genes) are used for each p-value calculation, there is little distinction between resampling strategies based on uniform allocation of resamples across genes versus differential allocation of resamples across genes. However, a concatenation procedure is only reasonable when the resampling-based null distribution is similar across genes, which is an uncomfortable assumption in many applications, such as genome-wide association studies when the allelic frequencies vary across loci. In these applications, a non-concatenation or *gene-separate* procedure would be preferred. Recent work (eg. [13,14]) proposes concatenation of statistics across only subsets of genes to correct for the fact that the null distribution is likely to differ between significant and non-significant genes. In this paper, we

will focus on situations where *gene-separate* (non-concatenation) procedures are preferred, which is the area where differential allocation of resamples provides a substantial efficiency gain over a uniform allocation strategy.

In most multiple testing situations the vast majority of units are truly non-significant, which means that a uniform allocation strategy is devoting a large proportion of resamples to test statistics that are not even close to significant. For most estimated p-values that are quite large or extremely small (ie. far away from our decision threshold $p_0$), then we are reasonably confident about our decision based on those p-values without need for a high degree of p-value accuracy (large number of resamples). Instead, we should focus a larger number of resamples on the estimation of p-values that are near to our decision-making threshold. For example, one may limit the number of resamples for a gene when the number of resamples with test statistic exceeding the actual statistic is larger than $p_0 \times B$, since the p-value of this gene will definitely be higher than the threshold $p_0$ when all $B$ resamples are computed. The gene is clearly nonsignificant, so we can stop evaluating more resamples for this gene and save computational time. This simple heuristic, which we call the *shortcut* approach, has been discussed previously [15] and implemented more recently in the popular software PLINK [16].

In this paper, we develop a principled iterative procedure for allocating different numbers of resamples to each unit. The overall intuition behind our approach is similar to the shortcut method in that we want to preferentially allocate more resamples to genes which have "borderline" p-values, i.e., p-values near to our classification threshold. The main difference is how the resample allocations are determined: we use a Bayesian-inspired approach that assigns resamples to each unit based on its individual "risk", the chance that the current p-value estimate leads to a misclassification of the unit. The goal is to lower the numbers of classification errors, since we are giving a higher resolution to the null distribution of genes that are more likely to be misclassified in a uniform allocation setting. This higher resolution comes at the sacrifice of resamples to non-borderline genes that should not need a very resolute null distribution for correct inference.

A detailed description of our differential allocation procedure is provided in the Methods section. The Results Section includes an experimental comparison that demonstrates the gains of our procedure over uniform procedures using two publicly available datasets: one microarray dataset on breast cancer [17], and one genome-wide association study [18] where computational efficiency in p-value estimation is a necessary concern due to its size. Our procedure maintains a low error-rate (low rates of false positives and false negatives) while using substantially fewer resamples in total. We then provide an additional experimental comparison to demonstrate that our method outperforms the shortcut method.

## Methods

### *Differential Allocation of Resamples Using Risks*

We separate the description of our procedure into several subsections for clarity of presentation.

#### *Algorithm Initialization*

The input data for our algorithm is an $N \times J$ matrix of data values, where $N$ is the number of genes and $J$ is the number of observations per gene. Our algorithm is initialized by a uniform allocation *burn-in* round, in which we assign $B_0$ resamples to each gene, where $B_0$ is a proportion of the $B$ resamples that would be assigned to each gene by the typical uniform resampling procedure. Each of these resamples gives us a test statistic under the resampling-based null distribution, which we can use to get an initial p-value estimate for each gene.

Based on the given threshold $p_0$ and our current estimated p-values $\hat{p}_i$, we have the current classification for each gene $i$: gene $i$ is significant if $\hat{p}_i \leq p_0$ or gene $i$ is non-significant if $\hat{p}_i > p_0$. In case when $p_0$ is determined using other criterion such as FDR, we use these p-value estimates to calculate our decision threshold $p_0$ using the original FDR-control procedure proposed by [19].

#### *Differential Allocation*

Our algorithm now proceeds sequentially through multiple rounds and in each round a total of $K$ new resamples are assigned. We want to allocate new resamples differentially to each gene $i$ with the goal of minimizing the expected number of *mis-classified* genes ie. either non-significant genes that are inferred to be significant (false positives) or significant genes that are inferred to be non-significant (false negatives). Our framework treats either type of error (false-positives vs. false-negatives) as equally bad, though our approach could be easily generalized to differentially weight the two types of errors. Our proposed strategy is to assign new resamples with probability proportional to the *risk* $R_i$ of each gene $i$: the current probability of that gene $i$ being misclassified.

$$\begin{aligned} R_i &= \mathrm{P}(\mathrm{miss}_i \mid \hat{p}_i, p_0) \\ &= \mathrm{P}(\mathrm{FN}_i \mid \hat{p}_i, p_0) + \mathrm{P}(\mathrm{FP}_i \mid \hat{p}_i, p_0) \\ &= \mathrm{P}(p_i \leq p_0) \cdot \mathrm{I}(\hat{p}_i > p_0) + \mathrm{P}(p_i > p_0) \cdot \mathrm{I}(\hat{p}_i \leq p_0) \end{aligned}$$

where $p_i$ represents the true p-value for gene $i$. Only one of these two terms is non-zero, since any gene $i$ can only be

considered as either a false positive or false negative (not both) based on its current estimated p-value $\hat{p}_i$. We estimate the probabilities $P(p_i \le p_0)$ and $P(p_i > p_0)$ based on the posterior distribution of p-value $p_i$ for gene $i$. Let $n_i$ be the number of resamples already performed on gene $i$ and let $a_i$ be the number of resample test statistics that are more extreme than the observed test statistic for gene $i$. This pair of numbers $(a_i, n_i)$ contains all the information we currently have for gene $i$. Assuming a uniform prior on each $p_i$, $p_i \sim \text{Beta}(1, 1)$, and with a binomial likelihood for our extreme resample counts $a_i \sim \text{Bin}(n_i, p_i)$, then we have:

$$p_i \mid a_i, n_i \sim \text{Beta}(a_i + 1, n_i - a_i + 1) \quad (1)$$

so each probability $R_i$ becomes

$$\begin{aligned} R_i &= P(p_i \le p_0) \cdot I(\hat{p}_i > \hat{p}_0) + P(p_i > p_0) \cdot I(\hat{p}_i \le p_0) \\ &= B(p_0, a_i + 1, n_i - a_i + 1) \cdot I(\hat{p}_i > p_0) \\ &\quad + [1 - B(p_0, a_i + 1, n_i - a_i + 1)] \cdot I(\hat{p}_i \le p_0) \end{aligned} \quad (2)$$

where $B(x, a, b)$ is the CDF of the Beta$(a, b)$ distribution evaluated at $x$. $B(x, a, b)$ is often also referred to as the *incomplete Beta function*. In Figure 1, we see the risk for different locations of the posterior distribution $p(p_i|a_i, n_i)$. This illustration shows that the risk $R_i$ is the amount by which the posterior distribution (1) for gene $i$ overlaps the significance threshold $p_0$.

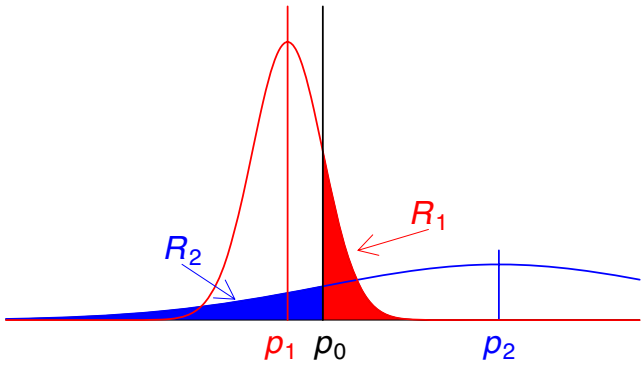


**Figure 1**
**Sample figure title**. Illustration of the risks associated with two different p-values. The red density is the posterior distribution of p-value $p_1$. The blue density is the posterior distribution of p-value $p_2$. The decision threshold for assessing significance is denoted as $p_0$. The risk $R_1$ (associated with p-value $p_1$) is the red area on the right of the decision threshold $p_0$. The risk $R_2$ (associated with p-value $p_2$) is the blue area on the left of the decision threshold $p_0$.

After the end of each round, $K$ new resamples have been assigned proportional to the risks given in (2), and for each affected gene, the new p-value $p_i$ and the risk $R_i$ must be calculated. The algorithm stops when the total number of resamples assigned reaches a preset cap $B_{tot}$. We should note that the above scheme considers the decision threshold $p_0$ to be fixed and known, when it is actually itself an estimated quantity. A more general procedure that acknowledges the uncertainty in both the p-values $p_i$ and decision threshold $p_0$ for FDR is the focus of continuing research. We provide a more detailed description of our proposed differential allocation algorithm below.

**Input**: Microarray measurements $\boldsymbol{g}_i = (g_{i1}, \ldots g_{ij})$ for each gene $i$, $1 \le i \le N$.

**Output**: Set of significant genes as defined by threshold $p_0$.

*Parameters*

1. $B_0$: number of reseamples per gene for burn-in.

2. $B$: average number of resamples to allocate per gene, so that $N \times B$ is total number of resamples to be used.

3. $K$: number of resamples allocated in each round.

*Algorithm*

1. For each gene $i$, compute observed test statistic $f_i = f(\boldsymbol{g}_i)$.

2. *Burn-in Allocation*: $n_i \leftarrow B_0$

3. *Iterative Allocation*: Repeat:

(a) For each gene $i$, calculate $a_i$ = number of $n_i$ resamples with test statistic $\ge f_i$ and set $\hat{p}_i = a_i/n_i$

(b) For each gene $i$, compute $R_i \leftarrow B(p_0, a_i + 1, n_i - a_i + 1)$. If $p_i \ge p_0$ then set $R_i \leftarrow 1 - R_i$.

(c) For each gene $i$, compute $w_i = R_i/\Sigma_i R_i$.

(d) While $j < K$:

i. Select a gene $b$ from the set $(1, 2, \ldots N)$ with probability $(w_1, w_2, \ldots, w_N)$

ii. Assign a resample to selected gene $b$: $n_b \leftarrow n_b + 1$

iii. $j \leftarrow j + 1$

4. Output the set of significant genes by applying threshold $p_0$ on final set of $\{\hat{p}_i\}$.

### *The Shortcut Method*

An alternative differential allocation idea that we call the *the shortcut method* is to stop allocating resamples to any genes which have already accumulated enough non-extreme test statistics to guarantee that the null hypothesis for those genes will not be rejected. [15] discuss a sequential shortcut method for Monte Carlo estimation of p-values and more recently a shortcut method has been implemented in [20]. The popular software PLINK [16] for genome-wide association studies allows for more sophisticated approaches, such as using a confidence interval of the estimated p-value of a unit to decide if more resamples are needed.

Again let $N$ be the number of genes and let $B$ be the number of resamples that we would allocate to each gene in a uniform allocation scheme, so that we have a total of $N \times B$ resamples available to us. We again consider an iterative scheme where $n_i$ is the number of resamples already performed for gene $i$ and let $a_i$ is the number of resample test statistics that are more extreme than the observed test statistic for gene $i$. If a particular gene $i$ has accumulated enough non-extreme resample test statistics, i.e. if $(n_i - a_i) > B \cdot p_0$, then the resampling-based p-value $\hat{p}_i$ is guaranteed to exceed the threshold $p_0$ and so allocating any more resamples to gene $i$ is pointless. All remaining $(B - n_i)$ resamples that we would have devoted to gene $i$ can now be allocated to other genes that still have a chance of rejecting the null hypothesis. This shortcut approach clearly differs from our proposed method in terms of how resamples are differentially allocated, but both should still be more efficient than a uniform allocation scheme. Another major difference is that our differential allocation method will also assign fewer resamples to genes when the p-value is much lower than the cutoff, whereas the shortcut method always tends to allocate more resamples to genes with a lower p-value.

### *Experimental Comparision*

#### *Application to a breast cancer microarray dataset*

The Hedenfalk et al. breast cancer dataset [17] consisted of 7 sporadic cases, 7 cases with BRCA1 mutations, and 8 cases with BRCA2 mutations. Following the guidelines in [7], we only examine samples associated with either BRCA1 and BRCA2 mutations, which results in 8 samples for BRCA1 and 7 samples for BRCA2. Following the pre-processing procedure in [7], we $\log_2$-transformed all measurements and removed outlier genes (defined as genes having any expression level above 20 in [7]); this left us with 3170 genes for further analysis. The subjects were divided into two groups. For each gene, we tested whether the mean expression levels of the two groups are significantly different. We used the absolute value of the Student's t-statistic and used permutation tests to compute the significance: the p-value of the gene is the fraction of random permutation resamples with larger statistic scores than the correct grouping of subjects. We varied the number of resamples per gene to see how the p-value estimation of our algorithm and the uniform allocation improved as the number of resamples increased. We assessed the accuracy by computing, as a reference, the exact p-values calculated by enumerating all $\binom{15}{7} = 6435$ possible resamples for each gene. The error of any p-value estimation is the number of genes mislabeled as significant or nonsignificant when compared with the significance calls using these reference exact p-values and a significance threshold of 0.0001. All computations were done using the R statistical software [21].

#### *Application to a Parkinson disease genome-wide association study dataset*

The Parkinson's dataset [18] consisted of the genotype information of 402,582 SNPs on 271 cases and 270 controls. We randomly partitioned the dataset into 30 subsets of 13,626 SNPs each on average, and applied our algorithm to each subset separately. For each SNP, we used the chi-square statistic for the $3 \times 2$ contingency table, and computed the exact chi-square test p-value with 2 degrees of freedom as the "reference" p-value. We also applied our differential allocation algorithm by setting $B = 1000$, $B_0 = 100, 250, 500, 1000$ (uniform allocation), and $p_0 = 10^{-4}$ in the differential allocation algorithm. We then computed the accuracy and false discovery rate of the output from the four allocation algorithms using different p-value cutoffs; the "reference" set of significant SNPs were determined using the "reference" p-value using the same p-value cutoff.

#### *Simulation study to compare our algorithm and the shortcut method*

We compared our method and the shortcut method using the following parameter settings:

1. We use $N = 300,000$, typical for genome-wide association studies. The actual p-values of all markers are generated as follows. First, for each marker we randomly sample an integer between 1 and $N$; the p-value

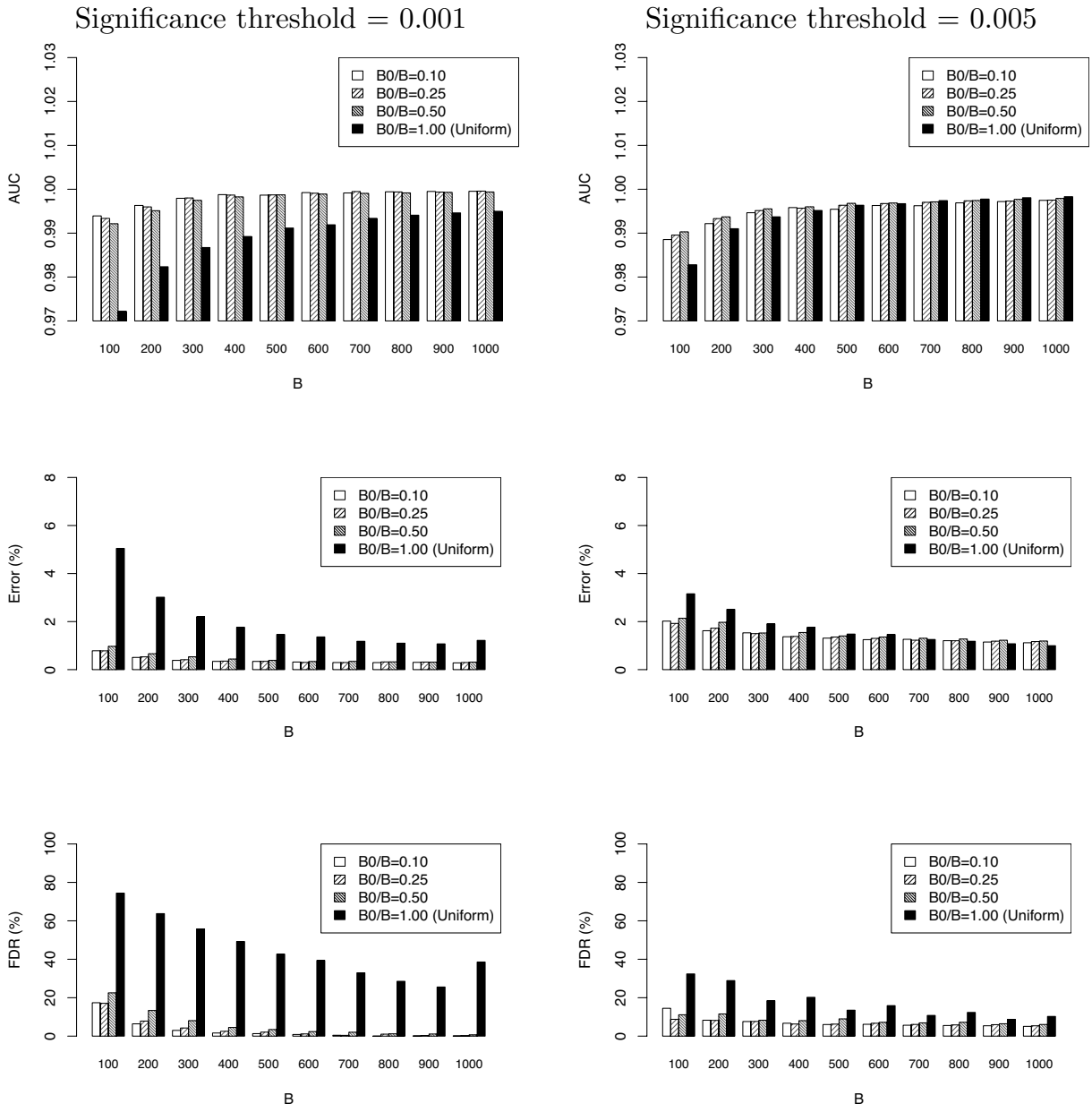


**Figure 2**
**Sample figure title**. Area under ROC curve (AUC), Error (defined as (FN+FP)/(P+N)), and false discovery rate (FDR) of the uniform ($B/B_0$ = 1) and differential allocation algorithms using the Hedenfalk et al. gene expression dataset. Left: p-value cutoff = 0.001; right: p-value cutoff = 0.005.

of the marker is this number divided by $N$. Thus each marker will have a p-value between $1/N$ and 1 at this moment. We then replace the p-values of five of the markers by $10^{-7}$ to represent real significant markers.

2. For both methods, we use the same p-value cutoff settings:

$10^{-5}$, $2 \times 10^{-5}$, $5 \times 10^{-5}$, $10^{-4}$, $2 \times 10^{-4}$, $5 \times 10^{-4}$, $10^{-3}$.

3. For the shortcut method, each iteration allocates $B$ = 10 resamples. The algorithm stops when the average number of resamples per marker exceed 100.

4. We use a simplified version of the adaptive permutation algorithm in PLINK, a program widely used in the analysis of genome-wide association studies [16]. At each iteration, the p-value estimate of marker $i$ is $\hat{p}_i$

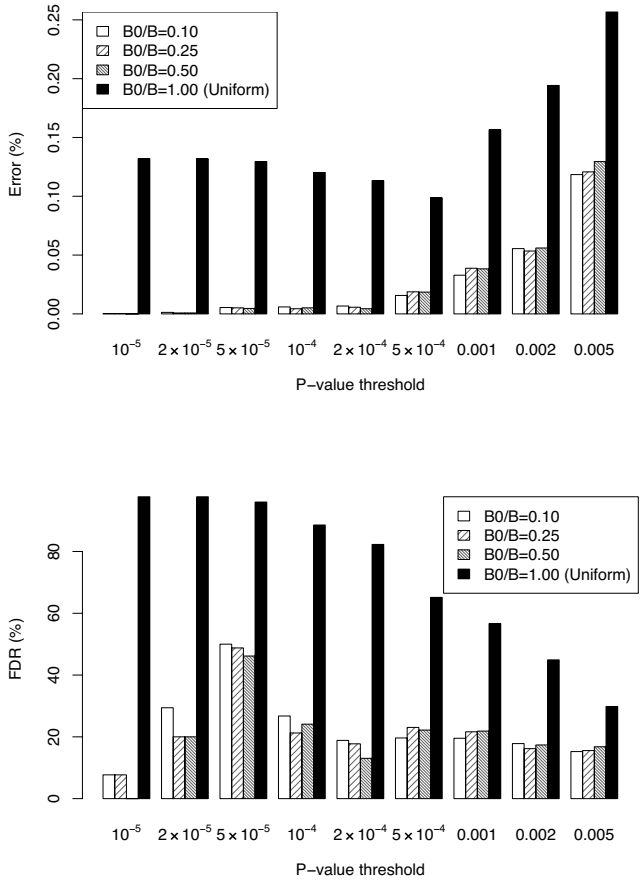


**Figure 3**
**Sample figure title**. Error (defined as (FN+FP)/(P+N)) and false discovery rate of the uniform ($B/B_0$ = 1) and differential allocation algorithms using the public Parkinson Disease genome-wide association study dataset.

= $(1 + D_i)/(2 + F_i)$, where $F_i$ is the total resamples allocated to $i$ so far, and $D_i$ is the number of such resamples that yield higher statistics than the actual statistic (this is determined in the simulation by Bernoulli trials with success probability $p_i$). This $\hat{p}_i$ estimate is equivalent to the posterior mean when assuming a uniform prior distribution, and improves upon the poor performance [22] of the usual estimate $\hat{p}_i = D_i/F_i$ when $D_i = 0$. If the actual p-value cutoff $p_0$ is outside the $c$-level confidence interval for $p_i$ then marker $i$ will not be included for resample allocation in the next round. The confidence interval is approximated by a normal distribution with mean $\hat{p}_i$ and standard deviation $\hat{p}_i(1-\hat{p}_i)/\sqrt{F_i}$. We use $c$ = 0.01,0.05, 0.1, 0.3, 0.5 in our simulation.

5. For our algorithm, $B_0 = K = 10$, $B = 100$.

## Results and Discussion

### *Experimental Validation*

We applied our algorithm to two different datasets to check how efficient it is compared with the conventional uniformly-allocated re-sampling. The first dataset is a publicly available microarray dataset to detect genes differentially expressed across two conditions. The second, much larger dataset, is a publicly-available genome-wide assocation study on Parkinson's disease [18].

#### *Application to a breast cancer microarray dataset*

Our algorithm was first applied to the microarray dataset presented in [17]. The details of preprocessing and application of the algorithm to this data are presented in the Methods Section. The results are in Figure 2. We observe that in the microarray dataset, the differential allocation algorithm ($B_0/B < 1$) outperforms the uniform allocation algorithm ($B_0/B = 1$) substantially, though the gap becomes smaller when $B$ increases. We also measured the areas under ROC curve to eliminate the effect of selecting a particular threshold of significance, and observed the same trends (data not shown). In both datasets, the choice of burn-in proportion $B_0/B$ for the differential allocation algorithm did not seem to affect the performance of the algorithm.

#### *Application to a Parkinson disease genome-wide association study dataset*

The results from the previous section suggest our algorithm has the best improvement over the uniform allocation when the number of possible resamples is relatively small. In this section, we test our algorithm on a publicly-available genome-wide association study where the number of possible resamples is relatively large. Typical datasets in genome-wide association (GWA) studies may consist of several thousand case and control subjects each, using single nucleotide polymorphism (SNP) genotyping arrays that can genotype up to $10^6$ SNPs. The most common goal of a genome-wide association study is to find SNP(s) that are highly correlated with the case/control status. One simple way to test the association is to run chi-square tests on the two-way 3 × 2 contingency table between the genotype of each SNP (zero, one, or two copies of the minor allele) and the case-control status [23]. Existence of such SNPs suggests nearby genomic regions may carry significant genes, regulatory motifs, or other DNA sequences that may affect the disease risk.

This setting is an important test of computationally efficient resampling-based procedures for several important reasons. First, the high number of SNPs being tested implies a very stringent p-value threshold if we take the issue of multiple testing into consideration: setting p-

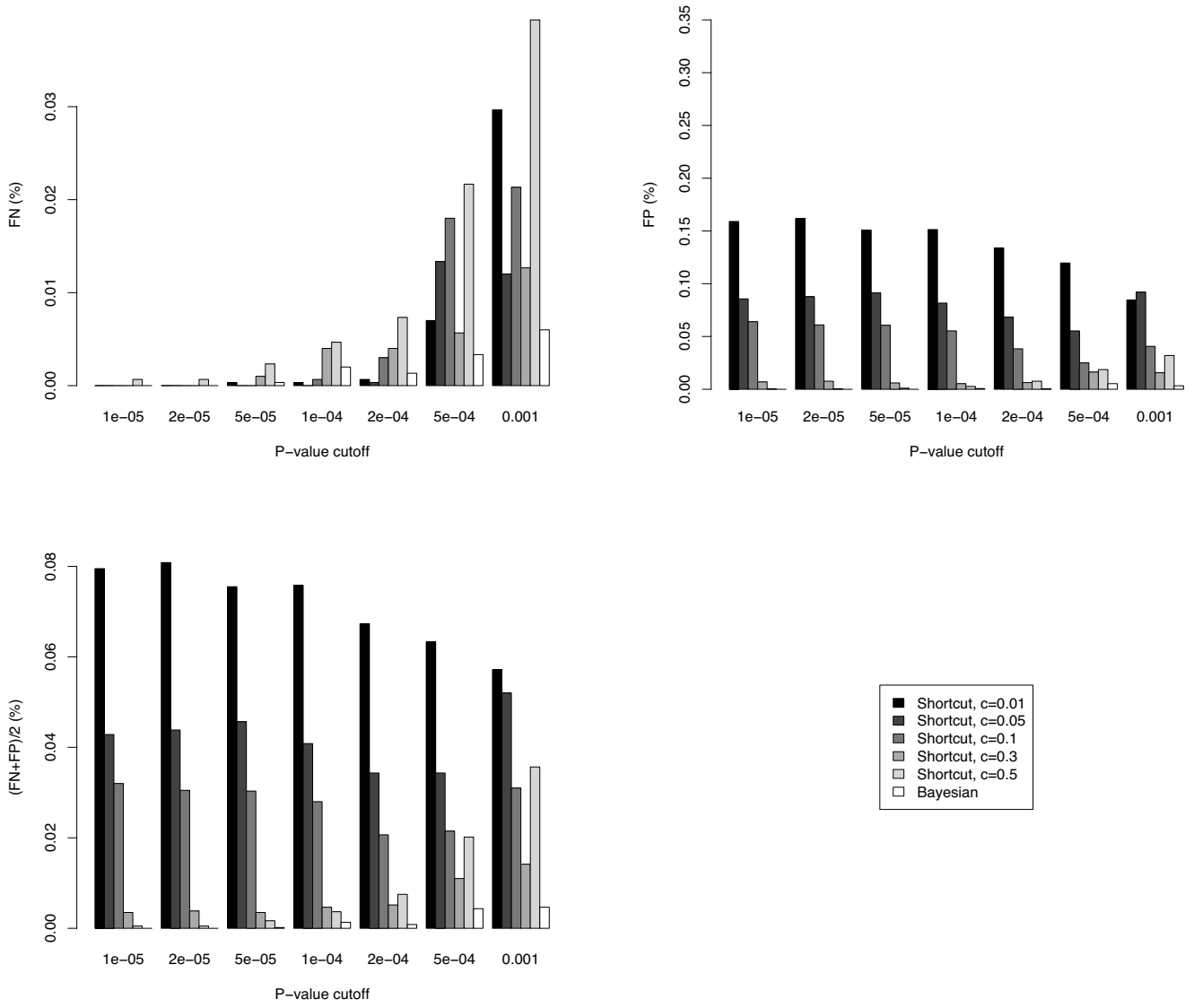


**Figure 4**

**Sample figure title**. FN, FP, and Average Error (defined as (FN+FP)/2) of the shortcut and our differential allocation algorithm (bayesian) in our simulation study. The value *c* in the legend is the level of confidence interval used in the shortcut method; see text for more details.

value cutoff at $10^{-5}$ or lower is typical, so any resampling-based p-value computation for each SNP requires at least $10^5$ resamples if uniform allocation is used. Second, the high number of subjects means evaluating the test statistic for each resample is more costly. Finally, although we focus on simple chi-square tests as a proof of concept for our procedure, even more complex and computational demanding tests that may involve interactions between multiple SNPs and pedigree information relating subjects are being actively developed and applied to improve the sensitivity of GWA studies. As a example, it is common to consider the maximum p-value between multiple tests, such as an allelic test and a genotypic test, in a GWA analysis. These tests may employ statistics that are computationally expensive, and p-values have to be evaluated using resampling if exact p-value formulas are not available.

As an illustration of our procedure in this difficult setting, we applied our algorithm to a public Parkinson genome-wide association study dataset [18]. Refer to the Methods Section on details of the dataset and the application of our algorithm. The results are summarized in Figure 3. Notice that our procedure has excellent accuracy and false discovery rate: at most 20% except when the p-value cutoff is $5 \times 10^{-5}$. Moreover, the proportion of "burn-in" permutation resamples has little effect on the accuracy of the differential allocation algorithm, probably because the

enormous number of total SNPs implies there are always enough resamples from nonsignificant SNPs to be reallocated when needed. Nonuniform allocation always outperforms uniform allocation by a substantial amount. Since $B = 1000$, for p-value thresholds lower than $10^{-4}$ only SNPs with 0 as their estimated p-values can pass the threshold under the uniform allocation. The uniform allocation algorithm has much higher error and false discovery rate because p-value estimation for significant and borderline SNPs is less accurate, and the situation is not improved even when the p-value threshold increases to $10^{-3}$.

*Simulation comparison to shortcut method*

In addition to demonstrating increased efficiency over a uniform allocation scheme, we also evaluate our method against the shortcut method, which is also described in our Methods section. We use $N = 300,000$ markers, typical for genome-wide association studies. We generate the "actual" 300,000 p-values following a uniform distribution since we know that the p-values of all (but a few) markers should be uniformly distributed in a well-designed genome-wide association study where no confounding factors such as population stratification exist. We evaluate our performance relative to the shortcut method using simulated p-values directly. See the Methods section on details of the simulation.

Please see Figure 4 for the results of the simulation. As can be seen, our method consistently outperforms the shortcut method. Note that as we increase the p-value cutoff, the FN rate increases and the FP rate decreases for the shortcut method. Moreover, when the value of confidence level $c$ increases in the shortcut method, the FP rate decreases but the FN rate varies in a more complex pattern affected by both the confidence level $c$ and $p_0$. Small values of $c$ in the shortcut method has good FN rate in general (and outperforms the bayesian method for $p_0 = 10^{-4}$ and $2 \times 10^{-4}$, but the FP rate is too high. On the other hand, a large setting of $c$ has good FP rate and bad FN rate. These observations hint that a symmetric test for both FP and FN in the shortcut method may be suboptimal (using the same value of $c$, level of confidence interval for both FP and FN scenarios) and an asymmetric approach such as our algorithm is preferred. Another contributing factor is that our approach is more global in the sense of allocating resamples proportional to the risks across all markers as opposed to the shortcut approach that treats each marker independently of the progression of other markers.

*Running time*

We explored the overhead associated with our differential allocation approach and found it to be negligible on a modern computer. We computed the running time of the shortcut method and our differential allocation algorithm for 5 repetitions of our simulation involving 300,000 SNPs on a dual-quad-core Xeon linux server using R (64-bit version 2.8.1; our implementation is single-threaded and no parallelization is involved). Since the permutation tests in this simulation are generated by random p-values, the running time is almost entirely a function of the overhead of the allocation algorithms, not the individual statistical tests. The average running time of shortcut method in this situation was about 3.5 minutes, and the average running time of our algorithm was 1 minute, suggesting that neither method has substantial overhead. Certainly in a situation with more complex individual statistical tests, the running time of either approach will be dominated by the unavoidable calculations of each test statistic.

## Conclusion

In this paper we presented a new approach to more efficiently assign resamples (such as bootstrap samples or permutations) within a nonparametric multiple testing framework. We formulated a Bayesian-inspired approach to this problem, and devised an algorithm that adapts the assignment of resamples iteratively with negligible space and running time overhead. In two experimental studies, a breast cancer microarray dataset and a genome wide association study dataset for Parkinson's disease, we demonstrated that our differential allocation procedure is substantially more accurate compared to the traditional uniform resample allocation. In a simulation study we showed our algorithm outperforms the simpler shortcut method under various settings. It is worth emphasizing that our methodology is not ideally suited for the accurate estimation of all p-values, especially p-values far from the significance threshold (in either direction). Rather, our methodology focusses on the accuracy of significance decisions by ensuring that p-values near the decision threshold are most accurately estimated.

The idea of using a non-uniform search among a large number of tests is quite common in other multiple testing situations. An example is efficient variable selection in regression models where the number of covariates is very large. Similar applications can also be found elsewhere: in finance, [24] used a stepwise regression procedure to predict bankruptcy, where significant predictors are added (from a large pool of possible predictors) sequentially using a procedure where there is differential allocation for the threshold of significance. Techniques such as this are different from our situation since we are taking a non-parametric approach to a simpler testing situation, but we still share the similar idea that one can gain power by differentially allocating resources towards the tests that are most likely to be significant. When individual tests are simple to compute, e.g., Fisher's exact test on small contingency tables when the p-value can be computed exactly, the gain by our algorithm or other differential allocation methods is limited. However, a differential

allocation approach is much more important when more computationally intensive tests are used, such as in Gene Set Enrichment Analysis [25], or family-based association tests in genome-wide association studies [26].

## Authors' contributions

SJ and LW designed this study and developed the new algorithm. LW coded the new algorithm and the shortcut method. SS and LW ran the experiments. All three authors wrote and approved the manuscript.

## Acknowledgements

This study used data from the SNP Database at the NINDS Human Genetics Resource Center DNA and Cell Line Repository http://ccr.coriell.org/ninds, as well as clinical data. The original genotyping was performed in the laboratories of Drs. Singleton and Hardy, (NIA, LNG), Bethesda, MD USA.

## References

1. Tusher VG, Tibshirani RJ, Chu G: **Significance analysis of microarrays applied to the ionizing radiation response.** *Proc Nat Acad Sci* 2001, **98:**5116-5121.
2. Peddada SD, Lobenhofer EK, Li L, Afshari CA, Weinberg CR, Umbach DM: **Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference.** *Bioinformatics* 2003, **19:**834-841.
3. Conover W: *Practical Nonparametric Statistics* 3rd edition. New York, NY, USA: Wiley; 1998.
4. Efron B, Tibshirani RJ: *An Introduction to the Bootstrap* Boca Raton, FL, USA: Chapman & Hall/CRC; 1994.
5. Miller RG: *Simultaneous statistical inference* 2nd edition. New York, NY, USA: Springer Verlag; 1981.
6. Efron B, Tibshirani R: **Empirical Bayes methods and false discovery rates for microarrays.** *Genetic Epidemiology* 2002, **23:**70-86.
7. Storey JD, Tibshirani R: **Statistical significance for genomewide studies.** *Proceedings of the National Academy of Sciences* 2003, **100:**9440-9445.
8. Scheid S, Spang R: **A Stochastic Downhill Search Algorithm for Estimating the Local False Discovery Rate.** *IEEE Transactions on Computational Biology and Bioinformatics* 2004, **1(3):**98-108.
9. Reiner A, Yekutieli D, Benjamini Y: **Identifying differentially expressed genes using false discovery rate controlling procedures.** *Bioinformatics* 2003, **19:**368-375.
10. Ge Y, Dudoit S, Speed TP: **Resampling-based multiple testing for microarray data analysis.** *TEST* 2007, **12:**1-77.
11. Jain N, Cho H, O'Connell M, Lee JK: **Rank-invariant resampling based estimation of false discovery rate for analysis of small sample microarray data.** *BMC Bioinformatics* 2005, **6:**187.
12. Tusher VG, Tibshirani R, Chu G: **Significance analysis of microarrays applied to the ionizing radiation response.** *Proceedings of the National Academy of Sciences* 2001, **98:**5116-5121.
13. Xie Y, Pan W, Khodursky A: **A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data.** *Bioinformatics* 2005, **21:**4280-4288.
14. Yang H, Churchill G: **Estimating p-values in small microarray experiments.** *Bioinformatics* 2007, **23:**38-43.
15. Besag J, Clifford P: **Sequential Monte Carlo p-values.** *Biometrika* 1991, **78:**301-304.
16. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira M, Bender D, Maller J, Sklar P, de Bakker P, Daly M, Sham P: **PLINK: a toolset for whole-genome association and population-based linkage analysis.** *American Journal of Human Genetics* 2007, **81(3):**559-575.
17. Hedenfalk I, Duggan D, Chen YD, Radmacher M, Bittner M, Simon R, Meltzer P, Gusterson B, Esteller M, Kallioniemi OP, *et al.*: **Gene-Expression Profiles in Hereditary Breast Cancer.** *N Engl J Med* 2001, **344:**539-548.
18. Fung HC, Scholz S, Matarin M, Simón-Sánchez J, Hernandez D, Britton A, Gibbs JR, Langefeld C, Stiegert ML, *et al.*: **Genome-wide genotyping in Parkinson's disease and neurologically normal controls: first stage analysis and public release of data.** *Lancet Neurology* 2006, **5(11):**911-916.
19. Benjamini Y, Hochberg Y: **Controlling the false discovery rate: a practical and powerful approach to multiple testing.** *J R Stat Soc Ser B* 1995, **57:**289-300.
20. Diskin SJ, Eck T, Greshock J, Mosse YP, Naylor T, Christian J, Stoeckert J, Weber BL, Maris JM, Grant GR: **STAC: A method for testing the significance of DNA copy-number aberrations across multiple array-CGH experiments.** *Genome Research* 2006, **16:**1149-1158.
21. R Development Core Team: *R: A Language and Environment for Statistical Computing* 2005 [http://www.R-project.org]. R Foundation for Statistical Computing, Vienna, Austria [ISBN 3-900051-07-0]
22. Brown LD, Cai TT, DasGupta A: **Interval Estimation for a Binomial Proportion.** *Statistical Science* 2001, **16:**101-133.
23. Martin ER: **Linkage Disequilibrium and Association Analysis.** In *Genetic Analysis of Complex Disease* 2nd edition. Edited by: Haines JL, Pericak-Vance M. New York, NY, USA: Wiley; 2006:329-354.
24. Foster DP, Stine RA: **Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy.** *Journal of the American Statistical Association* 2004, **99:**303-313.
25. Subramaniana A, Tamayoa P, Moothaa VK, Mukherjeed S, Eberta BL, Gillettea MA, Paulovichg A, Pomeroyh SL, Goluba TR, Landera ES, Mesirova JP: **Family-based designs in the age of large-scale gene-association studies.** *Proceedings of National Academy of Sciences* 2005, **102:**15545-15550.
26. Laird NM, Lange C: **Family-based designs in the age of large-scale gene-association studies.** *Nature Reviews Genetics* 2006, **7:**385-394.