

Using ANOVA for gene selection from microarray studies of the nervous system

Paul Pavlidis*

*Department of Biomedical Informatics and Columbia Genome Center, Columbia University, Rm 121J, 1150 St. Nicholas Avenue,
New York, NY 10032, USA*

Accepted 25 April 2003

Abstract

Methods are presented for detecting differential expression using statistical hypothesis testing methods including analysis of variance (ANOVA). Practicalities of experimental design, power, and sample size are discussed. Methods for multiple testing correction and their application are described. Instructions for running typical analyses are given in the R programming environment. R code and the sample data set used to generate the examples are available at <http://microarray.cpmc.columbia.edu/pavlidis/pub/aovmethods/>.

© 2003 Elsevier Science (USA). All rights reserved.

Keywords: Analysis of variance; ANOVA; Gene expression; Microarray; Statistics; Differential expression

1. Introduction

The nervous system poses special challenges for the application of microarray technologies. The heterogeneity of nervous system tissue means that even large changes in gene expression may be obscured if they occur only in a small number of cells or a single cell type. In addition, many RNA messages of interest are scarce in the nervous system: one would like to accurately measure RNA transcripts for rare channel subunits, for example. A third issue is that often the anatomical region of interest is very small and obtaining sufficient RNA to assay expression can be problematic. Finally, in contrast to the study of cancer, inflammation, or the cell cycle, where numerous genes can be expected to show dramatic changes in expression, in the nervous system more subtle changes are often expected and of interest. While these issues are often relevant to the study of other tissues, and are not major issues for all nervous system microarray studies, it is safe to say that microarray technology is pushed to its limits by the demands of neuroscience. These issues motivate the

need for sensitive statistically driven methods for data analysis.

The approach I describe is based on methods of hypothesis testing, such as analysis of variance (ANOVA), that have been in use for many years [1]. In our implementation, each assayed gene is tested independently statistically for a difference in expression between experimental group [2]. The output of the analysis is a probability (p value) that a difference in expression could have been observed by chance. The methods rely on having an appropriate study design and sufficient replication. A common alternative to a statistical approach is to perform only one or two microarrays for each condition and rely on non-statistically motivated criteria such as “2-fold-change” to select genes. This approach has the advantage of using very few microarrays (saving money and materials in the short term), but its sensitivity and reliability except for the largest and most robust changes in expression are questionable [3]. The “fold-change” approach is appropriate if the microarray study is used purely as a preliminary or coarse screen, and one is committed to using detailed follow-up studies to sift through the results. For studies of the nervous system the fold-change approach can fail completely, as in some experiments very few if any genes

* Fax: 1-212-851-5149.

E-mail address: pp175@columbia.edu.

make a “2-fold-change” cutoff due to the issues raised above. A statistical analysis can reveal genes which show small but highly significant changes in expression; in the nervous system such differences may be extremely important biologically.

2. Description of the method

2.1. Replication, power, and the problem of multiple testing

Intuitively the more samples are measured, the more confident one can be about the results of an experiment. This is because increasing sample size increases both sensitivity and specificity, which are, respectively, the ability to detect real changes in expression and to correctly reject cases which have no difference. In the approach I describe, each gene is statistically tested independently. An important issue which arises is the effect of multiple testing on power. Each time we statistically test a gene with a statistical test, we incur the risk of a false positive. It is standard practice in bio-statistics to use a p value threshold of 0.05 for the decision as to whether a difference is significant or not. This p value is the probability of getting a false positive result, so on average we would expect to get a false positive result about once every 20 times the test is used ($1/0.05$). For an experiment with 10,000 genes, this translates to 500 false positives ($0.05 \times 10,000$ tests). This calculation is conservative, as it ignores correlations between genes, but it is a useful guideline. It shows that a much smaller p value threshold than 0.05 is needed to keep the number of false positives at an acceptable level. The need for multiple test correction reinforces the need to perform reasonably large numbers of replicates. The issues of study design, replication, and multiple test correction are described more fully in the next sections.

2.2. Factorial designs

A *factor* is a variable which is in some way under the control of the experimenter, such as a drug treatment. The possible values for each factor are the *levels*. It is often of interest to examine multiple factors simultaneously. Perhaps we already know that mouse strain A has a lower physiological response to a drug than a different strain B and we wish to identify genes which respond to the drug in strain B more (or less) than in strain A. This type effect is an *interaction*—the effect of the drug interacts with strain. In this case, the factors are *treatment* (with levels *drug* and *vehicle*) and *strain* (levels A and B). An appropriate experimental design is to test both the drug and the vehicle in both strains A and B (Fig. 1A). Thus, four *experimental conditions* are to be

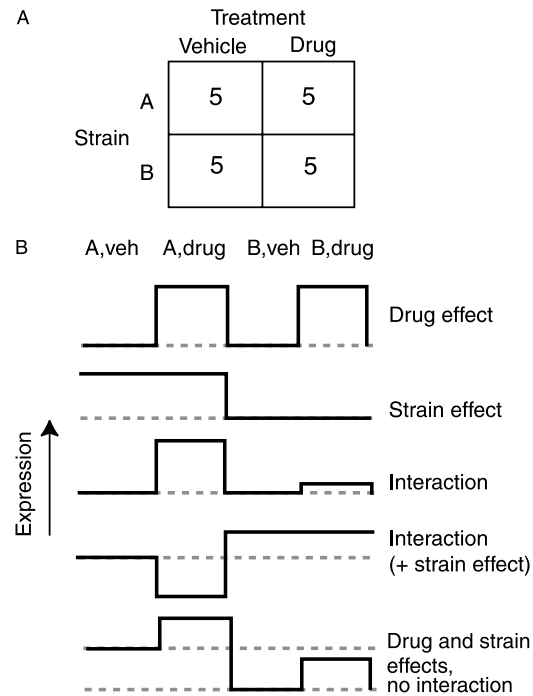


Fig. 1. (A) Complete two-way factorial experimental design. Combinations of experimental conditions are boxes, with the number of replicates [5] shown. (B) Examples of different kinds of effects that can be identified with ANOVA. The grey lines are to assist visualizing “baseline” levels.

tested (vehicle, strain A; vehicle, strain B; drug, strain A; and drug, strain B). This is a classical two-way factorial design. Importantly, this design allows us to also identify genes which are affected only by the drug, or only by strain, which do not show any interaction, as described below. These are called *main effects* because they reflect independent effects of the two main factors. Some examples of gene expression profiles exhibiting different main and interaction effects are given in Fig. 1B.

It is clearly necessary in this kind of design to test all four conditions if interactions are to be studied. Less obviously, it is important to collect equal (or at least approximately equal) numbers of replicates in each of the four groups. Note that each factor can have any number of levels; for example, if one had tested four strains of mice, and three drugs, this would still be a two-way design. A common situation is to have a one-way design with two or more levels. The two-way design can be extended to three-way or even more complex designs; imagine adding gender as a factor to our experiment.

2.3. Replication and sample size

A *biological replicate* is an mRNA sample extracted from an individual sample and run on a single microarray.

In our experience, such replicates are more important than *technical* replicates (running the same labeled mRNA on multiple arrays) or *sample* replicates, where RNA from the same sample is analyzed multiple times. This is because variability between samples tends to be larger than variability from array to array (if this is not true for your microarrays, improvements in experimental technique could yield impressive performance gains; otherwise, technical replicates are in order as well as biological replicates). Thus, if one is studying gene expression in the hippocampus of mice, it is desirable to perform one microarray per hippocampus. Having said this, it is advisable to perform at least some technical replicates in the process of technique validation.

Some other aspects of data collection are worth mentioning briefly. Pooling of samples, so that each microarray represents combined RNA taken from multiple sources, should be avoided if possible because this can hide sources of variability. If pooling is necessary, multiple independent pools should be analyzed rather than replicates of a single pool. Consistency of the methodology used to generate the data is also important, as any technical variability can introduce intolerable artifacts into an experiment. For example, better results are obtained when samples are all processed by the same person, using the same batch of reagents, always interleaving experimental and control samples.

How many replicates are needed? While it is possible to make rough estimates of how many samples are needed to get a pre-specified power, in practice for microarrays this is a complex issue [4,5]. In some cases, pilot studies will be needed to get an idea of how large a sample must be taken. Fewer microarrays might be permissible if large effects are expected and of sole interest. All else being equal, experiments where interaction effects or subtle changes in expression are of interest will need more replicates, and human studies typically need more replicates (individual subjects in one group being considered replicates of the same condition) than laboratory animal or cell line studies, due to the variability of human populations and sample collection methods.

To give some rough guidelines, we start by assuming that at least five replicates per group will be needed and adjust this number up or down (usually up) depending on the demands of the particular study. Thus, in the two-way design described above, we might do 20 microarrays. We have used up to 12 replicates per group in mouse studies and for human studies we plan on doing more. These values are supported by computational studies [4,5]. The cost of this level of replication is justified when one considers the cost of getting no meaningful results from the study if insufficient replicates are collected. If the required number of replicates is not affordable, we consider a simpler experimental design

with more modest (and attainable) goals. Instead of testing the effect of the drug on two mouse strains, we might settle for testing just one, hoping to garner encouragement to complete a more ambitious study based on the resulting data.

3. Analysis of variance

The fundamental idea behind analysis of variance (ANOVA) is that, given an appropriate experimental design, variability in the quantity being measured (gene expression) can be partitioned into various identifiable sources. The assumed sources of variability will include the experimental factors, as well as random noise. ANOVA allows one to examine whether the variability due to a particular experimental factor, or combination of factors, is statistically significant compared to the measured variability due to random sources. Thus, ANOVA can be used to examine differences between groups. In the example already given, we can examine whether the variability between the drug treated vs. the untreated animals is higher than the variability within the groups, and likewise between the strains.

ANOVA and its relatives can accommodate many types of experimental designs. The simplest is a single factor, two-level design, in which case a *t* test can be substituted. For two-dye (ratiometric) arrays, this might mean that only one type of array was run, where the experimental and control groups occupy different channels. In this situation, a one-sample *t* test can be applied. For one-dye oligonucleotide arrays, the same design encompasses two groups of arrays and a two-sample *t* test can be used. The next level of complexity is a single factor, multi-level design (one-way). In the remainder of this paper, I focus on the more complex two-way designs. In a basic two-way ANOVA, the expression level of a gene is expressed as

$$E_{ijk} = \mu + T_i + S_j + (T \cdot S)_{ij} + \varepsilon_{ijk} \quad \begin{cases} i = 1, \dots, n, \\ j = 1, \dots, m, \\ k = 1, 2, \dots, p. \end{cases} \quad (1)$$

Eq. (1) is a linear model of gene expression in replicate *k* of level *i* of factor *T* and level *j* of factor *S* (factors treatment and strain in our example), with *n* and *m* levels, respectively, and *p* replicates per group, μ is the mean expression level of the gene, and ε represents random error. Thus, the expression level *E* can be impacted by *main effects* due to *T*, *S*, and *interactions* between them (*TS*), plus random error. Parameters for this model are fitted to the data for one gene. The extent to which *T*, *S*, and *TS* are non-zero for a gene is then tested statistically, yielding three F statistics, which yield corresponding *p* values. The same model is fitted to the data

for each gene in the data set. Thus for a microarray with 10,000 genes, there will be a total of 30,000 p values calculated in a two-way analysis of variance. The genes can then be ranked by the strength of either of the main effects or the interaction effect. Note that if a gene shows interaction effects, the main effects for that gene are no longer straightforward to interpret [1].

There are a number of assumptions which must be made before ANOVA can be applied; deviation from the assumptions will lead to misleading or inaccurate results. These assumptions include the independence, normality, and uniformity of variance of the errors. Importantly, we assume (or at least hope) that the linear model we have selected is adequate to describe the data. In some situations, there are adjustments that can be made to reduce violations of these assumptions, such as log transformation of the data (for ratiometric arrays, the data are usually log-transformed), and there are well-established methods for diagnosing the quality of the results [1]. However, for microarray data, there are likely to be genes for which the assumptions are valid and others for which they are not. The situation may improve for some genes and worsen for others after correction attempts. The use of non-parametric methods can greatly reduce the reliance on prior assumptions about the data, but also incur a loss of power. In practice, across many genes we have found parametric ANOVA and t tests to give excellent and biologically meaningful results.

4. Multiple test correction

The issue of multiple test correction has been mentioned already and there are multiple ways of correcting for this problem. The simplest is the Bonferroni method, in which each p value is simply multiplied by the number of tests done (and capped at 1.0). The usual $p < 0.05$ is then applied. This method is very conservative because it sets the *family-wise error rate* at 0.05, which is typically lower than is needed: it means that there is only a 0.05 chance of having a *single* false positive result. If 100 genes are selected, we can usually accept a few false positives, so this stringency is not necessary. The Bonferroni method is additionally conservative because of the correlations between genes.

A better way to establish p value thresholds is to pay attention to the number of false positives that are expected at a given uncorrected p value threshold [2]. This idea has been formalized by Benjamini and Hochberg [6] as the false discovery rate (FDR). The idea can best be explained by an example. If there are 100 genes that meet a p value threshold of 0.0001, and there are 10,000 genes tested, we would expect there to be 1 false positive among those 100 genes ($10,000 \times 0.0001 = 1$). The FDR in this case is 0.01. Benjamini and Hochberg [6] give a

simple method for finding the right p value threshold to control the FDR at a pre-specified level. The algorithm is simple enough to be implemented in a Microsoft Excel spreadsheet. For further discussion of FDR and its application to microarray data, see [7].

5. Post hoc testing and template matching

ANOVA identifies genes which show significant group differences; however, it does not identify which groups show the differences. Thus, in a one-way ANOVA with four groups (A, B, C, and D), it is possible for one gene that A is different from B, C, and D, but B, C, and D are not significantly affected, while for another the reverse is true. This motivates the need for post hoc tests, which are used to examine genes at the next level of detail. Post hoc tests are applied only to genes which have already been selected based on ANOVA. The Tukey and Scheffé methods are the commonly used post hoc tests [8].¹ As a more flexible alternative or adjunct to these tests, we use a very simple and generic method we refer to as “template matching” [2]. In this method a template, or profile, of gene expression which is sought is defined by the experimenter and genes which match the template (have similarly shaped profiles) are identified statistically. Our usual implementation of template matching is the same as linear regression, but non-parametric or so-called robust methods can be substituted. The advantage of template matching is that it is very simple, flexible, and can be used in many contexts in addition to post hoc testing. In many cases, template matching is the same as a t test. A disadvantage of template matching, compared to the Tukey or Scheffé tests, is that it does not automatically correct for multiple testing when more than one pattern is tested. The Bonferroni method can be used to conservatively reduce errors due to testing of multiple templates.

Examples of some templates are shown in Fig. 2. Templates can have any shape (and it is only shape that matters). In the context of group comparisons, it makes sense to use templates that relate to the groups, as shown in Figs. 2A–C. In other situations, templates can be crafted based on an experimental response variable (behavioral measurement, etc.), or even use the profile of a particular gene of interest to find other genes with similar patterns (Fig. 2D). The latter application can be thought of as ‘directed clustering.’ Usually we choose to allow a gene to match the template or its reverse; that is, both positive and negative correlations are considered equally interesting.

¹ To my knowledge, at this writing the Tukey and Scheffé methods are available in R only in add-on packages if at all, and as implemented are not very convenient for microarray data analysis.

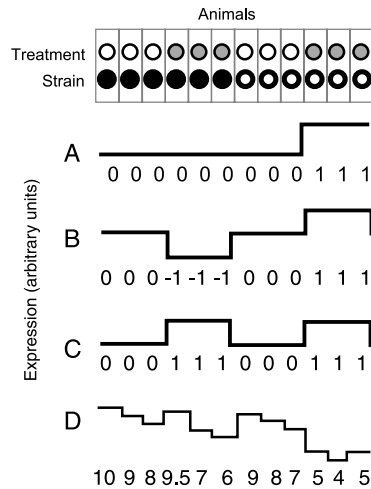


Fig. 2. Template matching. Several templates are shown, along with the numerical representation that is used for the actual analysis. Units are arbitrary. (A–C) Templates which are designed to select specific patterns of expression relevant to the experimental design. These would be designed based on prior knowledge of the experimenter that these patterns are of interest. (D) This template might represent the expression levels for a particular gene of interest, and can be used to scan the data for genes with similar expression profiles.

6. Example: analysis of a complex data set

In this example, we replicate part of the analysis of the data of Sandberg et al. [9] described in Pavlidis and Noble [2], here using the “R” statistical language, which is freely available on the Internet and runs under many platforms [10]. The file `sandberg-sampleddata.txt` contains data for 1000 genes measured using one-channel oligonucleotide arrays (out of the >11,000 in the original data set), and is available on the author’s web site along with all the commands used and additional information. In what follows, some familiarity with R is assumed. Readers who are unfamiliar with R can consult the R web site (<http://cran.r-project.org/index.html>) or any of The numerous books and web sites devoted to R or its cousins, S and S-plus.

This data set contains 24 samples (microarrays), with two mouse strains tested (129 and B6), and six brain regions: amygdala (ag), cerebellum (cb), cortex (cx), entorhinal cortex (ec), hippocampus (hp), and midbrain (mb). Two replicates were performed for each region in each strain [9]. This is a balanced two-way design with high-quality data, though with fewer replicates per group than one would like in order to sensitively detect interaction effects. The factors are strain and region, with levels 129/B6 and ag/cb/cx/ec/hp/mb, respectively. We are interested in identifying genes which are differentially expressed between strains, regions, and those which show interactions between strain and region. An example of the latter would be a gene that is expressed at

high levels in the cerebellum of 129 mice, but at low levels in B6 cerebellum, compared to other regions.

The data file is set up so that each row is the data for a gene, and the samples are arranged in a regular fashion, so that the 129 samples come first, then the B6 samples, and within each set of 12 the replicates for each region are in the same order. There are a single header row and a single column of gene labels. The analysis proceeds as follows (any output by R is omitted for space considerations):

```
sdata<-read.table("sandberg-sampleddata.txt",
  header=T, row.names=1)
strain<-gl(2,12,24,
  label=c("129", "b16"))
region<-gl(6,2,24,
  label=c("ag", "cb", "cx", "ec", "hp", "mb"))
aof<-function(x)
{m<-data.frame(strain,region,x);
anova(aov(x~strain+region+strain*region,m))}
```

These steps reads the data into the variable `sdata`, define the factors `strain` and `region`, and finally define the analysis we will do on each gene as a function `aof`. The *model formula* `x ~ strain+region+strain*region` is equivalent to Eq. (1) for this data set. Note that if the data in the two-way experimental design were unbalanced (unequal numbers of replicates in each group), changing the order of the terms in the model would give somewhat different results; unbalanced designs should be interpreted carefully. If we were only doing a one-way ANOVA on region, the formula would be `x ~ region`. The model `x ~ strain` is equivalent to using the R function `t.test` because there are only two strains.

We then apply the ANOVA analysis to each gene using

```
anovaresults<-apply(sdata, 1, aof)
```

This step can take several minutes to run on all the genes on a microarray. On a modern PC, 1000 genes takes about 30 s. You can get a look at the results of the ANOVA on one gene by using `anovaresults[[i]]`, where *i* is the number of the gene you want to access (where *i* is an integer, $1 \leq i \leq 1000$ for the example). However, examining each result manually for thousands of genes is not practical though it can (and should) be done for individual genes of interest. To just print out all the *p* values in one tab-delimited table more suitable for additional processing, you can use the inelegant but effective:

```
pvalues<-data.frame(lapply(anovaresults,
  function(x){x["Pr(>F)"][1:3,]}))
write.table(t(pvalues),
  file="anova-results.txt",
  quote=F, sep='\t')
```

The resulting file `anova-results.txt` can then be opened in Excel, for example. The first column contains the gene name, followed by *p* values for strain, region, and interaction effects, respectively.

In R, for illustration purposes we can select some genes which have strong ANOVA region effects (say, $p < 0.0001$) but no evidence of interactions ($p > 0.1$) with

```
reg.hi.p<-sort(t(data.frame
  (pvalues[2, pvalues[2,] < 0.0001
  & pvalues[3,] > 0.1])))
```

This selects, in sorted order, 41 genes from the 1000 we started from; the FDR is very low ($<0.01: 0.0001 \times 1000/41$) so we could have used a less stringent p value and still maintained a reasonable FDR (see below). Similar commands can be used to select genes showing strain effects or interaction effects.

7. Template match analysis

The genes selected by ANOVA above all have significant region main effects, but it is not yet clear which region(s) are relevant because the ANOVA finds all types of regional differences. In many cases, the effects are obvious on visual inspection, but to do it systematically (and statistically) we use template matching or another post hoc test. To perform template matching for genes that are expressed differentially in cerebellum (cb) in both strains as compared to all other regions, we get the data for the 41 genes selected for further analysis:

```
reg.hi.pdata <- sdata[row.names(reg.hi.p),]
```

We define an appropriate template as follows:

```
cbtempl<-c(0,0,1,1,0,0,0,0,0,0,0,0,
           0,0,0,1,1,0,0,0,0,0,0,0,0)
```

This is correct because the 3rd and 4th and 15th and 16th samples are from cerebellum, while the rest are from other regions. This template seeks a very simple pattern, where the relative levels of expression are the same in all the cerebellum samples, and similar in all the other samples. However, we generally allow matches to both this pattern and its reverse. More complex or specific patterns are easily specified, but in this case we think of the template as representing an ‘ideal expression pattern’ we are seeking.

In the most general case, we perform a regression of a gene against this template. In R, this can be done easily with `cor.test`, which has the advantage of offering some non-parametric options (e.g., Kendall's Tau). With the default settings, testing the 14th gene, which happens to match the template well:

```
cor.test(t(reg.hi.pdata[l4,]),cbtempl)
```

reports a correlation coefficient of about 0.9, with a p value of less than 10^{-9} . The template match method can be run automatically on multiple genes using similar techniques as demonstrated above:

```
template.match <- function(x, template)
{ k<-cor.test(x, template)k$p.value}
cbtempl.results <- apply(reg.hi.pdata, 1,
  template.match, cbtempl)
write.table(cbtempl.results,
  file="cbtempl-results.txt",
  quote=F, sep='\t')
```

Additional templates can be defined as needed. In [2], a template was designed for each region. Note that in this simple case, the template match method is equivalent to a Student's t test between the samples of one type and the rest. A demonstration of this is useful, as the t test can also be used in situations where only two groups of samples exist and ANOVA methods are not explicitly needed. Applying this to the 14th gene on the list:

```
t.test(t(reg.hi.pdata[l4,region!="cb"]),
      t(reg.hi.pdata[l4,region=="cb"]),
      var.equal=T)
```

Somewhat limited versions of these methods are available in Microsoft Excel's "Analysis Toolpak" (a standard Add-in) as the 'ttest' and 'correl' functions.

A final useful step is to print out data so that it can be used in programs other than R. To print out the data for the 41 selected region effect genes in order of cerebellum-template p value we use:

```
write.table(reg.hi.pdata[order
  (cbtempl.results), ], "
  cbtempl.pdata.txt",
  quote=F, sep='\t')
```

8. False discovery rate analysis

To find the appropriate p value threshold for a given FDR, we can define a function called `bh` (for Benjamini–Hochberg). The code for `bh` is given in Appendix A. The function takes two arguments: `x`, which is a list of p values; and the desired FDR. This list could come from any test. This procedure is also simple to implement in a Microsoft Excel spreadsheet. To use this function on the 1000 region effect ANOVA p values from our sample data, with the FDR controlled at 0.05:

```
regionp <- sort(t(pvalues[2,])) # the p values
  must be sorted in increasing order!
fdr.result <- bh(regionp, 0.05) # reports
  that 192 genes are selected
bhthresh <- cbind(regionp, fdr.result)
  # put the bh results in our table.
write.table(bhthresh, "bhthresh.txt",
  sep = "\t", quote = F) # print to a file.
```

This creates a file called `blhthresh.txt` which has the p value in the first column and an indicator in the second column (1 = meets FDR criterion, 0 = does not). One hundred ninety-two genes are selected at this FDR;

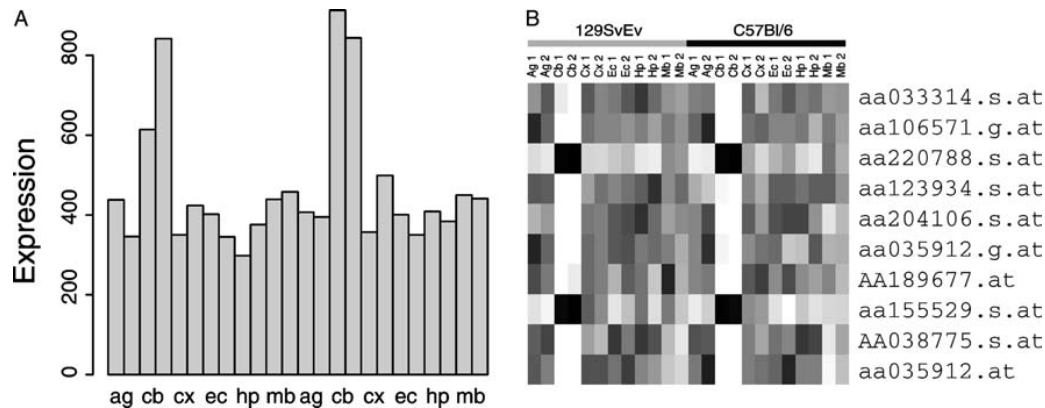


Fig. 3. Visualization of genes selected for differences in cerebellum compared to other regions, based on ANOVA and template matching. (A) Plot of the top probe set from the region ANOVA and cerebellum template analysis of 1000 genes. (B) Visualization of the top 10 genes, created with the help of “matrix2png” [12]. Lighter shades signify higher relative expression levels. The genes are listed in order of increasing p value. Note that two of the genes show under-expression in the cerebellum, as compared to the other regions. See author’s web site for color version.

about 10 of these ($0.05 \times 192 = 9.6$) are expected to be false positives. If this is determined to be too many false positives, *bh* can be run again with a lower FDR. Naturally when running this on the full data set of $\sim 11,000$ genes, the selected threshold will be different. Multiple test correction should be run on the template-match results as well; the total number of genes tested is limited to that selected by ANOVA.

9. Visualizing the results

It is always helpful to take a look at the data graphically. Fig. 3A is an example of a graph made in R for the best probe from the cerebellum analysis. This probe set, aa033314_s_at, corresponds to plexin B2, which functions as a semaphorin receptor in neural pathfinding [11]. It is interesting to note that this gene barely, if at all, meets a “two-fold-change” cutoff, though the regional difference is highly statistically significant. Finally, “colorgram” images like those seen in many microarray papers can easily be made using the author’s “matrix2png” [12] (<http://microarray.cpmc.columbia.edu/matrix2png/>). An example is in Fig. 3B, where the genes are ranked by p value. This ranking is

much more relevant in the current context than an ordering derived from clustering.

10. Concluding remarks

The use of classical statistical methods such as ANOVA is a simple and highly effective alternative to methods such as “fold-change.” The proper use of such methods incurs some cost in the need to carefully design studies and collect sufficiently large samples, but increases sensitivity and specificity. Combined with appropriate correction for multiple testing, statistically based methods allow researchers to control errors in a meaningful way, increase confidence in interpretation, and save effort in follow-up experiments. Importantly, statistically based methods can be used to detect small changes in expression which are of interest in the nervous system.

Acknowledgment

The author’s work is supported in part by NIH Grant MH60970.

Appendix A

The following is a routine for finding a p value threshold that maintains the FDR at a given level, according to the algorithm given by [6]. Note that this function is very stripped-down and does absolutely no checking for errors in the input. The input vector of p values x must already be sorted in increasing order.

```
bh <- function(x, fdr) {thresh <- F; crit <- 0; len <- length(x)
  answer <- array(len); first <- T; for(i in c(len:0)) {crit <- fdr*i/len;
  if (x[i] < crit || thresh == T) {answer[i] <- T; thresh <- T
  if (first) {cat(i, "genes selected at FDR=", fdr, "\n")
  first = F;}} elset {answer[i] <- F}} answer}
```

References

- [1] H. Sahai, M.I. Agell, *The Analysis of Variance*, Birkhauser, Boston, 2000.
- [2] P. Pavlidis, W.S. Noble, *Genome Biol.* 2 (2001).
- [3] V.G. Tusher, R. Tibshirani, G. Chu, *Proc. Natl. Acad. Sci. USA* 98 (2001) 5116–5121.
- [4] A. Zien, J. Fluck, R. Zimmer, T. Lengauer, in: *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology*, 2002.
- [5] P. Pavlidis, Q. Li, W.S. Noble, *Bioinformatics*, in press.
- [6] Y. Benjamini, Y. Hochberg, *J. R. Stat. Soc. B* 57 (1995) 289–300.
- [7] A. Reiner, D. Yekutieli, Y. Benjamini, *Bioinformatics* 19 (2003) 368–375.
- [8] J.H. Zar, *Biostatistical Analysis*, Prentice Hall, Upper Saddle River, NJ, 1999.
- [9] R. Sandberg, R. Yasuda, D.G. Pankratz, T.A. Carter, J.A. Del Rio, L. Wodicka, M. Mayford, D.J. Lockhart, C. Barlow, *Proc. Natl. Acad. Sci. USA* 97 (2000) 11038–11043.
- [10] R.R.G. Ihaka, *J. Comput. Graph. Stat.* 5 (1996) 299–314.
- [11] L. Tamagnone, S. Artigiani, H. Chen, Z. He, G.I. Ming, H. Song, A. Chedotal, M.L. Winberg, C.S. Goodman, M. Poo, M. Tessier-Lavigne, P.M. Comoglio, *Cell* 99 (1999) 71–80.
- [12] P. Pavlidis, W.S. Noble, *Bioinformatics* 19 (2003) 295–296.