How to use cghMCR

Jianhua Zhang Bin Feng

April 25, 2006

1 Overview

This vignette demonstrate how to use *cghMCR* to locate minimum common regions (MCR) across arrayCGH profiles derived from different samples. MCR was initially proposed by Dr. Lynda Chin's lab (Aguirre et. al. 2004) to identify chromosome regions showing common gains/losses across samples using arrayCGH platform. The *cghMCR* pacakge implements the algothrim.

2 Getting Started

The example data used in this vignette are artificially constructed following the *Agilent* arrayCGH format within 5000 probes to maintain speed.

2.1 Read the sample data

The sample data are stroed in the *data* subdirectory and can be loaded using `data`.

```
> require("cghMCR")
```

```
Loading required package: cghMCR
Loading required package: DNAcopy
Loading required package: marray
Loading required package: limma

Attaching package: 'cghMCR'


        The following object(s) are masked from package:DNAcopy :

         plot.DNAcopy

[1] TRUE
```

```
> data("sampleData")
```

The sample data was created by reading three fabricated files using `read.Agilent` and then normalized using `maNorm` (norm = "loess") of *marray*. Readers are referred to *marray* for more information on how to read in Agilent profile data.

`sampleData` has three samples with intensity measures for 5000 probes.

```
> maNsamples(sampleData)

[1] 3

> length(maLabels(maGnames(sampleData)))

[1] 5000
```

2.2 Identify chromosome segments

For each sample, we need to first identify chromosome segments having similar intensity measures. The function `getSegments` is a wrapper around the main functions provided by *DNAcopy* that are capable of detecting chromosome regions within which probe intensities remain similar.

```
> segments <- getSegments(sampleData)

Analyzing: X.home.john.lib64.R.library.cghMCR.sampledata.sample1.agi
Analyzing: X.home.john.lib64.R.library.cghMCR.sampledata.sample2.agi
Analyzing: X.home.john.lib64.R.library.cghMCR.sampledata.sample3.agi
```

Results from the segmentation analysis (`segments`) is a list with three elements:

```
> names(segments)

[1] "data"   "output" "call"
```

The *data* element contain the normalized data, the *output* element contains the chromosome segments identified, and the *call* elements contains the function call with parameters passed indicated. Now, let's plot the original data and the segments to see what the segments look like.

```
> plot(segments)
```

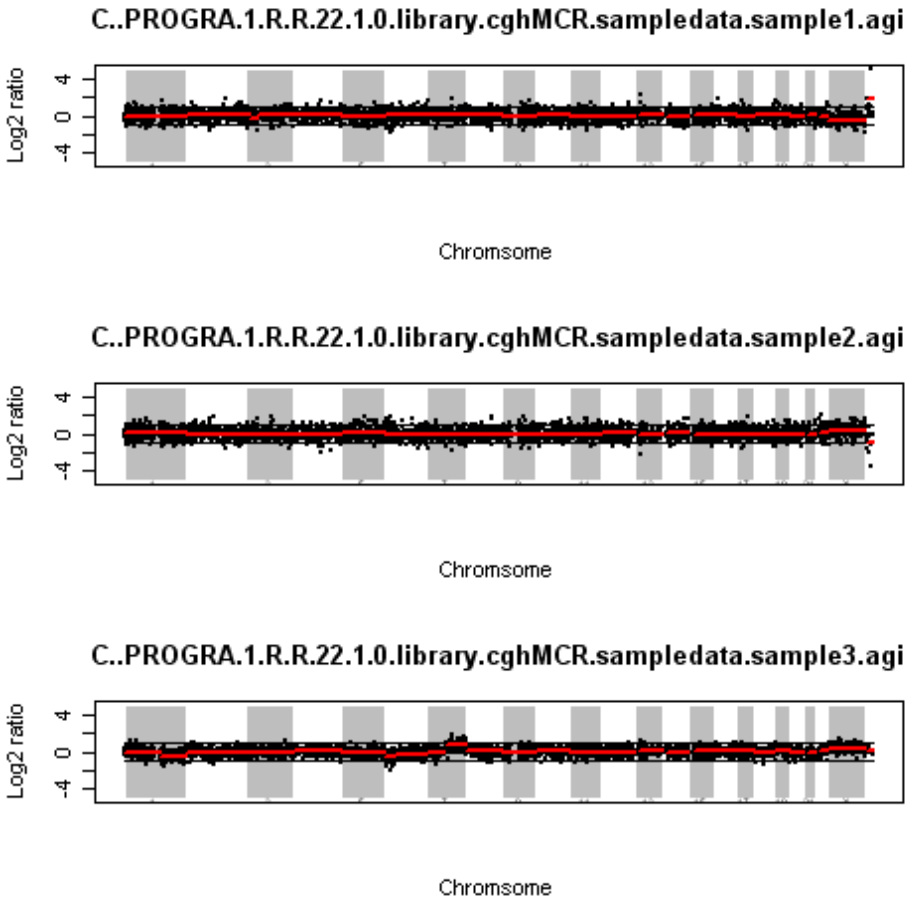
C..PROGRA.1.R.R.22.1.0.library.cghMCR.sampledata.sample1.agi
Log2 ratio
-4 0 4
Chromsome
C..PROGRA.1.R.R.22.1.0.library.cghMCR.sampledata.sample2.agi
Log2 ratio
-4 0 4
Chromsome
C..PROGRA.1.R.R.22.1.0.library.cghMCR.sampledata.sample3.agi
Log2 ratio
-4 0 4
Chromsome

Figure 1:

2.3 Identify MCRs

The element - `output` of the `segments` object generated in the previous section contains the segmentation data and will be used to get the MCRs. The parameter `margin` is numeric indicating how many basepairs should two adjancent segments be allowed to apart to be considered as one locus. Parameters `gain.threshold` and `loss.threshold` are also numerics indicating the minimum positive or maximum negative values for chromsome segments to be considered as gains or losses.

```
> cghmcr <- cghMCR(segments, margin = 0, gain.threshold = 0.8,
+     loss.threshold = -0.8)
> mcrs <- MCR(cghmcr)
```

Using the above settings, we get four MCRs but only one (on chromosome 7) of them are common to two samples.

```
> print(cbind(mcrs[, c("chromosome", "status", "mcr.start", "mcr.end",
+     "samples")]))
```

```
  chromosome status mcr.start  mcr.end
7 "7"        "loss" "36220646" "39814784"
Y "Y"        "gain" "4283463"  "26983049"
Y "Y"        "loss" "4283463"  "26983049"
  samples
7 "X.home.john.lib64.R.library.cghMCR.sampledata.sample1.agi,X.home.john.lib64.R.librar
Y "X.home.john.lib64.R.library.cghMCR.sampledata.sample1.agi"
Y "X.home.john.lib64.R.library.cghMCR.sampledata.sample2.agi"
```

To include probe ids for the MCRs identified, we can call the function `mergeMCR-Probes` to have probe ids within each MCR appended. Multiple probes are separated by a ",".

```
> mcrs <- mergeMCRProbes(mcrs, segments[["data"]])
> print(cbind(mcrs[, c("chromosome", "status", "mcr.start", "mcr.end",
+     "probes")]))
```

```
  chromosome status mcr.start  mcr.end
7 "7"        "loss" "36220646" "39814784"
Y "Y"        "gain" "4283463"  "26983049"
Y "Y"        "loss" "4283463"  "26983049"
  probes
7 "A_14_P119613,A_14_P121347,A_14_P101357"
Y "A_14_P120393,A_14_P124965,A_14_P127818,A_14_P120961,A_14_P106297,A_14_P112182,A_14_P
Y "A_14_P120393,A_14_P124965,A_14_P127818,A_14_P120961,A_14_P106297,A_14_P112182,A_14_P
```

3 Session Information

The version number of R and packages loaded for generating the vignette were:

```
Version 2.3.0 (2006-04-24)
x86_64-unknown-linux-gnu

attached base packages:
[1] "tools"     "methods"   "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "base"

other attached packages:
  cghMCR   marray    limma  DNAcopy
 "1.2.0" "1.10.0"  "2.6.0"  "1.6.0"
```

4 References

Aguirre, AJ, C. Brennan, G. Bailey, R. Sinha, B. Feng, C. Leo, Y. Zhang, J. Zhang, N. Bardeesy, C. Cauwels, C. Cordon-Cardo, MS Redston, RA DePinho and L. Chin. High-resolution Characterization of the Pancreatic Adenocarcinoma Genome. Proc Natl Acad Sci U S A. 2004. 101(24):9067-9072.