aroma.affymetrix
-
Analysis of small to very large Affymetrix data sets

Henrik Bengtsson

Department of Statistics, University of California, Berkeley

August 7, 2007

Outline

Objectives

- Show you how to analyze large Affymetrix data sets in *R*.
- Walk through: Copy-number analysis of some HapMap samples.

Acknowledments

Aroma.affymetrix would not be available if it wasn't for:

- Ken Simpson, WEHI, Melbourne (aroma.affymetrix; exon)
- James Bullard, UC Berkeley (affxparser)
- Kasper D. Hansen, UC Berkeley (affxparser)
- Ben Bolstad, Affymetrix (affyPLM)

Today's problem analyzing Affymetrix data in R

- Need to analyze 100-1,000 of arrays.
- "Error: Out of memory!"
- Hard to develope real-world statistical models.
- New chip types are added fast - need a generic framework.
- Tons of file types and file conversions.

aroma.affymetrix - Yet another tool?!?

Features:

- Number of arrays: unlimited.
- Chip types: all Affymetrix chip types with a CDF, e.g. SNP, exon, and gene expression.
- Memory requirements: 1.0-1.5GB RAM (without swapping!).
- Cross platform: Any R installation, e.g. Linux, Windows XP & OSX.
- Cross platform: Setup and analysis the same anywhere.
- Easy to migrate data: works directly with CEL and CDF files.
- Design goals: usability, quality & extendibility.

aroma.affymetrix - Yet another tool?!?

Features:

- Persistent memory: Final and intermediate results and estimates are stored on file.
- No need to rerun analysis between R sessions.
- Robustness: Analysis picks up where last interrupted.
- Export to/Import from: Bioconductor, CNAG, CNAT, dChip, Affymetrix Power Tools.
- Dynamic HTML reports: ArrayExplorer, ChromosomeExplorer.
- Online documentation & forum: aroma.affymetrix Google Group.
- Home page: `http://www.braju.com/R/aroma.affymetrix/`

aroma.affymetrix - Yet another tool?!?

Features:

- Pre-processing: Background correction, allelic cross-talk calibration, quantile normalization etc.
- Probe-level modelling: RMA, MBEI, affine.
- Post-processing: PCR fragment-length normalization.
- Copy-number analysis: GLAD. Possible to combine data from multiple chip types.
- Alternative splicing: FIRMA (under development).
- Transparent interface to BRLMM and CRLMM (a bit out of date).
- Upcoming: Utilization of Bioconductor annotation packages too.

Script for identifying CN regions

```
library(aroma.affymetrix)
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
qn <- QuantileNormalization(cs)
csN <- process(qn)
plm <- RmaCnPlm(csQN, combineAlleles=TRUE, mergeStrands=TRUE)
fit(plm)
ces <- getChipEffects(plm)
fln <- FragmentLengthNormalization(ces)
cesN <- process(ces)
glad <- GladModel(cesN)
fit(glad)
```

Outline

What's needed to get started?

1. The most recent version of R
2. Some CRAN, some BioC and some other packages installed
3. Data files CEL files
4. Annotation files: CDF files and some dChip files.
5. Know R

To install from online repositories, see group page.

TASK: Installation from BioC2007 disk

1. Change directory to the repos/ of the installation disk/directory. This directory contains subdirectory bioc2007/.
2. Start R.
3. Install all required package, if missing:
   ```
   source("hbLite.R")
   hbInstall("aroma.affymetrix", repos="bioc2007")
   ```
4. Test the installation:
   ```
   library(aroma.affymetrix)
   ```
5. Quit R.

To install from online repositories, see the group page (`http://www.braju.com/R/aroma.affymetrix/`).

Directory structures of aroma.affymetrix

1. Strict directory structures!
   - All annotation data in one directory tree, e.g `annotationData/chipTypes/Mapping50K_Hind240/Mapping50K_Hind240.CDF`
   - All raw data in one directory tree, e.g `rawData/HapMap100K-bioc2007/Mapping50K_Hind240/`
   - All processed data in other directories, e.g `probeData/HapMap100K-bioc2007,QN/Mapping50K_Hind240/` `plmData/HapMap100K-bioc2007,QN,RMA/Mapping50K_Hind240/`
2. Only relative pathnames; No absolute pathnames!
   - Easy to migrate data and analysis between systems and platforms.
   - Use Unix soft links or Windows Shortcuts to avoid copying files.

Outline

Annotation data

- Annotation data files contains information about a chip types or a genome studied.
- It is not specific to a data set.
- Different projects may share the same annotation data.
- Example of annotation data files: Affymetrix CDF files, dChip SNP information files, dChip genome information files.
- Root directory: annotationData/

TASK: Setup the CDF

Create the correct annotationData/ directory and copy the Mapping50K_Hind240.CDF file to it. You'll find the CDF file under labFiles/CDFs/.
Remember, everything is done with relative pathnames, so put it under the current working directory!
Verify that aroma.affymetrix can find it. You should get:

```
> library(aroma.affymetrix)
> cdf <- AffymetrixCdfFile$fromChipType("Mapping50K_Hind240")
> cdf
AffymetrixCdfFile:
Path: annotationData/chipTypes/Mapping50K_Hind240
Filename: Mapping50K_Hind240.CDF
Filesize: 53.43MB
File format: v4 (binary; XDA)
Chip type: Mapping50K_Hind240
Dimension: 1600x1600
Number of cells: 2560000
Number of units: 57299
Cells per unit: 44.68
Number of QC units: 9
RAM: 0.00MB
```

TASK: File- and directory names

A *filename* consist of a *full name* and an *extension*:

```
<filename> = <fullname>.<extension>
Example: Mapping50K_Hind240.CDF
```

The *path* to a file is the directory where the file lives:

```
Example: annotationData/chipTypes/Mapping50K_Hind240/
```

The *pathname* of a file is the path plus the filename:

```
<pathname> = <path>/<filename>
Example: annotationData/chipTypes/Mapping50K_Hind240/
                                  Mapping50K_Hind240.CDF
```

Try these commands:

```
cdf <- AffymetrixCdfFile$fromChipType("Mapping50K_Hind240")
getPathname(cdf)
getPath(cdf)
getFilename(cdf)
getFileSize(cdf)
```

TASK: Querying CDFs

Try these commands:

```
cdf <- AffymetrixCdfFile$fromChipType("Mapping50K_Hind240")
print(cdf)
nbrOfUnits(cdf)
nbrOfCells(cdf)
nbrOfRows(cdf)
nbrOfColumns(cdf)
AffymetrixCdfFile
```

TASK: Querying CDFs

Try these commands:

```
# Get the unit (probeset) names
unitNames <- getUnitNames(cdf)
unitNames <- getUnitNames(cdf, units=101:120)

# Locate a specify unit
unit <- indexOf(cdf, "SNP_A-1683825")

# Get all annotation data for unit 121
data <- readUnits(cdf, units=121)
names(data)
names(data[[1]]$groups)

# Get the cell (probe) indices for unit 121
idxs <- getCellIndices(cdf, units=121)
```

Outline

Raw data

- The term "raw data" refers to the initial data obtained from the scan of the array.
- Raw data is specific to a data set.
- Don't loose it! Make sure to back it up.
- Consider it to be read only.
- Example of raw data files: Affymetrix CEL files.
- Root directory: rawData/

Sample names and sample tags

Example: Typical, but poor filename format

- MCF7_Hind.CEL,
- MCF7_Xba.CEL,
- C358 Nsp 24-01-2007 250K.CEL, and
- NA06985_Hind_B5_3005533.CEL

Strict filename format: The fullname of the filename contains of a name followed by optional comma-separated tags:

```
<fullname> = <name>(,<tag>)*
```

Example: "Computer-readable" filename format

- MCF7,Hind.CEL,
- MCF7,Xba.CEL,
- C358,Nsp,24-01-2007,250K.CEL, and
- NA06985,Hind,B5,3005533.CEL.

Data set names, tags, and directory names

```
<dirname> = <name>(,<tag>)*
<path> = <path>/<dirname>
```

Example:

```
Affymetrix_2006-500k,ACC,QN/
```

means

- Data set: Affymetrix_2006-500k
- Tags: ACC & QN

Example: Same data set, different tags

```
Affymetrix_2006-500k/
Affymetrix_2006-500k,ACC/
Affymetrix_2006-500k,ACC,QN/
```

CEL files should go under rawData/

- Data set name: SinclairA_etal_2006
- Chip types: Mapping250K_Nsp & Mapping250K_Sty
- Data files: 11 Nsp and 11 Sty CEL files (10 in common)

```
rawData/
 +- SinclairA_etal_2006/
     +- Mapping250K_Nsp/
     |    +- 01,XX,2006-07-12.cel
     |    +- 02,XY,2006-07-14.cel
     |    |  ...
     |    +- 10,XX.cel
     |    +- C1.cel 02,XY,2006-07-12.cel
     +- Mapping250K_Sty/
          +- 01,XX,2006-07-19,2nd.cel
          +- 02,XY,2006-07-20.cel
          |  ...
          +- 10,XX.cel
          +- C358,control.cel
```

TASK: Setup of CEL files

Setup a data set named 'HapMap100K-bioc2007' in the rawData/ directory and copy the Hind CEL files found in labFiles/CELs/ to it. Verify that aroma.affymetrix can find it. You should get:

```
> cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                  chipType="Mapping50K_Hind240")
> print(cs)
AffymetrixCelSet:
Name: HapMap100K-bioc2007
Tags:
Path: rawData/HapMap100K-bioc2007/Mapping50K_Hind240
Chip type: Mapping50K_Hind240
Number of arrays: 6
Names: NA06985, NA06991, ..., NA07019
Time period: 2004-01-14 14:02:08 -- 2004-01-17 13:53:58
Total file size: 146.86MB
RAM: 0.01MB
```

TASK: Query a CEL set

Try these commands:

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
getName(cs)
getPath(cs)
getNames(cs)
getFullNames(cs)
getFileSize(cs)
cdf <- getCdf(cs)
AffymetrixCelSet
```

TASK: Querying a CEL set

Try these commands:

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
lapply(cs, getFullName)
as.list(cs)
cf <- getFile(cs, 1)
cf
```

TASK: Accessing a single CEL file

Try these commands:

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
cf <- getFile(cs, 1)
cf
```

Questions:

- What is the class of the 'cf' object?
- How can you quickly find out what methods you can apply to it?

Summary: annotation and raw data directories

```
annotationData/
 +- chipTypes/
     +- Mapping50K_Hind240/
         +- Mapping50K_Hind240.CDF

rawData/
 +- HapMap100K-bioc2007/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533.CEL
         +- NA06991,B6,3005533.CEL
         +- NA06993,B4,4000092.CEL
         +- NA06994,A7,3005533.CEL
         +- NA07000,A8,3005533.CEL
         +- NA07019,A12,4000092.CEL
```

TASK: PM-signal densities for all arrays

Plot the probe-signal densities by:

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
plotDensity(cs, stratifyBy="pm")
```

TASK$^{\text{EBImage}}$: Spatial image of an array

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
cf <- getFile(cs, 1)
plotImage(cf)
```

Quantile normalization - setup

```
> qn <- QuantileNormalization(cs)
> qn
QuantileNormalization:
Data set: HapMap100K-bioc2007
Input tags:
Output tags: QN
Number of arrays: 6 (146.86MB)
Chip type: Mapping50K_Hind240
Algorithm parameters: (subsetToUpdate: NULL, typesToUpdate: NULL,
  subsetToAvg: NULL, typesToAvg: NULL, .targetDistribution: NULL)
Output path: probeData/HapMap100K-bioc2007,QN/Mapping50K_Hind240
Is done: FALSE
RAM: 0.00MB
```

Note the output path and the added tag. Transformed probe data *will be* stored in the probeData/ directory.

Quantile normalization - processing

```
> qn <- QuantileNormalization(cs)
> process(qn, verbose=TRUE)  # Approx 2 min = 20 secs/arrays
Quantile normalizing data set...
 Retrieving target distribution...
 Retrieving target distribution...done
 Normalizing data towards target distribution...
  Identifying the probes to be updated...
  Identifying the probes to be updated...done
  Normalizing 2560000 probes
  Normalizing 6 arrays...
   Array #1...
   :
   :
   Array #6...done
  Normalizing 6 arrays...done
 Normalizing data towards target distribution...done
Quantile normalizing data set...done
```

TASK: Quantile normalization - result

Do quantile normalization:

```
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
qn <- QuantileNormalization(cs)
csQN <- process(qn, verbose=TRUE) # Approx 2 min = 20 secs/arrays
```

Look the normalized densities:

```
plotDensity(csQN, stratifyBy="pm")
```

Summary: data directories

```
rawData/
 +- HapMap100K-bioc2007/
      +- Mapping50K_Hind240/
           +- NA06985,B5,3005533.CEL
           +- NA06991,B6,3005533.CEL
           +- NA06993,B4,4000092.CEL
           +- NA06994,A7,3005533.CEL
           +- NA07000,A8,3005533.CEL
           +- NA07019,A12,4000092.CEL
probeData/
 +- HapMap100K-bioc2007,QN/
      +- Mapping50K_Hind240/
           +- NA06985,B5,3005533.CEL
           +- NA06991,B6,3005533.CEL
           +- NA06993,B4,4000092.CEL
           +- NA06994,A7,3005533.CEL
           +- NA07000,A8,3005533.CEL
           +- NA07019,A12,4000092.CEL
```

TASK: Quantile normalization - restart R

Quit R, and do the following:

```
library(aroma.affymetrix)
cs <- AffymetrixCelSet$fromName("HapMap100K-bioc2007",
                                chipType="Mapping50K_Hind240")
qn <- QuantileNormalization(cs)
csQN <- process(qn, verbose=TRUE)
```

Did you notice what happened when you ran process()?

Memory usage

Now, after all of the above, check the memory usage:

```
> gc()
         used (Mb) gc trigger (Mb) max used (Mb)
Ncells 1173991 31.4    1590760 42.5  1476915 39.5
Vcells  476976  3.7    1031040  7.9   711152  5.5
```

Outline

Total copy numbers: PLM

PLM:

$$\log_2 y_{ik} = \log_2 \theta_i + \log_2 \phi_k + \varepsilon_{ik}$$

$$y_{ik} = y_{Aik} + y_{Bik} (\text{“combining alleles”})$$

```
> plm <- RmaCnPlm(csQN, combineAlleles=TRUE, mergeStrands=TRUE)
> plm
RmaCnPlm:
Data set: HapMap100K-bioc2007
Chip type: Mapping50K_Hind240
Input tags: QN
Output tags: QN,RMA,A+B
Parameters: (probeModel: chr "pm"; shift: num 0; flavor: chr "affyPLM";
             mergeStrands: logi TRUE; combineAlleles: logi TRUE).
Path: plmData/HapMap100K-bioc2007,QN,RMA,A+B/Mapping50K_Hind240
RAM: 0.00MB
> fit(plm, verbose=TRUE) # 5-6 minutes
```

Behind the scenes: Monocell CDFs

Monocell CDFs describes how chip effects should be stored. If missing, a monocell CDF is created first time it is needed, which takes approx 5-10 minutes for our chip type.

```
> plm <- RmaPlm(csQN)
> fit(plm, verbose=TRUE)  # Will create the monocell CDF, if missing!
```

TASK: Copy Monocell CDFs (to speed up lab)

Instead of having to wait for the Hind monocell CDF to be created, copy it from labFiles/ to the corresponding annotationData/ directory.

You should now have something like:

```
> list.files("annotationData/", recursive=TRUE)
[3] "chipTypes/Mapping50K_Hind240/Mapping50K_Hind240,monocell.cdf"
[4] "chipTypes/Mapping50K_Hind240/Mapping50K_Hind240.CDF"
```

TASK: Total copy numbers: PLM

Do a total copy-number PLM fit:

```
plm <- RmaCnPlm(csQN, combineAlleles=TRUE, mergeStrands=TRUE)
fit(plm, verbose=TRUE) # 5-6 minutes
```

Where is the fitted data stored? What happens if you call fit() again?

Summary: data directories

```
probeData/
 +- HapMap100K-bioc2007,QN/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533.CEL
         +- NA06991,B6,3005533.CEL
         +- NA06993,B4,4000092.CEL
         +- NA06994,A7,3005533.CEL
         +- NA07000,A8,3005533.CEL
         +- NA07019,A12,4000092.CEL
plmData/
 +- HapMap100K-bioc2007,QN,RMA,A+B/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533,chipEffects.cel
         +- NA06991,B6,3005533,chipEffects.cel
         +- NA06993,B4,4000092,chipEffects.cel
         +- NA06994,A7,3005533,chipEffects.cel
         +- NA07000,A8,3005533,chipEffects.cel
         +- NA07019,A12,4000092,chipEffects.cel
         +- probeAffinities.cel
```

Total copy numbers: Chip effects

```
> ces <- getChipEffectSet(plm)
> readUnits(ces, units=100)
$`SNP_A-1684395`
$`SNP_A-1684395`$AG
$`SNP_A-1684395`$AG$theta
          [,1]    [,2]     [,3]     [,4]    [,5]     [,6]
[1,] 4153.886 4003.84 4513.098 4511.328 5446.43 5853.46
```

dChip SNP information files

Fragment-length normalization regresses chip effects on the *physical position* of the SNPs. The SNP positions are not available in the CDF. However, aroma.affymetrix recognized dChip's *SNP information files*, which contains this information. Download them from `http://www.dchip.org/` (or copy them from the lab disk), and put them under the corrsponding chip type annotationData/ directory, e.g. annotationData/Mapping50K_Hind240/Mapping100K_Hind snp info.txt.

Contents of Mapping100K_Hind snp info.txt:

```
Probe Set ID  dbSNP RS ID Flank                        Fragment Length Start Stop  FreqAsian FreqAfAm FreqCauc
SNP_A-1712762       ---   gggcag...atca[A/G]tctt...ggt  ---                                1      0.98     0.92
SNP_A-1718890 rs1496555   ggccta...agtg[A/G]actt...ctg  1444 // 2265866 // 2267309      0.18       0.1     0.12
:
:
SNP_A-1686301 rs547956    ggcagc...gaaa[C/T]gagg...tgc   537 // 153980985 // 153981521   0.01      0.11        0
```

dChip genome information

For the copy-number segmentation modelling coming later, aroma.affymetrix will need to know the genome position for each of the loci. This is available in dChip's *genome information file*, e.g. '50k hind genome info AfAm june 05 hg17.xls'.

Probe Set ID	Chromosome	Physical Position	Genetic Map	Strand	dbSNP RS ID	Freq AfAm	heterrate
SNP_A-1718890	1	2266413		+	rs1496555	10.94	
SNP_A-1668776	1	3118283		-	rs4587514	21.43	
:							
SNP_A-1686301	X	153981072		+	rs547956	11.9	

TASK: Setup dChip SNP and genome information files

Copy both the dChip SNP information file and the dChip genome information file from the labFiles/dChip/ directory to the correct annotationData/ subdirectory.

You should now have something like:

```
> list.files("annotationData/chipTypes/", recursive=TRUE)
[1] "Mapping50K_Hind240/50k hind genome info AfAm june 05 hg17.xls"
[2] "Mapping50K_Hind240/Mapping100K_Hind snp info.txt"
[3] "Mapping50K_Hind240/Mapping50K_Hind240,monocell.cdf"
[4] "Mapping50K_Hind240/Mapping50K_Hind240.CDF"
```

Verify that aroma.affymetrix finds these files:

```
cdf <- AffymetrixCdfFile$fromChipType("Mapping50K_Hind240")

gi <- getGenomeInformation(cdf)
print(gi)

si <- getSnpInformation(cdf)
print(si)
```

Fragment-length effects

```
ces <- getChipEffectSet(plm)

# Extract them as matrix where then columns are the arrays
theta <- extractMatrix(ces)

# Calculate the average across arrays
thetaAvg <- rowMedians(theta)

# Calculate the log-ratios (relative copy numbers)
M <- log2(theta/thetaAvg)

# The SNP information contains information about SNP fragment lengths.
cdf <- getCdf(ces)
si <- getSnpInformation(cdf)
fl <- getFragmentLengths(si)

layout(matrix(1:6, ncol=3, byrow=TRUE))
for (kk in seq(ces)) {
  smoothScatter(fl, M[,kk], ylim=c(-1.5,1.5), main=kk)
  lines(smooth.spline(fl, M[,kk]), col="red", lwd=3)
}
```

Fragment-length normalization

```
> ces <- getChipEffectSet(plm)
> fln <- FragmentLengthNormalization(ces)
> fln
FragmentLengthNormalization:
Data set: HapMap100K-bioc2007
Input tags: QN,RMA,A+B
Output tags: QN,RMA,A+B,FLN
Number of arrays: 6 (14.08MB)
Chip type: Mapping50K_Hind240,monocell
Algorithm parameters: (subsetToFit: NULL, .targetFunction: NULL)
Output path: plmData/HapMap100K-bioc2007,QN,RMA,A+B,FLN/Mapping50K_Hind
Is done: FALSE
RAM: 0.00MB
```

TASK: Fragment-length normalization

Normalize the chip-effects for fragment-length artifacts:

```
ces <- getChipEffectSet(plm)
fln <- FragmentLengthNormalization(ces)
cesN <- process(fln, verbose=TRUE)

theta <- extractMatrix(cesN)
thetaAvg <- rowMedians(theta)
M <- log2(theta/thetaAvg)

layout(matrix(1:6, ncol=3, byrow=TRUE))
for (kk in seq(ces)) {
  smoothScatter(fl, M[,kk], ylim=c(-1.5,1.5), main=kk)
  lines(smooth.spline(fl, M[,kk]), col="red", lwd=3)
}
```

Summary: data directories

```
plmData/
 +- HapMap100K-bioc2007,QN,RMA,A+B/
     +- Mapping50K_Hind240/
         +- .average-intensities-median-mad,34d27e9e3a5c610f3e49ef4b792
         +- NA06985,B5,3005533,chipEffects.cel
         +- NA06991,B6,3005533,chipEffects.cel
         +- NA06993,B4,4000092,chipEffects.cel
         +- NA06994,A7,3005533,chipEffects.cel
         +- NA07000,A8,3005533,chipEffects.cel
         +- NA07019,A12,4000092,chipEffects.cel
         +- probeAffinities.cel
 +- HapMap100K-bioc2007,QN,RMA,A+B,FLN/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533,chipEffects.cel
         +- NA06991,B6,3005533,chipEffects.cel
         +- NA06993,B4,4000092,chipEffects.cel
         +- NA06994,A7,3005533,chipEffects.cel
         +- NA07000,A8,3005533,chipEffects.cel
         +- NA07019,A12,4000092,chipEffects.cel
         +- probeAffinities.cel
```

Total copy numbers: Segmentation

```
> glad <- GladModel(cesN)
> glad
GladModel:
Name: HapMap100K-bioc2007
Tags: QN,RMA,A+B,FLN
Chip type (virtual): Mapping50K_Hind240
Path: gladData/HapMap100K-bioc2007,QN,RMA,A+B,FLN/Mapping50K_Hind240
Number of chip types: 1
Chip-effect set & reference file pairs:
Pair #1:
Chip-effect set:
CnChipEffectSet:
Name: HapMap100K-bioc2007
Tags: QN,RMA,A+B,FLN
Path: plmData/HapMap100K-bioc2007,QN,RMA,A+B,FLN/Mapping50K_Hind240
Chip type: Mapping50K_Hind240,monocell
Number of arrays: 6
Names: NA06985, NA06991, ..., NA07019
Time period: 2007-08-02 10:41:29 -- 2007-08-02 10:41:30
Total file size: 14.08MB
RAM: 0.01MB
Parameters: (probeModel: chr "pm", mergeStrands: logi TRUE,
             combineAlleles: logi TRUE)
Reference file:
<average across arrays>
```

Total copy numbers: Segmentation

```
> glad <- GladModel(cesN)
> fit(glad, arrays=1, chromosomes=19, verbose=TRUE)
Retrieving reference data files...
 No reference available.
 Calculating average chip effects...
 Calculating average chip effects...done
Retrieving reference data files...done
Using references:
[[1]]
CnChipEffectFile:
Name: .average-intensities-median-mad
Tags: 34d27e9e3a5c610f3e49ef4b7924d9de
Pathname: plmData/HapMap100K-bioc2007,QN,RMA,A+B,FLN/Mapping50K_Hind240/
  .average-intensities-median-mad,34d27e9e3a5c610f3e49ef4b7924d9de.CEL
File size: 2.35MB
RAM: 0.02MB
File format: v4 (binary; XDA)
Chip type: Mapping50K_Hind240,monocell
Timestamp: 2007-08-02 13:27:26
:
```

Total copy numbers: Segmentation

```
> glad <- GladModel(cesN)
> fit(glad, arrays=1, chromosomes=19, verbose=TRUE)
:
Parameters: (probeModel: chr "pm", mergeStrands: logi TRUE,
   combineAlleles: logi TRUE)
Reference tags: 34d27e9e3a5c610f3e49ef4b7924d9de
Chip-effect tags: B5,3005533
Array #1 ('NA06985') of 1 on chromosome 19...
Attempting to read CDF File: annotationData/chipTypes/Mapping50K_Hind240
 Mapping50K_Hind240,monocell.cdf
 :
 Saving to file...
  Pathname: gladData/HapMap100K-bioc2007,QN,RMA,A+B,FLN/Mapping50K_Hind2
  NA06985,B5,3005533,chr19,34d27e9e3a5c610f3e49ef4b7924d9de.xdr
 Saving to file...done
 Calling onFit() hooks...
 Calling onFit() hooks...done
Array #1 ('NA06985') of 1 on chromosome 19...done
```

TASK: Fit and plot GLAD segmentation

Plot the raw CN estimates and the GLAD segmentation:

```
> glad <- GladModel(cesN)
> fit(glad, arrays=1, chromosomes=19, verbose=TRUE)
> plot(glad, arrays=1, chromosome=19)
```

If you try to plot a non-fitted array/chromosome, it will be fitted automatically.

```
> plot(glad, arrays=1, chromosome=18)
```

Total copy numbers: Browse copy numbers

The Chromosome Explorer is a HTML based tool for browsing copy number results across arrays, chromosomes, and chip types.

```
> ce <- ChromosomeExplorer(glad)
ChromosomeExplorer:
Name: HapMap100K-bioc2007
Tags: QN,RMA,A+B,FLN
Number of arrays: 6
Path: reports/HapMap100K-bioc2007/QN,RMA,A+B,FLN/
      Mapping50K_Hind240/glad
RAM: 0.00MB
> process(ce, array=1, chromosomes=1:4, verbose=TRUE)
> display(ce)  # Requires Firefox
> process(ce, verbose=TRUE)  # 5 mins/array - Have a coffee!
```

TASK: Generate Chromosome Explorer report

Generate the results yourself:

```
ce <- ChromosomeExplorer(glad)
process(ce, array=1, chromosomes=19:22, verbose=TRUE)
display(ce)
process(ce, verbose=TRUE)  # 5 mins/array - Have a coffee!
```

Outline

Multiple chip types

```
annotationData/
 +- chipTypes/
     +- Mapping50K_Hind240/
     |   +- Mapping50K_Hind240.CDF
     +- Mapping50K_Xba240/
         +- Mapping50K_Xba240.CDF

rawData/
 +- HapMap100K-bioc2007/
     +- Mapping50K_Hind240/
     |   +- NA06985,B5,3005533.CEL
     |   +- NA06991,B6,3005533.CEL
     |   :
     |   +- NA07019,A12,4000092.CEL
     +- Mapping50K_Xba240/
         +- NA06985,B5,3005533.CEL
         +- NA06991,B6,3005533.CEL
         :
         +- NA07019,A12,4000092.CEL
```

Multiple chip types

Generate the results yourself:

```
> glad <- GladModel(list(cesHind, cesXba))
> ce <- ChromosomeExplorer(glad)
> process(ce)
```

Outline

PLM Quality Assessment

For unit (probeset) $j = 1, 2, \ldots, J$, the log-additive ("RMA") PLM is:

$$\log_2 y_{ik} = \log_2 \theta_i + \log_2 \phi_k + \varepsilon_{ik}$$

where y_{ik} is the PM signal for sample i and probe k; θ_i is the chip effect for this sample; ϕ_k is the probe affinity for this probe; and ε_{ik} is zero-mean noise.

```
> plm <- RmaPlm(csQN)
> plm
RmaPlm:
Data set: HapMap100K-bioc2007
Chip type: Mapping50K_Hind240
Input tags: QN
Output tags: QN,RMA
Parameters: (probeModel: chr "pm"; shift: num 0; flavor: chr "affyPLM")
Path: plmData/HapMap100K-bioc2007,QN,RMA/Mapping50K_Hind240
RAM: 0.00MB
```

Probe-level modelling - chip effects

PLM:

$$\log_2 y_{ik} = \log_2 \theta_i + \log_2 \phi_k + \varepsilon_{ik}$$

```
> plm <- RmaPlm(csQN)
> fit(plm, verbose=TRUE)
> plm
```

Summary: data directories

```
plmData/
 +- HapMap100K-bioc2007,QN,RMA/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533,chipEffects.cel
         +- NA06991,B6,3005533,chipEffects.cel
         +- NA06993,B4,4000092,chipEffects.cel
         +- NA06994,A7,3005533,chipEffects.cel
         +- NA07000,A8,3005533,chipEffects.cel
         +- NA07019,A12,4000092,chipEffects.cel
         +- probeAffinities.cel
 +- HapMap100K-bioc2007,QN,RMA,A+B/
     +- Mapping50K_Hind240/
         +- NA06985,B5,3005533,chipEffects.cel
         +- NA06991,B6,3005533,chipEffects.cel
         +- NA06993,B4,4000092,chipEffects.cel
         +- NA06994,A7,3005533,chipEffects.cel
         +- NA07000,A8,3005533,chipEffects.cel
         +- NA07019,A12,4000092,chipEffects.cel
         +- probeAffinities.cel
```

Probe-level modelling - chip effects

PLM:

$$\log_2 y_{ik} = \log_2 \theta_i + \log_2 \phi_k + \varepsilon_{ik}$$

```
> rs <- calculateResidualSet(plm, verbose=TRUE) # 30-60 sec
> rf <- getFile(rs, 2)
> rf
ResidualFile:
Name: NA06991
Tags: B6,3005533,residuals
Pathname: plmData/HapMap100K-bioc2007,QN,RMA/
          Mapping50K_Hind240/NA06991,B6,3005533,residuals.cel
File size: 24.41MB
RAM: 0.01MB
File format: v4 (binary; XDA)
Chip type: Mapping50K_Hind240
Timestamp: 2007-08-02 01:37:16
Parameters: (probeModel: chr "pm")
> plotImage(rf)
```

TASK$^{\text{EBImage}}$: Probe-level modelling - Browse residuals

```
> rs <- calculateResidualSet(plm, verbose=TRUE) # 30-60 sec
> ae <- ArrayExplorer(rs)
> setColorMaps(ae, "log2,log2abs,rainbow")
> ae
ArrayExplorer:
Name: HapMap100K-bioc2007
Tags: QN,RMA
Number of chip types: 1
Number of arrays: 6
Color maps: log2,log2abs,rainbow
Main path: reports/HapMap100K-bioc2007/QN,RMA
RAM: 0.00MB
> process(ae, verbose=TRUE)  # 3 min = 30 secs/array
> display(ae)
```