CGHcall: Calling aberrations for array CGH tumor profiles.

Sjoerd Vosse and Mark van de Wiel

February 26, 2008

Department of Pathology
VU University Medical Center

mark.vdwiel@vumc.nl

Contents

1 Overview

CGHcall allows users to make an objective and effective classification of their aCGH data into copy number states (loss, normal, gain or amplification). This document provides an overview on the usage of the CGHcall package. For more detailed information on the algorithm and assumptions we refer to the article (van de Wiel et al., 2007) and its supplementary material. As example data we attached the first five samples of the Wilting dataset (Wilting et al., 2006). After filtering and selecting only the autosomes 4709 datapoints remained.

2 Example

In this section we will use CGHcall to call and visualize the aberrations in the dataset described above. First, we load the package and the data:

```
> library(CGHcall)
> data(Wilting)
```

Next, we apply the `preprocess` function which:

- removes data with unknown or invalid position information.
- shrinks the data to `nchrom` chromosomes.
- removes data with more than `maxmiss` % missing values.
- imputes missing values using `impute.knn` from the package `impute` (Troyanskaya et al., 2001).

```
> cghdata <- preprocess(Wilting, "dataframe", maxmiss = 30, nchrom = 22)
```

```
Changing impute.knn parameter k from 10 to 4 due to small sample size.
Cluster size 3982 broken into 2449 1533
Cluster size 2449 broken into 1472 977
Done cluster 1472
Done cluster 977
Done cluster 2449
Cluster size 1533 broken into 27 1506
Done cluster 27
Cluster size 1506 broken into 1060 446
Done cluster 1060
Done cluster 446
Done cluster 1506
Done cluster 1533
```

To be able to compare profiles they need to be normalized. In this package we provide very basic global median or mode normalization. Of course, other methods can be used outside this package. This function also contains smoothing of outliers as implemented in the DNAcopy package (Venkatraman and Olshen, 2007). Furthermore, when the proportion of tumor cells is not 100% the ratios can be corrected. See the article and the supplementary material for more information on cellularity correction (van de Wiel et al., 2007).

```
> tumor.prop <- c(0.75, 0.9, 0.8, 1, 1)
> norm.cghdata <- normalize(cghdata, type = "dataframe", method = "median",
+     cellularity = tumor.prop, smoothOutliers = TRUE)
```

```
Applying median normalization ...
Smoothing outliers ...
Adjusting for cellularity ...
Cellularity sample 1 :  0.75
Cellularity sample 2 :  0.9
Cellularity sample 3 :  0.8
Cellularity sample 4 :  1
Cellularity sample 5 :  1
```

The next step is segmentation of the data. This package only provides a simple wrapper function that applies the `DNAcopy` algorithm (Venkatraman and Olshen, 2007). Again, other segmentation algorithms may be used. To save time we will limit our analysis to the first two samples from here on.

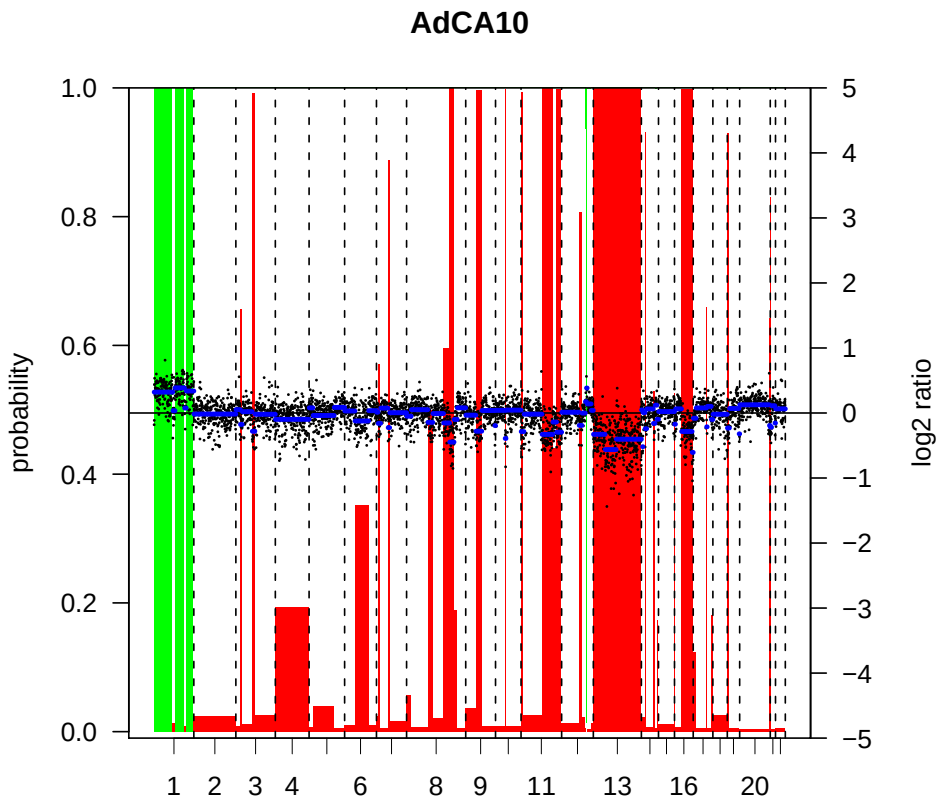
```
> norm.cghdata <- norm.cghdata[, 1:5]
> seg.cghdata <- segmentData(norm.cghdata, type = "dataframe",
+     method = "DNAcopy")

Start data segmentation ..
Analyzing: Sample.1
Analyzing: Sample.2
```

Now that the data have been normalized and segments have been defined, we need to determine which segments should be classified as losses, normal, gains or amplifications.

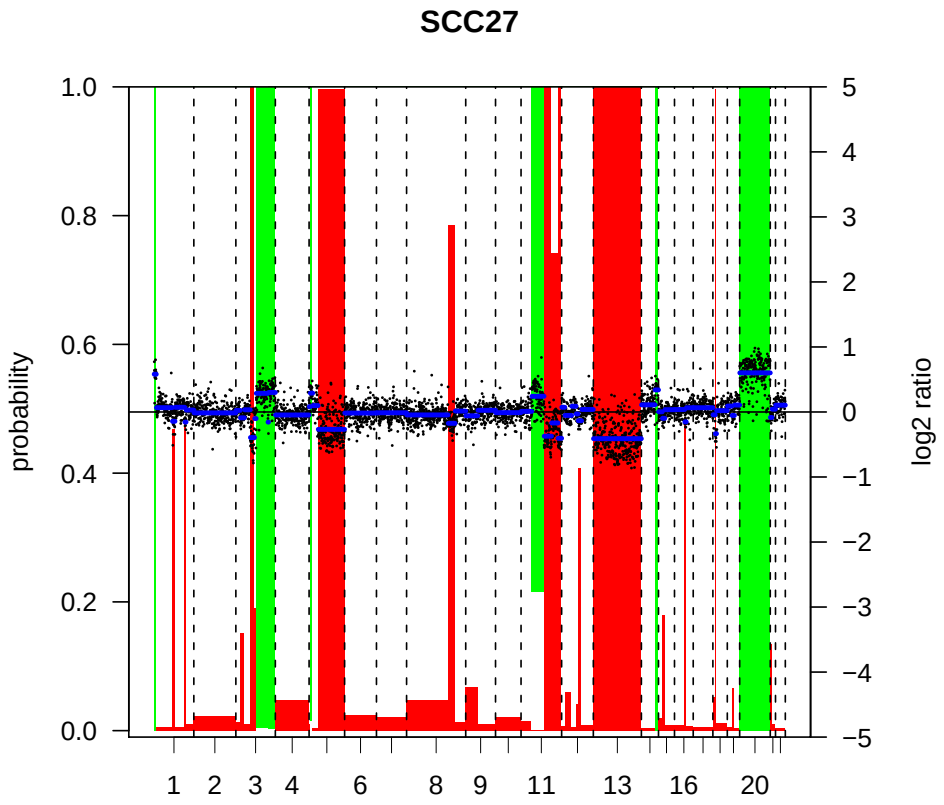
```
> result <- CGHcall(norm.cghdata, seg.cghdata)

EM algorithm started ...
Calling iteration 1 :
 [1]  2.000000e+00 -4.244272e+03 -5.832893e-01 -2.831588e-01  5.078760e-03
 [6]  3.289769e-01  1.157954e+00 -4.310852e-04  1.257185e-01  6.996470e-02
[11]  4.429449e-02  1.000000e-04
Calling iteration 2 :
 [1]  2.000000e+00 -4.243597e+03 -5.762174e-01 -2.760869e-01  7.852018e-03
 [6]  3.283777e-01  1.156755e+00 -2.971913e-04  1.215480e-01  6.854897e-02
[11]  3.598413e-02  1.000000e-04
EM algorithm done ...
FINISHED!
Total time: 1 minutes
```

To visualize the results per profile we use the `plotProfile` function:

```
> plotProfile(result, samples = 1, export = "no")

Plotting sample 1
```

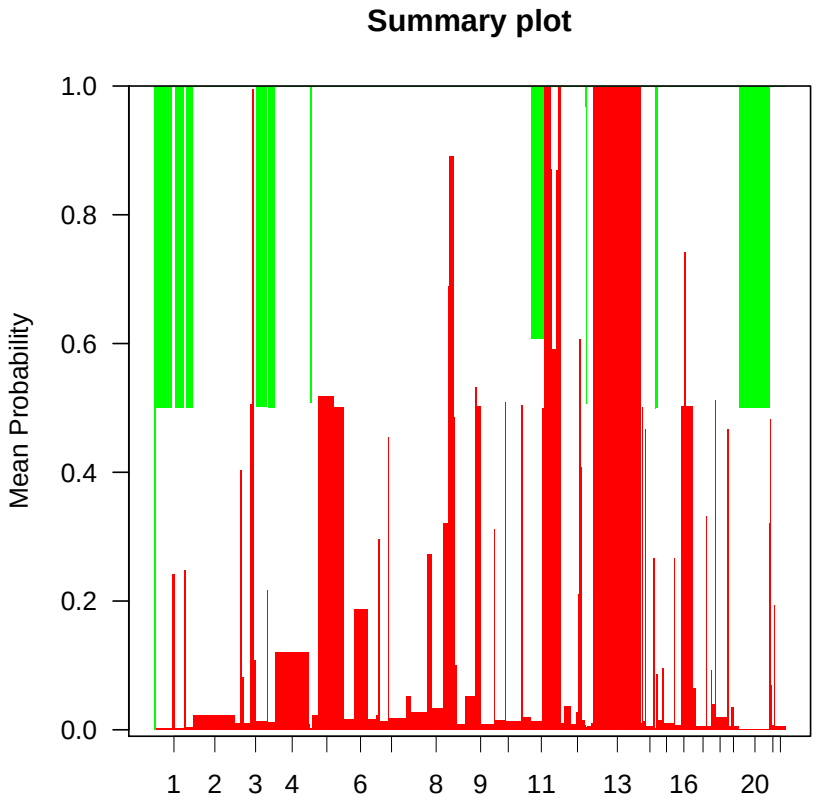


```
> plotProfile(result, samples = 2, export = "no")

Plotting sample 2
```



Alternatively, we can create a summary plot of all the samples:

```
> plotSummary(result, samples = "all", export = "no")

Adding sample 1 to summary plot.
Adding sample 2 to summary plot.
```



References

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525.

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

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

Wilting, S. M., Snijders, P. J. F., Meijer, G. A., Ylstra, B., van den Ijssel, P. R. L. A., Snijders, A. M., Albertson, D. G., Coffa, J., Schouten, J. P., van de Wiel, M. A., Meijer, C. J. L. M., and Steenbergen, R. D. M. (2006). Increased gene copy numbers at chromosome 20q are frequent in both squamous cell carcinomas and adenocarcinomas of the cervix. *J Pathol*, 209:220–230.