Description of Logit-t: Detecting Differentially Expressed Genes Using Probe-Level Data

Tobias Guennel

April 24, 2017

Contents

1 Introduction 2
2 What’s new in this version 3
3 Preparing data for use with logitT 3
4 Generate t-statistics and probe set expression summaries 4
5 Using Logit-t in gene expression analysis 5
6 Version history 5
7 Acknowledgements 5
1 Introduction

The Logit-t algorithm (Lemon et al., 2003), also called Median t (Hess et al., 2007), was developed as an alternative to existing statistical methods for identifying differentially expressed genes. Unlike other commonly used algorithms for identifying differentially expressed genes, for example SAM (Tusher et al., 2001), the Logit-t algorithm does not require the calculation of an expression summary for each probe set prior to statistical analysis. Using spike-in datasets, the Logit-t algorithm was compared to statistical testing of probe set expression summaries MAS5 (Hubbell et al., 2002), RMA (Irizarry et al., 2003), and dChip (Li et al., 2001) and was found to have better sensitivity, specificity, and positive predicted value (PPV) (Lemon et al., 2003).

The Logit-t algorithm proceeds by first normalizing probe intensities $i$ within probe set $j$ for array $k$ using the logit-log transformation

$$\logit(y_{ijk}) = \log\left(\frac{y_{ijk} - N_k}{A_k - y_{ijk}}\right).$$

(1)

The parameters $A_k$ and $N_k$ are estimated for each array by

$$A_k = \max_{i,j}(y_{ijk}) + 0.001 \times (\max_{i,j}(y_{ijk}) - \min_{i,j}(y_{ijk}))$$

$$N_k = \min_{i,j}(y_{ijk}) - 0.001 \times (\max_{i,j}(y_{ijk}) - \min_{i,j}(y_{ijk})),$$

representing the maximum probe intensity (saturation) and the minimum probe intensity (background), respectively. Following this transformation, probe level intensities for each array were then standardized using a Z-transformation. Lemon et al. (2003).

For differential expression analysis, Student’s $t$-tests are then performed for each transformed PM probe within a probe set and the Logit-t is defined as the median $t$-statistic among all $t$-statistics for the probe set. The $t$-value cutoff corresponding to $p < 0.01$ using the df of the comparison was used as threshold for making detection calls of differential expression or no differential expression in the original paper (Lemon et al., 2003).

The Logit-t algorithm was originally coded as a stand-alone application in C++ provided by Dr. William J. Lemon. In order to make it available for a broader spectrum of users, the algorithm was implemented in the programming environment R, an open source programming environment (R Development Core Team, 2008) and the most commonly used software package for gene expression data analysis. There are no differences in results between the stand-alone application and its implementation in the R programming environment as most of the model fitting C++ code has been retained. However, the implementation into the R programming environment simplifies the use of the algorithm and the access to its results significantly compared to the stand-alone application. In order to make it available for a broader spectrum of users, the algorithm was implemented in R, an open source programming environment (R Development Core Team, 2008) and the most commonly used software package for gene expression data analysis.
There are no differences in results between the stand-alone application and its implementation in the R programming environment as most of the model fitting C++ code has been retained. However, the implementation into the R programming environment simplifies the use of the algorithm and the access to its results significantly compared to the stand-alone application. The logitT package implements the Logit-t algorithm in the R programming environment, making it available to users of the Bioconductor project.

2 What’s new in this version

This is the first release of this package.

3 Preparing data for use with logitT

The logitT package accepts data from Affymetrix CEL files that have been read into the R programming environment using the affy (Gautier et al., 2004) library and stored in an AffyBatch object. A CEL file contains, among other information, a decimal number for each probe on the chip that corresponds to its intensity. The Bioconductor packages affy, Biobase, and tools (Gentleman et al., 2004) are automatically loaded by logitT to provide functions for reading Affymetrix data files into R. The following steps create an AffyBatch object.

1. Create a directory containing all CEL files relevant to the planned analysis.
2. If using Linux / Unix, start R in that directory.
3. If using the Rgui for Microsoft Windows, make sure your working directory contains the CEL files (use “File -> Change Dir” menu item).
4. Load the library.

   > library(logitT)

5. Read in the data and create an AffyBatch object.

Additional information regarding the ReadAffy function and detailed description of the structure of CEL files can be found in the affy vignette.

4 Generate t-statistics and probe set expression summaries

To demonstrate the use of the logitT R package, the publicly available Affymetrix Latin Square data set (Affymetrix, 2008) was used. The Latin Square design for this data set consists of 14 spiked-in gene groups in 14 experimental groups on HG-U95Av2 arrays. The concentration of the 14 gene groups in the first experiment is 0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024pM. Each subsequent experiment rotates the spike-in concentrations by one group; i.e. experiment 2 begins with 0.25pM and ends at 0pM, on up to experiment 14, which begins with 1024pM and ends with 512pM. Each experiment contains at least 3 replicates.

Two of the 14 experimental groups, i.e. group one and group two with 3 replicates each, are available in the experimental data package SpikeInSubset as the data set spikein95. The package and subsequently the data set can be loaded into the R programming environment using R> library(SpikeInSubset) and R> data(spikein95). To compare the two groups of three arrays, where the first three arrays correspond to group 1 and the latter three to group 2, we define the groupvector as before. To invoke the logitT Affy method, the syntax is:

> library(SpikeInSubset)
> data(spikein95) ## get the example data
> groupvector<-c("A","A","A","B","B","B") ## specify group affiliations
> logitTex<-logitTAffy(spikein95, group=groupvector)

The first few t-statistics are:

> logitTex[1:10]

<table>
<thead>
<tr>
<th>Gene</th>
<th>At</th>
<th>At</th>
<th>F_at</th>
<th>S_at</th>
<th>At</th>
<th>At</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>-0.365796303</td>
<td>0.074983648</td>
<td>0.136804337</td>
<td>0.144461320</td>
<td>-0.021213882</td>
<td>-0.003133944</td>
</tr>
<tr>
<td>1006</td>
<td>0.100833725</td>
<td>0.266406403</td>
<td>0.277591814</td>
<td>0.860347914</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The input arguments for the function call are:

**object** – object of type AffyBatch

**group** – vector describing the group affiliation for each array

The vector describing the group affiliation for each array must be of the same length as the number of CEL files contained in the AffyBatch object. For example, suppose six CEL files are stored in the AffyBatch object where the first three arrays are from one condition and the last three arrays are from a second condition. The object groupvector in the function call would be
Note that the group labels can be defined as desired provided only one unique label is used for identifying a specific group. After invoking the `logitTAffy` function, the result is a named vector containing the t-statistics for each probe set.

5 Using Logit-t in gene expression analysis

As described in the introduction, Student’s t-tests are then performed for each transformed PM probe within a probe set and the Logit-t is defined as the median t-statistic among all t-statistics for the probe set. The t-value cutoff corresponding to \( p < 0.01 \) using the df of the comparison was used as threshold for making detection calls of differential expression or no differential expression in the original paper (Lemon et al., 2003). Lemon et al. suggested to use the df of the comparison to obtain p-values for the calculated t-statistics. In this example, \( n_1 = 3 \) and \( n_2 = 3 \) and therefore \( df=n_1+n_2-2=4 \). Two-sided p-values for each probe set can be obtained using:

```r
> pvals <- (1-pt(abs(logitTex),df=4))*2  ## calculate p-values
> signifgenes<-names(logitTex)[pvals<0.01]  ## find probe sets with p-values
> signifgenes

[1] "1024_at" "1708_at" "32660_at" "36202_at" "36311_at" "38734_at"
```

6 Version history

1.0.0 initial development version

7 Acknowledgements

We thank Sandy Liyanarachchi for providing the original C++ implementation of the Logit-t algorithm as described by Dr William J. Lemon, Dr Ming You, and Dr Sandya Liyanarachchi. This work was partially supported by National Institute of Diabetes and Digestive and Kidney Diseases R01DK069859.

References


