

Circular Binary Segmentation of aCGH data

Lori Shepherd and Jonathan Dare

July 11, 2007

Please download and utilize example1.

We have provided all the files that would have been created leading up to and including an aCGH object.

In the directory example1 begin an R session and load the package library and aCGH object.

```
> library(aCGHplus)
> load("RData/aCGH.RData")
```

This aCGH object has 200 samples. We therefore will perform CBS on the aCGH using batch calls. We will use four calls, meaning each call will handle 50 samples.

```
> CBS.Batch(aCGH, overwrite = T, BatchDX = NA, nbatchjobs = 4,
+   ibatchjob = 1, time.order = F, alpha = 0.025, nperm = 10000,
+   outlier.mad.cut = 5, CBS.label = "CBS.nodesign", vrb = T)
> CBS.Batch(aCGH, overwrite = T, BatchDX = NA, nbatchjobs = 4,
+   ibatchjob = 2, time.order = F, alpha = 0.025, nperm = 10000,
+   outlier.mad.cut = 5, CBS.label = "CBS.nodesign", vrb = T)
> CBS.Batch(aCGH, overwrite = T, BatchDX = NA, nbatchjobs = 4,
+   ibatchjob = 3, time.order = F, alpha = 0.025, nperm = 10000,
+   outlier.mad.cut = 5, CBS.label = "CBS.nodesign", vrb = T)
> CBS.Batch(aCGH, overwrite = T, BatchDX = NA, nbatchjobs = 4,
+   ibatchjob = 4, time.order = F, alpha = 0.025, nperm = 10000,
+   outlier.mad.cut = 5, CBS.label = "CBS.nodesign", vrb = T)
```

Let's examine the second batch call.

```
CBS.Batch(aCGH,overwrite=T,BatchDX=NA,nbatchjobs=4,
          ibatchjob=2,time.order=F,alpha=0.025,nperm=10000,
          outlier.mad.cut=5,CBS.label="CBS.nodesign",vrb=T)
```

We may say the following:

- any files that already exist will be re-written
- all the samples are being considered

- there are four batch calls, this call handling the second of four. Samples 50 to 100 will undergo CBS.
- time.order IGNORE THIS ARGUMENT IT IS CURRENTLY NON FUNCTIONAL
- alpha, nperm, and outlier.mad.cut are all values that will be passed into the DNACopy function of bioconductors DNACopy package
- the CBS data will be stored in the samples' array image file as CBS.nodesign object

After CBS is run on all the samples, the CBS data must be assembled in the aCGH object. This is done with a call to assemble.CBS.Batch

```
> aCGH = Assemble.CBS.Batch(aCGH, CBS.label = "CBS.nodesign", overwrite.log2.ratios = F)
```

This will take all CBS data and store it in the aCGH object for convenience. The new smoothed log2 ratios are stored in the aCGH object. If overwrite.log2.ratios is T, the new smoothed values will replace the log2 ratios that were stored when creating the aCGH object. If overwrite.log2.ratios is F, a new variable object, log2.ratios.fitted, is created to store the new values.

Lastly, some functions do not allow missing data. There is a function what will fill in any missing data and store where missing data did occur.

```
> aCGH = flankNA.CBS(aCGH, saveFlag = T, file = "RData/aCGH.RData")
```

A NA matrix is created. Any value that was missing will be True in this matrix. The aCGH object is now loaded and processed. Statistical analysis and visualization may be performed on the data.