CGHcall: Calling aberrations for array CGH tumor profiles.

Sjoerd Vosse and Mark van de Wiel

April 23, 2012

Department of Epidemiology & Biostatistics
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 autosomal 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)
> Wilting <- make_cghRaw(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, maxmiss = 30, nchrom = 22)
```

```
Changing impute.knn parameter k from 10 to 4 due to small sample size.
```

To be able to compare profiles they need to be normalized. In this package we first provide very basic global median or mode normalization. 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).

```
> norm.cghdata <- normalize(cghdata, method = "median", smoothOutliers = TRUE)
```

```
Applying median normalization ...
Smoothing outliers ...
```

The next step is segmentation of the data. This package only provides a wrapper function that applies the `DNAcopy` algorithm (Venkatraman and Olshen, 2007). It provides extra functionality by allowing to undo splits differently for long and short segments, respectively. In the example below short segments are smaller than clen=10 probes, and for such segments undo.splits is effective when segments are less than undo.SD=3 (sd) apart. For long segments a less stringent criterion holds: undo when less than undo.SD/relSDlong = 3/5 (sd) apart. If, for two consecutive segements, one is short and one is long, splits are undone in the same way as for two consecutive short segments. To save time we will limit our analysis to the first two samples from here on.

```
> norm.cghdata <- norm.cghdata[, 1:2]
> seg.cghdata <- segmentData(norm.cghdata, method = "DNAcopy",
+     undo.splits = "sdundo", undo.SD = 3, clen = 10, relSDlong = 5)

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

Post-segmentation normalization allows to better set the zero level after segmentation.

```
> postseg.cghdata <- postsegnormalize(seg.cghdata)
```

Now that the data have been normalized and segments have been defined, we need to determine which segments should be classified as double losses, losses, normal, gains or amplifications. Cellularity correction is now provided WITHIN the calling step (as opposed to some earlier of CGHcall)

```
> tumor.prop <- c(0.75, 0.9)
> result <- CGHcall(postseg.cghdata, nclass = 5, cellularity = tumor.prop)

EM algorithm started ...
[1] "Total number of segments present in the data: 92"
[1] "Number of segments used for fitting the model: 92"
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 397374 10.7     667722 17.9   667722 17.9
Vcells 347059  2.7     786432  6.0   786432  6.0
Calling iteration 1 :
[1] "optim results"
[1] "time: 23"
[1] "minimum: 3757.89563517414"
     j       rl      mudl       musl        mun       mug      mudg      mua
[1,] 2 3738.879 -1.186574 -0.2971907 0.01269801 0.3266167 0.5583549 1.115040
          sddl      sdsl       sdn       sdg      sddg       sda
[1,] 0.7577915 0.0845596 0.0644851 0.1334502 0.1339624 0.5505281
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 398280 10.7     741108 19.8   741108 19.8
Vcells 347911  2.7     786432  6.0   786432  6.0
Calling iteration 2 :
[1] "optim results"
[1] "time: 18"
```

```
[1] "minimum: 3752.04053222869"
     j       rl      mudl       musl        mun       mug      mudg      mua
[1,] 2 3737.682 -1.066646 -0.2934961 0.01665657 0.3287696 0.5620353 1.353487
         sddl       sdsl        sdn       sdg      sddg       sda
[1,] 1.125041 0.08184933 0.05775554 0.1269264 0.1273813 0.5375684
EM algorithm done ...
Computing posterior probabilities for all segments ...
Total time: 1 minutes
```

The result of CGHcall needs to be converted to a call object. This can be a large object for large arrays.

```
> result <- ExpandCGHcall(result, postseg.cghdata)

Adjusting segmented data for cellularity ...
Cellularity sample 1 :  0.75
Cellularity sample 2 :  0.9
Adjusting normalized data for cellularity ...
Cellularity sample 1 :  0.75
Cellularity sample 2 :  0.9
[1] 1
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400213 10.7     741108 19.8   741108 19.8
Vcells 377636  2.9     786432  6.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400223 10.7     741108 19.8   741108 19.8
Vcells 395399  3.1     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400222 10.7     741108 19.8   741108 19.8
Vcells 395398  3.1     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400249 10.7     741108 19.8   741108 19.8
Vcells 423817  3.3     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400563 10.7     741108 19.8   741108 19.8
Vcells 425630  3.3     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400571 10.7     741108 19.8   741108 19.8
Vcells 427409  3.3     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
```

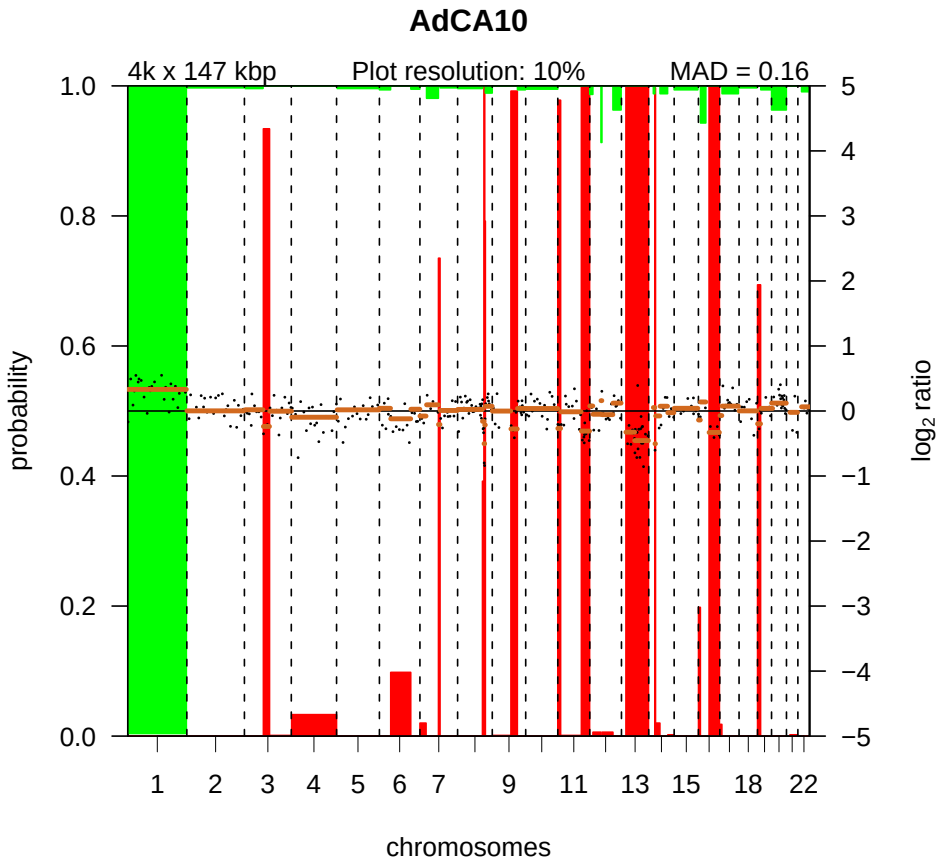
```
Ncells 400579 10.7     741108 19.8   741108 19.8
Vcells 429188  3.3     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400587 10.7     741108 19.8   741108 19.8
Vcells 430967  3.3     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400595 10.7     741108 19.8   741108 19.8
Vcells 432746  3.4     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400599 10.7     741108 19.8   741108 19.8
Vcells 434524  3.4     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 400633 10.7     741108 19.8   741108 19.8
Vcells 455848  3.5     905753  7.0   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401436 10.8     741108 19.8   741108 19.8
Vcells 463320  3.6     905753  7.0   786432  6.0
[1] 2
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401447 10.8     741108 19.8   741108 19.8
Vcells 481104  3.7    1031040  7.9   786432  6.0
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401448 10.8     741108 19.8   741108 19.8
Vcells 481105  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401447 10.8     741108 19.8   741108 19.8
Vcells 481104  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401451 10.8     741108 19.8   741108 19.8
Vcells 484657  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401447 10.8     741108 19.8   741108 19.8
Vcells 481104  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401447 10.8     741108 19.8   741108 19.8
Vcells 477552  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401455 10.8     741108 19.8   741108 19.8
Vcells 479331  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
```

```
Ncells 401463 10.8     741108 19.8   741108 19.8
Vcells 481110  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401471 10.8     741108 19.8   741108 19.8
Vcells 482889  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401475 10.8     741108 19.8   741108 19.8
Vcells 484667  3.7    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 401509 10.8     741108 19.8   741108 19.8
Vcells 505991  3.9    1031040  7.9  1030043  7.9
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 404504 10.9     741108 19.8   741108 19.8
Vcells 492233  3.8    1031040  7.9  1030772  7.9
FINISHED!
Total time: 0 minutes
```

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

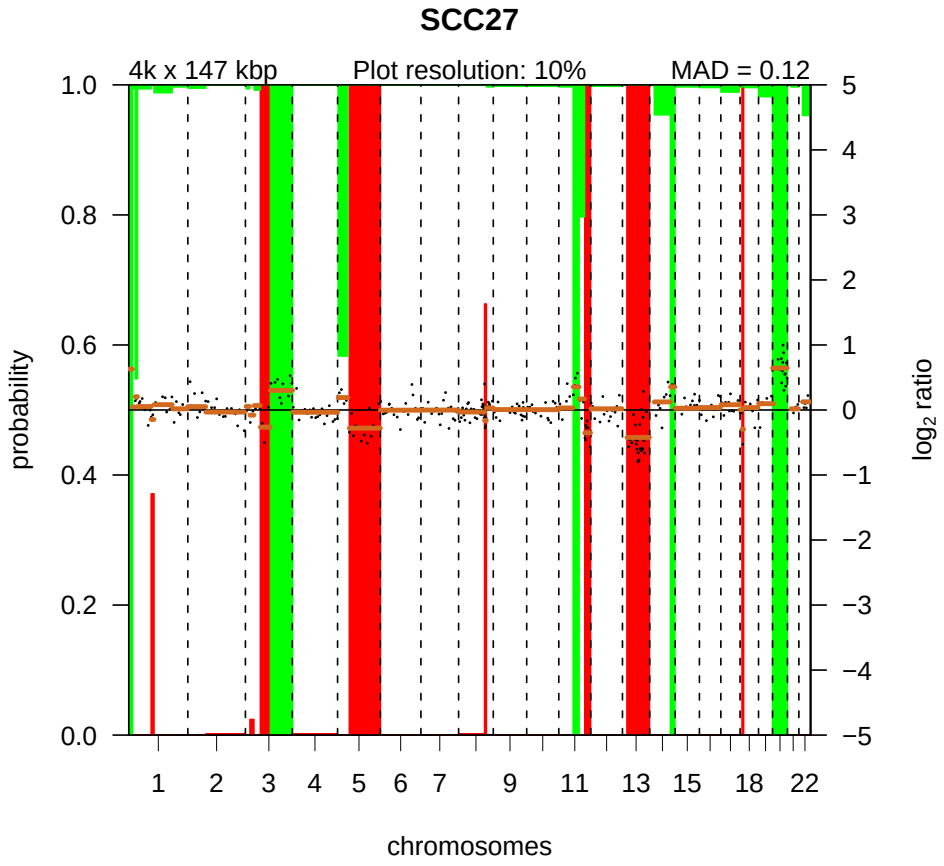
```
> plot(result[, 1])

Plotting sample AdCA10
```



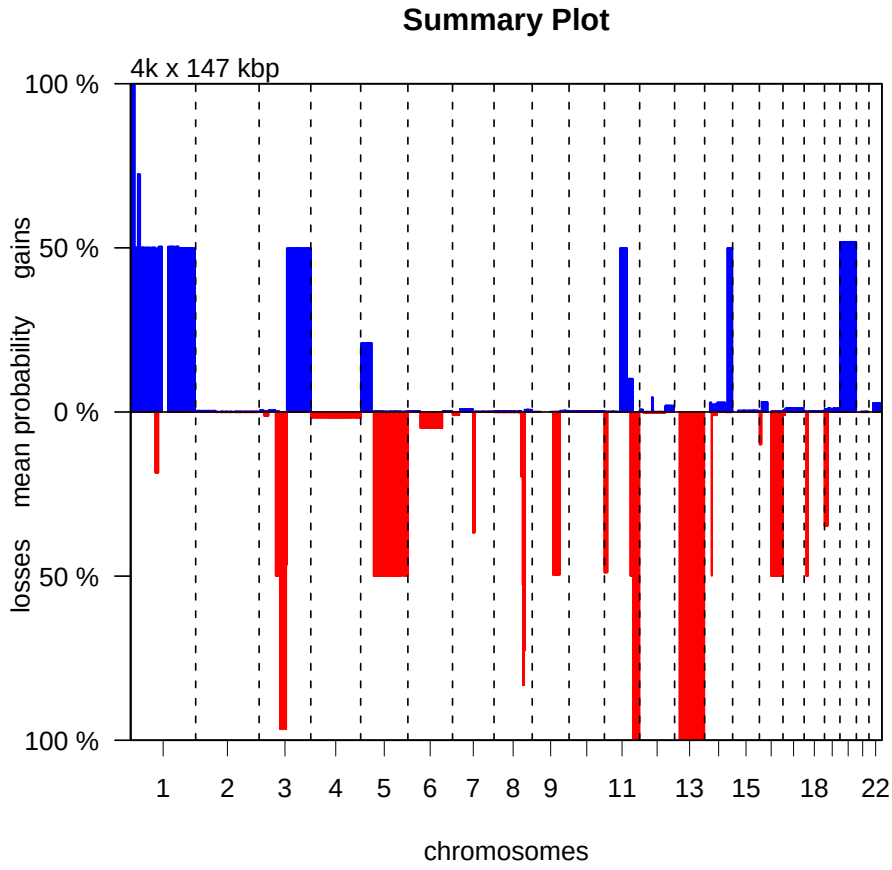
```
> plot(result[, 2])

Plotting sample SCC27
```



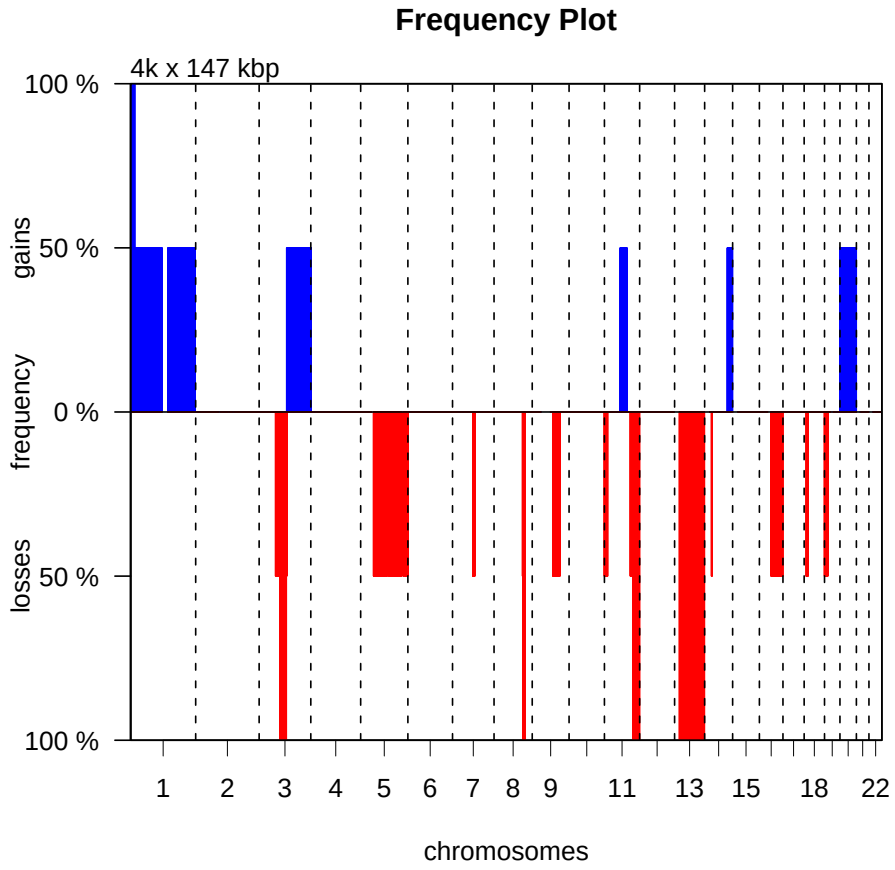
Alternatively, we can create a summary plot of all the samples:

```
> summaryPlot(result)
```



Or a frequency plot::

```
> frequencyPlotCalls(result)
```



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.