MACAT - MicroArray Chromosome Analysis Tool

Joern Toedling, Sebastian Schmeier,Matthias Heinig,
Benjamin Georgi, and Stefan Roepcke

Contents

1 Introduction

This project aims at linking the term *differential gene expression* to the chromosomal localization of genes.
Microarray data analysts have defined tumor subtypes by specific gene expression profiles, consisting of genes that show differential expression between subtypes (see, e.g., [1]). However, tumor subtypes have also been characterized by phenomena involving large chromosomal regions. For instance, Christiansen et al. [2] report on a subtype of acute myeloid leukemia, showing mutations in the *AML1* gene on chromosome 21 along with deletions or loss of chromosome arm 7q. A natural approach to bridging the gap between these two paradigms is to link scoring for differential gene expression to the chromosomal localization of genes. Tumor subtypes can be defined by differential expression patterns affecting sizeable regions of certain chromosomes.
In the following, we propose a statistical approach for identifying significantly differentially expressed regions on the chromosomes based on a regularized t-statistic (see section 2.2). We address the problem of interpolating the scores between genes by applying kernel functions (see section 2.3). In order to evaluate the significance of scores we conduct permutation experiments (section 2.4). An integral part of the project is the visualization of the results in order to provide convenient access to the statistical analysis without requiring a profound mathematical background (section 2.5).

The package is implemented in the R statistical programming language (*www.r-project.org*). It requires functionalities provided by the Bioconductor package [3], which is a collection of R-libraries dealing with various aspects of the analysis of biological data including normalization, assessing background information on genes, and visualization of data.
We apply our method on a publicly available data set of acute lymphoblastic leukemia (ALL), described in [1]. This data set consists of 327 tumor samples subdivided into 10 classes. In order to investigate chromosomal phenomena defining one tumor class, we consider the expression levels of that class versus all the other subclasses.

2 Methods

2.1 Data Preprocessing

We assume normalized expression data, which can be provided already as a matrix in R or in form of a delimited text file. In the preprocessing step the expression data is integrated with gene location data for the given microarray into one common data format. For this *macat* provides the `preprocessedLoader` or its convenience-wrapper `buildMACAT`, which employ various Bioconductor [3] functions.
The resulting data format is a list containing:

- Gene identifier

- Gene location (chromosome, strand, coordinate)
- Sample labels, denoting for instance tumor (sub)types
- Expression levels as a matrix
- An identifier for the type of microarray, used in the experiments

Currently *macat* is limited to commercial Affymetrix® oligonucleotide microarrays.

2.2 Scoring of Differential Gene Expression

For each gene we compute a statistic denoting the degree of differential expression between two given groups of samples. In the context of the leukemia data set the two groups of samples are given by one tumor class in the first group and the remaining nine classes forming the second. The statistic is the regularized t-score introduced in [4]. This so-called *relative difference* is defined as

$$d(i) = \frac{\overline{x}_A(i) - \overline{x}_B(i)}{s(i) + s_0}$$

where $\overline{x}_A(i)$ and $\overline{x}_B(i)$ are the mean expression levels of gene i in group A and B respectively, $s(i)$ is the pooled standard deviation of the expression values of gene i, and s_0 is constant for all genes. In essence, $d(i)$ is Student's t-statistic augmented by a fudge factor s_0 in the denominator, which is introduced to prevent a high statistic for genes with a very low standard deviation $s(i)$. We set s_0 to the median over all gene standard deviations $s(i)$, analogous to [5].
Since the null-distribution of these regularized t-scores is conceptually different from a known t-distribution, we have to approximate it by random permutations (for details see section 2.4).
However, *macat* also allows to compute Student's classical t-statistic for each gene and assess significance based on the underlying t-distribution.

2.3 Smoothing Kernels

On each chromosome only a limited number of genes can be measured with a microarray. The distance in base pairs between measured genes also varies greatly. Since we want to compute differential expression statistics for larger chromosomal regions, we need a method to interpolate scores between measured values. This interpolation, however, does not aim to assign statistics to non-coding regions, but to provide a smooth estimate of differential expression over large chromosomal regions.
Formally this can be seen as a regression problem, $y = f(x)$, with y being the t-score and x being one base pair position on the chromosome. (In the following, x will often be called *chromosomal coordinate.*) Score y_j is to be estimated for a large number of coordinates x_j, when $f(x)$ is known only for few observations x_i. Kernel methods achieve *smooth* estimates of the function $f(x)$ by fitting different models at each query

point x_0 using only observations close to x_0 [6].
Three different kernel approaches will be discussed here:

- k-Nearest Neighbor: For every chromosomal coordinate compute the average of the k nearest genes.
- Radial basis function (rbf): For every chromosomal coordinate compute the average over all genes weighted by distance as explained in detail below.
- Base-Pair-Distance Kernel: Similar to the k-Nearest-Neighbors, but using this kernel the average is taken over all genes within a certain radius of the position, whose value has to be determined.

The kernel approaches presented above can be expressed as weighted sums of scores. Thus, we use a matrix multiplication as the basic framework for the computations, where we multiply our score matrix with the so called kernel matrix.

Let E be the score matrix, with rows corresponding to n genes and columns corresponding to m samples. For each gene g the chromosomal coordinate $geneLocation_g$ is known. Further, let K be the kernel matrix. Let us assume that we want to interpolate the scores at s genomic locations that are stored in a vector we call $\vec{s}$. So the kernel matrix has the dimensions $(n \times s)$. One entry of $K(g, l)$ represents the weight that the gene g has at the location l. So the product

$$E^T K = S$$

gives the smoothed matrix S of dimension $(m \times s)$, where one entry $S(i, l)$ represents the interpolation of the score of sample i at location $\vec{s}_l$.

The smoothed matrix S is the product of the score matrix E times the kernel matrix K. The weights in K depend only on the location of the genes relative to the steps, for which interpolated values are to be computed.
For the three kernels listed above the kernel functions are:

- k-nearest neighbors

$$K_{kNN}(k, g, l) = \begin{cases} 1 & \text{if } g \text{ is one of the } k \text{ nearest neighbors of } l \\ 0 & \text{else} \end{cases}$$

- radial basis function

$$K_{rbf}(\gamma, g, l) = \exp(-\gamma \cdot \|geneLocation_g - \vec{s}_l\|^2)$$

- base-pair distance

$$K_{bpd}(d, g, l) = \begin{cases} 1 & \text{if } geneLocation_g \text{ is within radius } d \text{ of } l \\ 0 & \text{else} \end{cases}$$

The free parameters (k, γ, d) of the kernel functions determine the degree of smoothing. Take for instance the kNN kernel: the smaller k gets, the fewer genes are averaged and in the extreme case interpolations take only the value of the next gene ($k = 1$). In the case of the base-pair distance kernel with a very small distance d, you will see spikes at the locations, where genes are located, and zero elsewhere. For very large distances the smoothing will remove all spikes and the value for each position is roughly the same.
More formally, kernel parameters define the width of the window, over which values are used to estimate the regression function. Changing the window width is a *bias-variance tradeoff* situation. If the window around x_0 is narrow, only a small number of observations will be used to estimate $f(x_0)$. In this case the estimate will have a relatively large variance but only a small bias. However, if the window is wide, a large number of observations will be used for the estimate. This estimate tends to show a low variance but comes with a high bias, since now observations x_i far from x_0 are used as well and we still assume that $f(x_i)$ is close to $f(x_0)$ [6].
This shows that a good choice of the kernel parameters is very important. For the three kernel functions presented above, we fit the parameters from the data:

- kNN: The number of genes on the chromosome is determined and k is set to cover approximately 10% of the genes.
- rbf: The width of the window is controlled by the parameter γ. We require that for each position l, for which we interpolate the score, there should be at least two genes within the window for averaging. If both of these genes are located equally far from l, both will receive a weight of $1/2$, thus resulting in a simple average over these two genes' scores. To allow for these weights, the maximal gene distance (max) between any two neighboring genes on the chromosome has to be determined. Then γ can be computed as:

$$\gamma = \frac{\ln 2}{(max/2)^2}$$

- base-pair distance: as with the rbf kernel, we require that the average is computed at least over two genes. So the parameter d, which describes the radius, within which genes are to be used to compute the average, is set to max, the maximal gene distance between any two genes on the chromosome.

These biologically motivated heuristics for choosing the kernel parameters provide good smoothing results on test data without any overfitting of estimated scores to the observed values.
Nevertheless, we recommend to determine the optimal parameter settings by cross-validation [7], for which *macat* provides the function `evaluateParameters`. Hereby, one part of the genes is left out, the interpolation is computed based on the remaining genes only, and the quality of interpolation is assessed by the residual sum of squares between the scores of the left-out genes and their interpolated values. The above mentioned biologically heuristics, by default, provide a starting point for a grid-search over multiples of these parameter settings. The optimal parameter setting is

the one with the minimal residual sum of squares. For instance, in Figure 1, the result of an evaluation of the parameter k from the k-nearest-neighbor kernel is depicted. From this evaluation, the optimal k seems to be approximately 30.

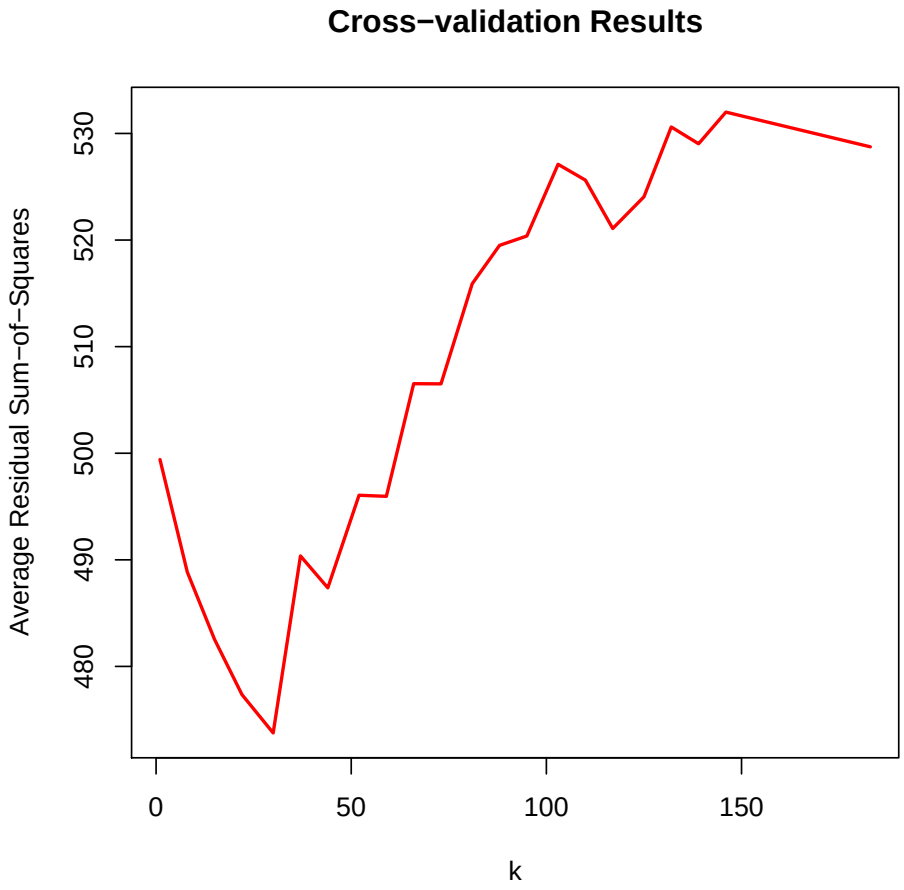

Figure 1: Result plot for evaluating kNN parameter settings

2.4 Statistical Evaluation

To judge the significance of differential gene expression, we propose to investigate random permutations of the class labels. To obtain a reliable simulation of the empirical distribution, we suggest to choose at least $B \geq 1000$ permutations, preferably more. For each of these permutations the (regularized) t-statistic is recomputed for each gene. Thus, for each gene we obtain B permutation statistics and consequently an *empirical p-value*, denoting the proportion of the B permutation statistics being greater or equal than the actual statistic that is based on the true class labels. The permutation statistics also provide upper and lower significance borders, which are

smoothed using the same kernel function as for the original statistics.

Optionally, to judge the significance of differential expression over whole chromosomal regions, one could instead investigate permutations of the ordering of genes on chromosomes. For each of these permutations the smoothing of gene-specific scores is recalculated. This way, one obtains a null-distribution for scores over chromosomal regions. Given this null-distribution, it is possible to define confidence intervals for scores from random sample groupings and assess significance of scores observed in a relevant sample grouping.
However, the standard procedure in *macat* is to permute the class labels, which is considerably faster and yields results that are easier to validate and interpret statistically.
This approach is implemented in the *macat*-function `evalScoring`, which can be seen as the core function of *macat*. See appendix A for an exemplary use of the function and its arguments.
One important issue is that, when analyzing many chromosomes and classes, one might obtain statistically significant results by chance (*multiple- testing problem*). In the given setting, classical procedures correcting for multiple testing, such as the *Bonferroni* correction, cannot be applied. We advise the user to be aware of the problem and to validate results by alternative methods.

2.5 Visualization

In order to facilitate a better understanding of both the data and the statistical analysis, it is helpful to employ meaningful and concise visualizations. *macat* includes plotting functionality for several questions of interest.

- Plotting raw and kernelized expression levels versus coordinates of genes on one chromosome.
- Visualize raw and kernelized statistical scores versus coordinates of genes on one chromosome.
- Emphasis of interesting chromosomal regions with listing of relevant genes.

Chromosomal regions showing significant differential expression can be visualized by plotting the result of the `evalScoring` function (see figure 2). Hereby scores for genes (black dots), the sliding average of the 0.025 and 0.975 quantiles of the permuted scores (grey lines), the sliding average permuted scores (red line), and highlighted regions (yellow dotted), where the score exceeds the quantiles, are plotted along one chromosome. The yellow regions are deemed interesting, showing significant over- or under-expression according to the underlying statistic. The plot region ranges from zero to the length of the respective chromosome.

One can generate an HTML-page (see figure 3) on-the-fly by setting the argument `output` in the function `plot.MACATevalScoring` to "html". The generated HTML-page provides information about genes located within the highlighted chromosomal

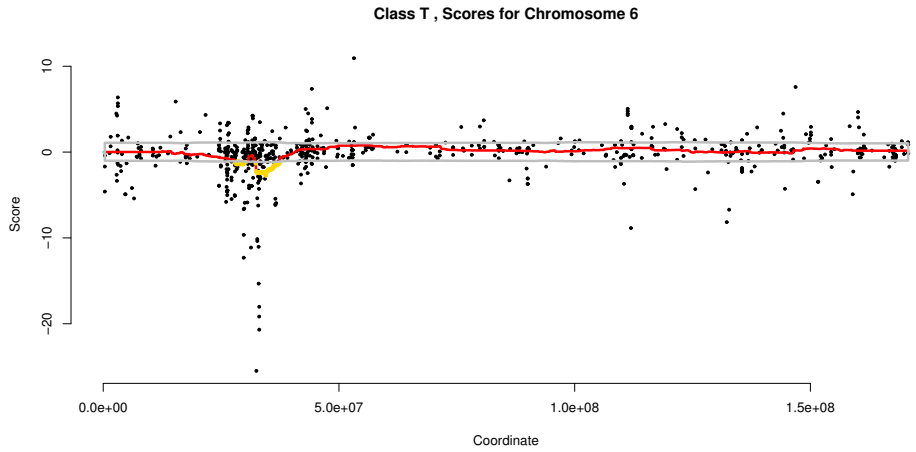

Figure 2: Example plot of plot.MACATevalScoring

regions. For each gene some annotation, a click-able LocusID linked to the NCBI web site, and the empirical p-value is provided.

3 Results

This section describes the results, we obtain from an exemplary analysis on "T- vs. B-lymphocyte ALL".
First the regularized t-score [5] is computed as described in section 2.2. To include information about the distances between the genes, we used the radial basis kernel for smoothing with the default parameters (see section 2.3). We have investigated regions of chromosome six for significant differential expression. Figure 3 shows that there is a region on the p-arm of chromosome six that is significantly under-expressed. The genes within that region comprise also the well known MHC class II genes that are known to be expressed by B-lymphocytes, but not by T-lymphocytes. Other genes in this region remain to be investigated in more detail.
Apart from this region the data reveals no other significant differences in gene expression between T- and B-lymphocyte ALL on chromosome six.

In table 1, we show a list of genes found to be located in significantly differentially expressed regions on different chromosomes when analyzing different subtypes of leukemia versus all other subtypes.

4 Discussion and Outlook

As described in the previous section, applying our method, we detected a chromosomal region as significantly differentially expressed, which is in agreement with biological

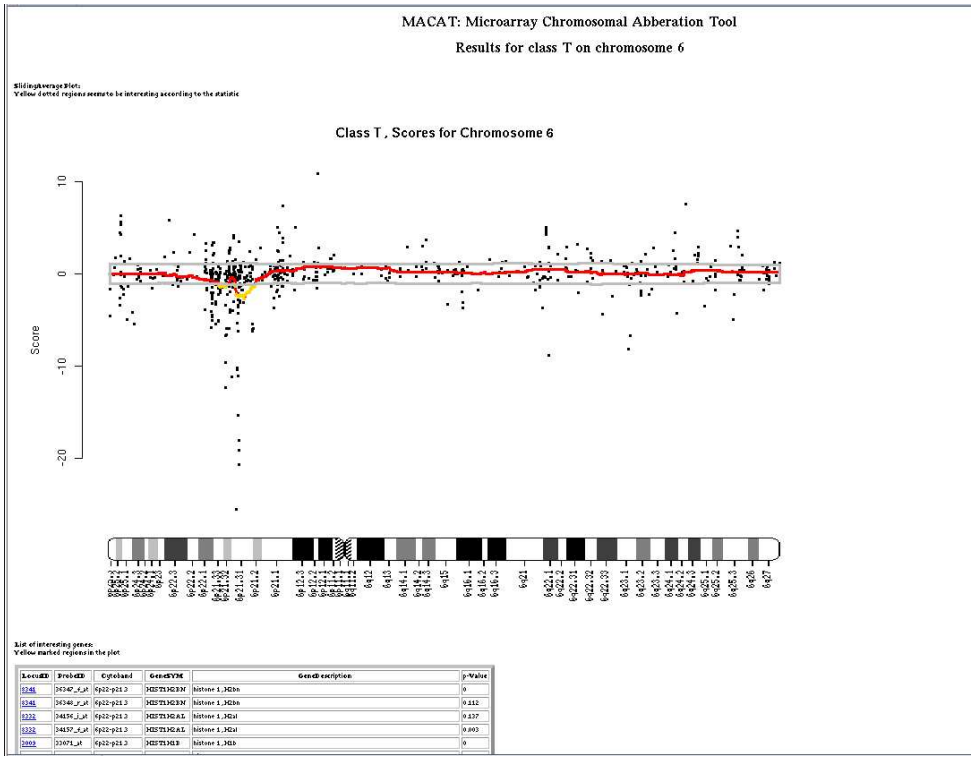

Figure 3: Example for a generated HTML-page

Class	Chrom	Cytoband	LocusID	OMIM Annotation
MLL	8	8q24.12	4609	"Alitalo et al. [8] found that the MYC gene, which is involved by *translocation* in the generation of *Burkitt lymphoma*, is amplified, resulting in homogeneously staining chromosomal regions (HSRs) in a human neuroendocrine tumor cell line derived from a *colon cancer*. The HSR resided on a distorted X chromosome; amplification of MYC had been accompanied by translocation of the gene from its normal position on 8q24."
MLL	9	9p22.3	6595	"Mammalian SWI/SNF complexes are ATP-dependent chromatin remodeling enzymes that have been implicated in the regulation of gene expression, cell cycle control, and *oncogenesis*."
MLL	9	9p24.3	23189	" The suggested role for this protein is in tumorogenesis of renal cell *carcinoma*."
MLL	11	11p13	7490	"Mutations in this gene can be associated with the development of *Wilms tumors* in the kidney or with abnormalities of the genitourinary tract."
E2A	1	1q23	5087	"Wiemels et al.[9] sequenced the genomic fusion between the PBX1 and E2A genes in 22 pre-B acute lymphoblastic leukemias and 2 cell lines. Kamps et al. [10] discussed the chimeric genes created by the human t(1;19) translocation in pre-B-cell acute lymphoblastic leukemias. The authors cloned 2 different E2A-PBX1 fusion transcripts and showed that NIH-3T3 cells transfected with cDNAs encoding the fusion proteins were able to cause malignant tumors in nude mice."
E2A	9	9p24.3	23189	"By RT-PCR and Western blot analysis, Sarkar et al. [11] demonstrated that KANK expression was suppressed in most renal tumors and in kidney tumor cell lines due to methylation at CpG sites in the gene."

Table 1: Example genes with OMIM annotation [12] detected by MACAT to be located in differentially expressed regions. 'Class' denotes the analyzed tumor subtype.

knowledge of the two sample classes involved. This gives an indication that the chromosomal regions annotated as being significant by the method are indeed biologically meaningful. Many of the genes found by MACAT (see table 1) are known oncogenes or are at least associated with oncogenesis.

This fits well with our expectation, since it appears reasonable that different tumor subtypes can be characterized on the molecular level by different expression levels for genes relevant to oncogenesis.

One point of interest for future research would be the application of *macat* on other publicly available data sets and contrasting the results with relevant biological expert knowledge to evaluate performance. This way, one could get an clearer impression of the extend of possible applications.

Another point would be the exploration of different approaches for obtaining relative gene expression values for groups of samples. For instance one could use another data set as a reference for samples with "normal" gene expression levels ("normal" denoting samples from patients with a tumor). Of course, due to the noisy nature of microarray data, this approach would have to overcome some structural and technical difficulties to make the measured expression levels comparable. Given that a suitable data set could be found and the compatibility issues resolved, one could examine the characteristic patterns of chromosomal phenomena in tumor subclasses as opposed to normal tissue.

The method, which we have described, can detect significant differential expression for chromosomal regions. However, the reason for the differential expression, be it a mutation, translocation, hypermethylation, loss of heterozygosity, or another event, remains to be investigated.

Thus, results obtained by our method should be verified by suitable experiments. One could think of a customized cDNA or oligo chip that contains all known genes, including fusion-genes, stemming from translocation events, in the regions that were found to be differentially expressed. To validate the borders of found regions, it would be useful to also incorporate genes that neighbor found regions.

Other possibly useful experiments include Real-time PCR to measure the amount of mRNA for one (or few) specific genes, comparing the results to measured expression levels. The nature of the chromosomal event leading to the differential expression can be investigated by methylation array experiments or by fluorescence in situ hybridization.

Nevertheless, we have shown that our method provides a solid baseline for gene-wise experiments.

A Example Session

A.1 The Beginning

In this section, we show an example *macat* session. First of all, one has to include the library and to load the data set provided in the data package *stjudem*.

```
> library(macat)
> loaddatapkg("stjudem")
> data(stjude)
```

We now have a data object called `stjude`.

A.2 First Investigation

Let us have a closer look on the data. The data is already in the right format. It has been pre-processed by the function `preprocessedLoader`.

```
> summary(stjude)
```

```
             Length  Class  Mode
geneName       12637 -none- character
geneLocation   12637 -none- numeric
chromosome     12637 -none- character
expr         4128375 -none- numeric
labels           327 -none- character
chip               1 -none- character
```

We can for example access the first 10 gene names by typing

```
> stjude$geneName[1:10]
```

```
 [1] "34916_s_at" "34917_at"   "34462_at"   "163_at"     "163_at"
 [6] "35219_at"   "31641_s_at" "33300_at"   "33301_g_at" "38950_r_at"
```

The different labels in the data set can be accessed by

```
> unique(stjude$labels)
```

```
 [1] "BCR"        "E2A"        "Hyperdip"   "Hyperdip47" "Hypodip"
 [6] "MLL"        "Normal"     "Pseudodip"  "T"          "TEL"
```

```
> table(stjude$labels)
```

```
       BCR         E2A    Hyperdip Hyperdip47     Hypodip         MLL      Normal
        15          27          64          23           9          20          18
 Pseudodip           T         TEL
        29          43          79
```

There are ten different classes of tumor patients. The next question is how many probe sets are mapped to chromosome 1.

```
> sum(stjude$chromosome==1)
```

```
[1] 1255
```

Now for some visualization. We want to plot the sliding average of the expression values from sample 3 along chromosome 6 with the default rbf-kernel (for details see section 2.5).

```
> plotSliding(stjude, 6, sample=3, kernel=rbf)
```

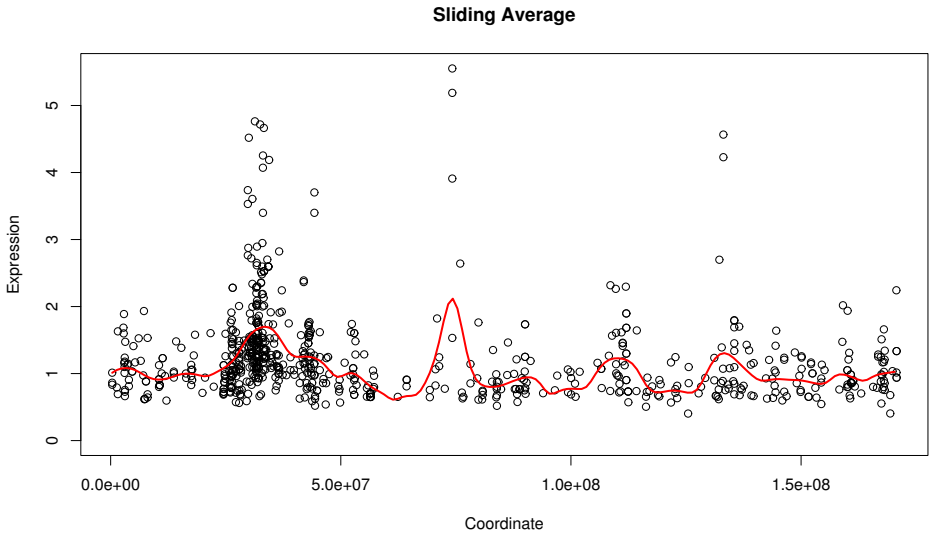

Figure 4: Sliding Average of the expression values of chromosome 6 with rbf-kernel.

See the result in figure 4.

A.3 Deeper Investigation

Next we investigate the data for chromosomal regions showing differential expression. Again, we search chromosome 6 for some specific differences between T-lymphocyte ALL (class "T") and B-lymphocyte ALL (all other classes). Take a look on section 2.2 (Scoring differential expression) for details on the score.
We want to use a kNN-kernel for interpolation between the scores. First, we determine the optimal parameter settings, using the function `evaluateParameters`.

```
> evalkNN6 = evaluateParameters(stjude, class="T", chromosome=6, kernel=kNN,
+                               paramMultipliers=c(0.01,seq(0.1,2.0,0.1),2.5))
```

Let's have a look at the result:

```
> evalkNN6$best

$k
[1] 30

> plot(evalkNN6)
```

The result should look similar to Figure 1. According to this result, the optimal parameter setting for the kNN kernel with our data seems to be $k = 30$. We use this k for the search for significantly differentially expressed chromosome regions. The `evalScoring` is used to build a *MACATevalScoring* object. This may take some time due to the number of permutations.

```
> e1 = evalScoring(stjude, class = "T", chromosome = 6, nperms = 1000,
       kernel = kNN, kernelparams = evalkNN6$best, cross.validate= FALSE)

Investigating 43 samples of class T ...
Compute observed test statistics...
Building permutation matrix...
Compute 1000 permutation test statistics...
250 ...500 ...750 ...1000 ...
Compute empirical p-values...
Compute quantiles of empirical distributions...
Computing sliding values for scores...
Compute sliding values for permutations...
All done.
```

A.4 Collecting Results

Next, we use the plot function `plot.MACATevalScoring` to generate an HTML-page. Therefore, we set the parameter `output` to "html" (see section 2.5).

```
> plot(e1, output="html")
```

See the result in your web browser; it should look similar to Figure 3. Some annotation of genes, which lie in the highlighted regions, is provided.

This ends our short example session. Have fun using *macat*!

Acknowledgments

The authors would like to thank Stefanie Scheid and Florian Markowetz for helpful discussions and proof-reading, Claudio Lottaz for countless hints on building an R-package, and Alexander Schliep, Rainer Spang, Eike Staub, and Martin Vingron for setting up our knowledge and motivation for this project. Our special thanks go to two reviewers, whose comments have led to great improvements of *macat*.

References

[1] E.J. Yeoh et al. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143, March 2002.

[2] D. H. Christiansen, M. K. Andersen, and J. Pedersen-Bjergaard. Mutations of *AML1* are common in therapy-related myelodysplasia following therapy with alkylating agents and are significantly associated with deletion or loss of chromosome arm 7q and with subsequent leukemic transformation. *Blood*, 104(5):1474–1481, 2004.

[3] Robert C. Gentleman, Vincent J. Carey, Douglas J. Bates, Benjamin M. Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Guenther Sawitzki, Colin Smith, Gordon K. Smyth, Luke Tierney, Yee Hwa Yang, , and Jianhua Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[4] V.G. Tusher, R. Tisbhirani, and G. Chu. Significance analysis of microarrays applied to ionizing radiation response. *Proc. Nat. Acad. Sci.*, 98(9):5116–5121, April 2001.

[5] R. Tibshirani et al. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Nat. Acad. Sci.*, 99(10):6567–6572, 2002.

[6] T. Hastie, R. Tibshirani, and J. Friedman *The Elements of Statistical Learning. Chapter 6.* Springer Verlag, New York, 2001.

[7] T. Hastie, R. Tibshirani, and J. Friedman *The Elements of Statistical Learning. Chapter 7.* Springer Verlag, New York, 2001.

[8] K. Alitalo, M. Schwab, C. Lin, H.E. Varmus, and J.M. Bishop. Homogeneously staining chromosomal regions contain amplified copies of an abundantly expressed cellular oncogene (c-myc) in malignant neuroendocrine cells from a human colon carcinoma. *Proc. Nat. Acad. Sci.*, 80:1707–1711, 1983.

[9] J. L. Wiemels, B. C. Leonard, Y. Wang, M. R. Segal, S. P. Hunger, M. T. Smith, V. Crouse, X. Ma, P. A. Buffler, and S. R. Pine. Site-specific translocation and

evidence of postnatal origin of the t(1;19) e2a-pbx1 fusion in childhood acute lymphoblastic leukemia. *Proc. Nat. Acad. Sci.*, 99:15101–15106, 2002.

[10] M. P. Kamps, A. T. Look, and D. Baltimore. The human t(1:19) translocation in pre-b all produces multiple nuclear e2a-pbx1 fusion proteins with differing transforming potentials. *Genes Dev.*, 5:358–368, 1991.

[11] S. Sarkar, B. C. Roy, N. Hatano, T. Aoyagi, K. Gohji, and R. Kiyama. A novel ankyrin repeat-containing gene (kank) located at 9p24 is a growth suppressor of renal cell carcinoma. *J. Biol. Chem.*, 277:36585–36591, 2002.

[12] V.A. McKusick. *Mendelian Inheritance in Man. A Catalog of Human Genes and Genetic Disorders.* Johns Hopkins University Press, Baltimore, 12 edition, 1982.