Preprocessing & Genotyping Affymetrix Arrays for Copy Number Analysis

Rob Scharpf

March 25, 2011

Abstract

This vignette illustrates the preprocessing and genotyping of Affymetrix 5.0 and 6.0 platforms. These steps must be completed prior to copy number analyses in crlmm. After completing these steps, users can refer to the `copynumber` vignette.

1 Set up

```
> library(crlmm)
> library(ff)
> library(cacheSweave)
```

This vignette analyzes HapMap samples assayed on the Affymetrix 6.0 platform. The annotation package for this platform is genomewidesnp6Crlmm. We assign the name of the annotation package without the `Crlmm` postfix to the name `cdfName`.

```
> cdfName <- "genomewidesnp6"
```

The HapMap CEL files are stored in a local directory assigned to `pathToCels` in the following code. The genotyping step will create several files with `ff` extensions. We will store these files to the path indicated by `outdir`.

```
> pathToCels <- "/thumper/ctsa/snpmicroarray/hapmap/raw/affy/1m"
> if (getRversion() < "2.13.0") {
     rpath <- getRversion()
 } else rpath <- "trunk"
> outdir <- paste("/thumper/ctsa/snpmicroarray/rs/ProcessedData/crlmm/",
     rpath, "/copynumber_vignette", sep = "")
> dir.create(outdir, recursive = TRUE, showWarnings = FALSE)
```

By providing the path in `outdir` as an argument to the R function `ldPath`, all of the `ff` files created during the genotyping step will be stored in `outdir`.

```
> ldPath(outdir)
```

This vignette uses the R package cacheSweave to cache long computations. The following step is only necessarily if one wishes to cache some of the computations. In particular, we specify that the cached computations will be saved in the `outdir` through the function `setCacheDir`. Users should refer to the cacheSweave package for additional details regarding cacheing.

The R functions `ocProbesets` and `ocSamples` manage the RAM required for our analysis. See the documentation for these functions and the `CopyNumberOverview` vignette for additional details.

```
> ocProbesets(1e+05)
> ocSamples(200)
```

Next we indicate the local directory that contains the CEL files. For the purposes of this vignette, we only analyze the CEPH ('C') and Yoruban ('Y') samples.

```
> celFiles <- list.celfiles(pathToCels, full.names = TRUE,
     pattern = ".CEL")
> celFiles <- celFiles[substr(basename(celFiles), 13, 13) %in%
     c("C", "Y")]
```

Finally, copy number analyses using crlmm require specification of a batch variable that is used to indicate which samples were processed together. For example, if some of the samples were processed in April and another set of samples were processed in June, we could name the batches 'April' and 'June', respectively. A useful surrogate for batch is often the chemistry plate or the scan date of the array. For the HapMap CEL files analyzed in this vignette, the CEPH (C) and Yoruban (Y) samples were prepared on separate chemistry plates. In the following code chunk, we extract the population identifier from the CEL file names and assign these identifiers to the variable `plate`.

```
> plates <- substr(basename(celFiles), 13, 13)
```

2 Preprocessing and genotyping.

The preprocessing steps for copy number estimation includes quantile normalization of the raw intensities for each probe and a step that summarizes the intensities of multiple probes at a single locus. For example, the Affymetrix 6.0 platform has 3 or 4 identical probes at each polymorphic locus and the normalized intensities are summarized by a median. For the nonpolymorphic markers on Affymetrix 6.0, only one probe per locus is available and the summarization step is not needed. After preprocessing the arrays, the crlmm package estimates the genotype using the CRLMM algorithm and provides a confidence score for the genotype calls. The function `genotype` performs both the preprocessing and genotyping.

```
> cnSet <- genotype(celFiles, batch = plates, cdfName = cdfName)
```

The value returned by genotype is an instance of the class `CNSet`. The normalized intensities, genotype calls, and confidence scores are stored as `ff` objects in the `assayData` slot. A concise summary of this object can be obtained throught the `print` or `show` methods.

```
> print(cnSet)
```

```
CNSet (storageMode: lockedEnvironment)
assayData: 1852215 features, 180 samples
  element names: alleleA, alleleB, call, callProbability
protocolData
  rowNames: NA06985_GW6_C.CEL NA06991_GW6_C.CEL ...
    NA19240_GW6_Y.CEL (180 total)
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  rowNames: NA06985_GW6_C.CEL NA06991_GW6_C.CEL ...
    NA19240_GW6_Y.CEL (180 total)
  varLabels: SKW SNR gender
  varMetadata: labelDescription
featureData
  featureNames: SNP_A-2131660 SNP_A-1967418 ... CN_954736
    (1852215 total)
  fvarLabels: chromosome position isSnp
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
```

```
Annotation: genomewidesnp6
batch:    C 90, Y 90, grandMean 90
batchStatistics:  29  elements,  1852215  features,  2  batches
```

Note that the object is fairly small as the intensities and genotype calls are stored on disk rather than in active memory.

```
> object.size(cnSet)

140824280 bytes
```

Users can proceed to the `copynumber` vignette for copy number analyses.