Analysis of copy number data

Oscar M. Rueda

September 12, 2011.

1 Overview

ℹ In this practical we will review the basics of analysing copy number data. There are a vast amount of methods and algorithms running in different platforms, and we will use one of the most popular: DNAcopy [3]. The data that we will analyze are aCGH samples of breast cancer tumors and cell lines from Pollack et al. [1]. Note that the original data has been preprocessed (the missing data have been imputed and replicate probes averaged).

2 Reading Data

First of all, we load the package, read the data and extract information about chromosome and position

```
> library(DNAcopy)
> load("PollackDataImputed.RData")
> head(CN)
> CN <- CN[order(CN$Chr, CN$Position),]
```

☞ See how we sort the data by chromosome and position with the `order` function.

ℹ We are going to analyze only the tumour samples, so first we extract them and then build a CNA object with them.

```
> Tumor <- CN[, c(grep("NORWAY", colnames(CN)), grep("STANFORD", colnames(CN)))]
> Tumor.CNA <- CNA(as.matrix(Tumor), CN$Chr, CN$Position, sampleid = colnames(Tumor))
```

☞ Note that we have to provide the logratios as a matrix.

ℹ The authors of the package recommend that we smooth the data in order to remove single point outliers. Then, we can segment our samples (it might take some minutes).

```
> smoothed.Tumor.CNA <- smooth.CNA(Tumor.CNA)
> seg.smoothed.Tumor.CNA <- segment(smoothed.Tumor.CNA)
```

☞ Explore the content of the objects using `str`

```
> str(seg.smoothed.Tumor.CNA)
> head(seg.smoothed.Tumor.CNA$output)
```

DNAcopy includes several options for plotting. We can plot each array separately, or compare directly samples genomewide or by chromosome...

```
> plot(seg.smoothed.Tumor.CNA, plot.type = "w", xmaploc = TRUE)
> plot(seg.smoothed.Tumor.CNA, plot.type = "s", xmaploc = TRUE)
> plot(seg.smoothed.Tumor.CNA, plot.type = "c", xmaploc = TRUE)
```

The xmaploc argument will plot the data in genomic coordinates. Now we should decide which regions are real copy number changes. We can check the plateau plots:

```
> plot(seg.smoothed.Tumor.CNA, plot.type = "p", xmaploc = TRUE)
```

Function mergeLevels in package aCGH [4] merges regions with similar segmented means, so they are easier to classify (it might take some time);

```
>  library(aCGH)
> merged.CN <- matrix(0, nrow = nrow(seg.smoothed.Tumor.CNA$data), ncol = 44)
> observed <- matrix(0, nrow = nrow(seg.smoothed.Tumor.CNA$data),ncol = 44)
> predicted <- matrix(0, nrow = nrow(seg.smoothed.Tumor.CNA$data), ncol = 44)
> colnames(merged.CN) <- colnames(Tumor)
> colnames(observed) <- colnames(Tumor)
> colnames(predicted) <- colnames(Tumor)
> for (i in colnames(Tumor)) {
+   observed[, i] <- seg.smoothed.Tumor.CNA$data[,i]
+   predicted[, i] <-
+     rep(seg.smoothed.Tumor.CNA$output[seg.smoothed.Tumor.CNA$output$ID==i, "seg.mean"],
+         seg.smoothed.Tumor.CNA$output[seg.smoothed.Tumor.CNA$output$ID==i, "num.mark"])
+   merged.CN[,i] <- mergeLevels(vecObs=observed[,i], vecPred=predicted[,i])$vecMerged
+ }
```

Now we can plot together the segmentation for any sample and compare with the segmented mean approach:

```
> par(mfrow = c(1, 1))
> plot(observed[, 3], pch = ".")
> lines(predicted[, 3], col = 2, lty = 2, lwd = 2)
> lines(merged.CN[, 3], col = 3, lty = 3, lwd = 2)
```

We can make calling for alterations using a simple threshold approach. First, we compute for each array its median and its standard deviation, and then use median+1.5sd and median-1.5sd to call gains and losses.

```
> Tum.m <- apply(Tumor, 2, median)
> Tum.sd <- apply(Tumor, 2, sd)
> calls <- matrix(0, nrow = nrow(Tumor), ncol = ncol(Tumor))
> colnames(calls) <- colnames(Tumor)
> for (i in colnames(Tumor)) {
+         calls[, i] <- 0
+         calls[which(predicted[, i] > (Tum.m[i] + 1.5 * Tum.sd[i])), i] <- 1
+         calls[which(predicted[, i] < (Tum.m[i] - 1.5 * Tum.sd[i])), i] <- -1
+       }
> apply(calls, 2, table)
```

We can plot the frequency of alterations and get a profile of the genomic alterations in breast cancer:

```
> Loss <- apply(calls, 1, function(x) mean(x == -1))
> Gain <- apply(calls, 1, function(x) mean(x == 1))
> plot(Gain, type = "h", col = "red", ylim = c(-1, 1),
+      xlab = "", ylab = "Freq.of alterations")
> lines(I(-Loss), type = "h", col = "blue")
> abline(v = which(diff(CN$Chr) != 0), lty = 2)
```

Repeat the calling and the plot using the mergeLevels values and changging the thresholds to see diffrences.

References

[1] Pollack JR, Sørlie T, Perou CM, Rees CA, Jeffrey SS, Lonning PE, Tibshirani R, Botstein D, Børresen-Dale AL, Brown PO. *Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors.* Proc Natl Acad Sci U S A. 2002 Oct 1;99(20):12963-8.

[2] Smith, M.L., Marioni, J.C., Hardcastle, T.J., Thorne, N.P. *snapCGH: Segmentation, Normalization and Processing of aCGH Data Users' Guide* Bioconductor, 2006.

[3] Olshen AB, Venkatraman ES, Lucito R, Wigler M. *Circular binary segmentation for the analysis of array-based DNA copy number data* . Biostatistics, 5(4):557-72. 2004

[4] Fridlyand J, Snijders AM, Pinkel D, Albertson DG. *Hidden markov models approach to the analysis of array cgh data. Journal of Multivariate Analysis* 90:132–153, July 2004.

[5] Willenbrock H, Fridlyand J. *A comparison study: applying segmentation to array CGH data for downstream analyses.* Bioinformatics, 21: 4084-4091, 2005.