Load the package snow. Enable 10 worker CPUs.

```
> library(cacheSweave)
> library(oligoClasses)
> if(FALSE){
+         library(snow)
+         options(cluster=makeCluster(10, "SOCK"))
+         library(doSNOW)
+         registerDoSNOW(getCluster())
+ }
```

Enable a large data back-end using the ff package. Specify a local file directory for storing the ff files to reduce I/O.

```
> library(ff)
> outdir <- "/local_data/r00/beaty"
> suppressWarnings(dir.create(outdir))
```

Load the MinimumDistance package.

```
> library(MinimumDistance)
> ## files containing LRRs and BAFs
> path <- file.path(outdir, "txtfiles")
> fnames <- list.files(path)
> ldPath(outdir)
```

Load information on the pedigrees as well as phenotypic information on the samples for the cleft study. An object of class `pedigree` is required, but a code `samplesheet` object is not. The latter is primarily useful for examining the effects of DNA source, chemistry plate, etc. on the resulting inference (quality control).

```
> library(CleftExperimentData)
> load("~/Projects/BeatyExperimentData/data/samplesheet.rda")
> samplesheet <- samplesheet[, -(30:31)]
> ped <- read.csv("/thumper/ctsa/beaty/family_info/may_peds.csv", sep=";",
+                 stringsAsFactors=FALSE)
> ## functions in CleftExperimentDAta needed for Cleft study
> samplesheet <- removeFlaggedSamplesAndMerge(fnames, samplesheet, ped)
> pedigree <- pedigreeFromSampleSheet(fnames, samplesheet)
```

Construct an object of class RclassTrioSetList using a constructor for large data (LD). The assay data elements of the `TrioSetList` object will be *ff*-derived classes. More precisely, the assay data elements have class *ff_array* with dimensions `number markers x number trios x 3`. The arrays are stored by chromosome to facilitate parallel processing of different chromosomes and to reduce the size of the individual *ff* files on disk.

```
> ped <- pedigree[1:5, ]
> trioSetListff <- TrioSetListLD(path=path, fnames=allNames(ped), ext=".txt",
+                                samplesheet=samplesheet, ##optional
+                                row.names=samplesheet$Sample.Name, ## must provide if samplesheeet is n
+                                pedigree=ped,
+                                annotationPkg="human610quadv1bCrlmm")
> dim(lrr(trioSetListff))[[1]]
> lrr(trioSetListff)[[1]][1, , ]
> baf(trioSetListff)[[1]][1, , ]
> ##save(trioSetListff, file=file.path(outdir, "trioSetListff.rda"))
```

```
> load("/local_data/r00/beaty/fffiles/trioSetListff.rda")
```

Calculate the minimum distance for the list of arrays containing the log R ratios:

```
> mdlist <- calculateMindist(lrr(trioSetListff))
> mdlist[[1]][1:10,]
```

Segment the log R ratios using circular binary segmentation. The function `foreach` creates a separate process depending on the available number of CPUs for each chromosome. A better understanding of the total RAM required for processing a given number of trios is needed ...

```
> lrr.segs <- segment2(trioSetListff, segmentParents=FALSE)
```

Segment the minimum distance using circular binary segmentation. Again, the function `foreach` creates a separate process for each chromosome. A better understanding of the total RAM required for processing a given number of trios is needed ...

```
> md.segs <- segment2(trioSetListff, md=mdlist)
```

Compute variance estimates needed for downstream processing. The backend of these functions uses `foreach`, though the implementation is currently less efficient than it could be. See `R/mad-methods.R` for the source code.

```
> mads.md <- mad2(mdlist, byrow=FALSE) ## mad across all autosomes
> md.segs2 <- narrow(md.segs, lrr.segs, thr=0.75, mad.minimumdistance=mads.md)
```

Finally, posterior calls for the copy number state is obtained for each range using the function `computeBayesFactor`.

```
> map.segs <- computeBayesFactor(trioSetListff, ranges=md.segs2)
```

# Session Information

```
> toLatex(sessionInfo())
```

- R Under development (unstable) (2012-01-30 r58229), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.iso885915`, `LC_NUMERIC=C`, `LC_TIME=en_US.iso885915`, `LC_COLLATE=en_US.iso885915`, `LC_MONETARY=en_US.iso885915`, `LC_MESSAGES=en_US.iso885915`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.iso885915`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: Biobase 2.15.3, BiocGenerics 0.1.4, BiocInstaller 1.3.7, bit 1.1-8, cacheSweave 0.6, CleftExperimentData 0.0.1, codetools 0.2-8, ff 2.2-5, filehash 2.2, foreach 1.3.2, IRanges 1.13.19, iterators 1.0.5, MinimumDistance 0.2.11, oligoClasses 1.17.20, stashR 0.3-4
- Loaded via a namespace (and not attached): affyio 1.23.1, Biostrings 2.23.6, digest 0.5.1, DNAcopy 1.29.0, grid 2.15.0, lattice 0.20-0, msm 1.1, mvtnorm 0.9-9992, splines 2.15.0, survival 2.36-10, VanillaICE 1.17.12, zlibbioc 1.1.1