CGHnormaliter Package (Version 1.0.1)

Thomas W. Binsl, Bart P.P. van Houte, Hannes Hettling

March 2, 2010

1 Introduction

This package contains an implementation of the CGHnormaliter strategy for improved normalization of dual-channel array Comparative Genomic Hybridization (aCGH) data displaying many copy number imbalances. The key idea of our method is that temporary exclusion of aberrations from the data allows for a more appropriate calculation of the LOWESS regression curve. As a result, after normalization, the $\log_2$ intensity ratios of the normals will generally be closer to zero and better reflect the biological reality. We coin this normalization strategy 'local-LOWESS' since only a subset of the $\log_2$ ratios is considered in the LOWESS regression.

The strategy can be summarized as follows. Initially the $\log_2$ intensity ratios are segmented using DNAcopy [4]. The segmented data are then given as input to a calling tool named CGHcall [2] to discriminate the normals from gains and losses. These normals are subsequently used for normalization based on LOWESS. These steps are then iterated to refine the normalization. An overview is given in Figure 1. For more detailed information we refer to the publication of our method [3].

2 Data format

The input should be either a `data.frame` or the file name of a tabseparated text file (text files must contain a header). The first four columns should describe the clone and its position on the genome:

1. ID : The unique identifiers of array elements.

2. Chromosome : Chromosome number of each array element.

3. Start : Chromosomal start position in bp of each array element.

4. End : Chromosomal end position in bp of each array element.

The start and end positions must be numeric. The next columns hold the actual data. For each sample in the experiment, there must be two adjacent

step 1   **segmentation**

step 2   **initial calling**

step 3   **local-LOWESS normalization**

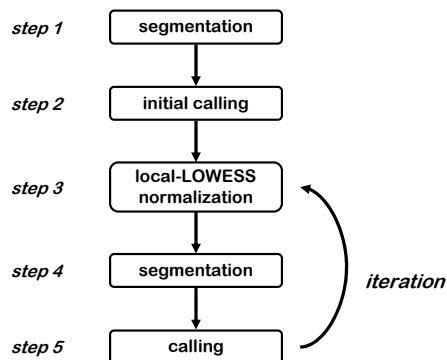step 4   **segmentation**

step 5   **calling**

iteration

Figure 1: Overview of the CGHnormaliter method.

columns with the *test* and *reference* intensities, respectively. All entries must be delimited by tabs, and missing entries must be denoted with *NA* or by an empty value. Below, an example is given of a correctly formatted data file or data.frame containing measurements on 7 clones in 2 samples.

```
ID            Chromosome   Start    End      Case1.test  Case1.ref  Case2.test  Case2.ref
RP11-34P13    1            1        254479   279         294        NA          NA
RP11-379K15   1            95421    244136   1815        2269       2793        3996
RP11-776O18   1            357737   465038   387         349        429         362
RP11-45C18    1            579118   696613   786         734        900         735
RP11-242B5    1            606617   711982   2955        4158       4478        5229
RP13-586C17   1            619355   783174   NA          NA         823         841
RP11-414L23   1            658751   846904   630         937        959         744
```

3 Example

First, we load the example Leukemia dataset [1] which is contained in the CGH-normaliter package:

```
> library(CGHnormaliter)
> data(Leukemia)
```

Next, we run the CGHnormaliter routine on the first three chromosomes of the Leukemia data, with a maximum of 3 iterations:

```
> result <- CGHnormaliter(Leukemia, nchrom = 3)

CGHnormaliter -- Running an initial segmentation and calling
Start data segmentation ..
Analyzing: Sample.1
Analyzing: Sample.2
```

```
Analyzing: Sample.3
Start data calling ...
CGHnormaliter -- Iteration #  1
CGHnormaliter -- Mean normalization shift per sample:
   Case1.test_Case1.ref : 0.4285348
   Case2.test_Case2.ref : 0.4598285
   Case3.test_Case3.ref : 0.3356325
Start data segmentation ..
Analyzing: Sample.1
Analyzing: Sample.2
Analyzing: Sample.3
Start data calling ...
CGHnormaliter -- Iteration #  2
CGHnormaliter -- Mean normalization shift per sample:
   Case1.test_Case1.ref : 0.002363037
   Case2.test_Case2.ref : 0.002599055
   Case3.test_Case3.ref : 0.002958949
CGHnormaliter -- Reached convergence. Running a final segmentation and calling...
Start data segmentation ..
Analyzing: Sample.1
Analyzing: Sample.2
Analyzing: Sample.3
Start data calling ...
CGHnormaliter -- FINISHED
```

Now we can access several fields of the `result` object, for example:

```
> normalized.data <- copynumber(result)
> segmented.data <- segmented(result)
> called.data <- calls(result)
```

To visualize the results per sample, the `plot` function of the `CGHcall` package can be used. In Figure 2 we plot the results for the second sample:

```
> plot(result[, 2])
```

```
Plotting sample Case2.test_Case2.ref
```

Finally, the package provides the function `CGHnormaliter.write.table` to save the normalized data into a tabdelimited plain text file:

```
> CGHnormaliter.write.table(result)
```

```
Saving normalized log2 ratios to file: normalized.txt
```

3

```
Plotting sample Case2.test_Case2.ref
```
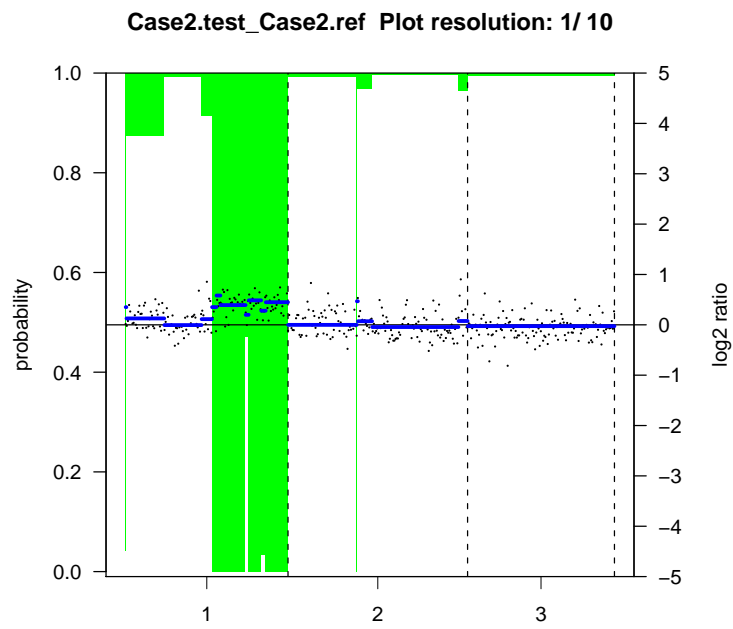
Case2.test_Case2.ref Plot resolution: 1/ 10

Figure 2: Results of the CGHnormaliter normalization for the second Leukemia sample.

References

[1] K. Paulsson, M. Heidenblad, H. Mörse, Å. Borg, T. Fioretos, and B. Johansson. Identification of cryptic aberrations and characterization of translocation breakpoints using array CGH in high hyperdiploid childhood acute lymphoblastic leukemia. *Leukemia*, 20:2002–2007, 2006.

[2] M.A. van de Wiel, K.I. Kim, S.J. Vosse, W.N. van Wieringen, S.M. Wilting, and B. Ylstra. CGHcall: calling aberrations for array CGH tumor profiles. *Bioinformatics*, 23:892–894, 2007.

[3] B.P.P. van Houte, T.W. Binsl, H. Hettling, W. Pirovano, and J. Heringa. CGH-normaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. *BMC Genomics*, 10:401, 2009.

[4] E.S. Venkatraman and A.B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, 23(6):657–663, 2007.