Managing Cardiogenics Genome-Wide Gene Expression Data: The *CardioExpress package*

Seraya Maouche, François Cambien

May 11, 2010

Contents

1 Overview

The *CardioExpress* package contains all gene expression data generated within the Cardiogenics project http://www.cardiogenics.eu/web/. Data was normalized within center and within cell type. In addition a dataset for all monocyte samples normalized together, all macrophages normalized together and a dataset for the whole study data is included in this package. *CardioExpress* is based on the *ExpressionSet* class, a standardized data structure defined in the package *Biobase* (part of the Bioconductor project) to represent genomic data. *ExpressionSet* is intended to store expression data, experimental data, phenotypic data and feature-level annotation informations. Expression data and phenotypic data are properly aligned. In addition, this class can be extended in many different ways (adding new slots, adding SNPs data,..).

2 Description of Cardiogenics gene expression data

Cardiogenics Gene expression data is a large-scale dataset which consists of 1,533 RNA samples (849 monocyte RNA samples and 684 macrophages). All arrays outliers were filtered out and not included in this package. Data were collected in 5 centers; Leicester (150 monocytes and 123 macrophages), Lubeck (90 monocytes and 70 macrophages), Regensburg (113 monocytes and 99 macrophages), Cambridge (435 monocytes and 336 macrophages) and Paris (61 monocytes and 56 macrophages). Microarray experiments were carried out in INSERM UMRS 937 (Prof Cambien's lab) using Illumina BeadChip Human-Ref v3.0 arrays and data are centralized within the intcardio server https://www.intcardio.org/wp5/ which is hosted by The Wellcome Trust Sanger Institute (Dr Kate Rice).

3 Installing *CardioExpress*

CardioExpress could be installed using the R CMD INSTALL command. Please use 'R CMD INSTALL –help' to obtain more help.

4 Loading required packages

We start by loading required packages

```
> require(Biobase)
> require(GSEAlm)
```

If these two packages are not already installed, the best way to do it is to use the Bioconductor biocLite () function:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite(c("Biobase", "GSEAlm"))

Using R version 2.10.1, biocinstall version 2.5.10.
Installing Bioconductor version 2.5 packages:
[1] "Biobase" "GSEAlm"
Please wait...
```

5 Loading *CardioExpress* package

The following command will load the *CardioExpress* package into R:

```
> library(CardioExpress)
```

Once this is done you need to load a specific dataset. For example, if you need to perform eQTL analysis on Leicester monocytes data, you need do load the 'LeicesterMono' dataset using data('LeicesterMono'). The different datasets included in the *CardioExpress* package are listed bellow.

> data(package='CardioExpress')
> CambridgeMacro
> CambridgeMono
> LeicesterMacro
> LeicesterMono
> LubeckMacro
> LubeckMono
> ParisMacro
> ParisMono
> RegensburgMacro
> RegensburgMono
> allWP5Data
> allWP5macrophagesData
> allWP5monocytesData

6 Illustration

6.1 Loading a specific dataset

```
> options(width=60)
> # Load allWP5Data dataset
> library(CardioExpress)
```

```
> data(CambridgeMono)
> class(CambridgeMono)

[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"

> # Use print (or show) to display summary information
> show(CambridgeMono)

ExpressionSet (storageMode: lockedEnvironment)
assayData: 24526 features, 435 samples
  element names: exprs
phenoData
  sampleNames: X4298650009_A, X4298650009_B, ..., X456464706
  2_H  (435 total)
  varLabels and varMetadata description:
    PID: Patient ID (CXMA code)
    Centre: Recruitment center(01:Leicester;02:Lubeck;03:Reg
  ensburg;04:Cambridge; 05:Paris
    ...: ...
    contaminated: Contaminated
    (48 total)
featureData
  featureNames: ILMN_1762337, ILMN_2055271, ..., ILMN_213753
  6  (24526 total)
  fvarLabels and fvarMetadata description:
    Probe_Id: Illumina probe ID
    Entrez_Gene_ID: Entrez_Gene_ID
    ...: ...
    FilteringMacro: Filtering on absence in all macrophage R
  NAs
    (52 total)
experimentData: use 'experimentData(object)'
Annotation: IlluminaHumanRef8v3.0

> # Accessing MIAME object
> experimentData(CambridgeMono)

Experiment data
  Experimenter name: INSERM U937 team
  Laboratory: François Cambien's Lab
  Contact information: seraya.maouche@upmc.fr
```

```
  Title: Cardiogenics WP5 gene expression data
  URL: http://genecanvas.idf.inserm.fr
  PMIDs:

  Abstract: A 3 word abstract is available. Use 'abstract' method.
```

```
> # Obtaining a description of the ExpressionSet class
> # help("ExpressionSet-class")
> # "?"(class,ExpressionSet)
> # slotNames("eSet")
>
> # Sample names
> CambridgeMono$PID[1:4]
```

```
[1] CXMA-04-020 CXMA-04-029 CXMA-04-024 CXMA-04-037
866 Levels: CXMA-01-001 CXMA-01-002 ... CXMA-05-078
```

```
> # Array names
> sampleNames(CambridgeMono)[1:5]
```

```
[1] "X4298650009_A" "X4298650009_B" "X4298650009_C"
[4] "X4298650009_D" "X4298650009_E"
```

6.2 Accessing gene expression data

Gene expression data is stored into an object named 'exprs'.

```
> options(width=60)
> exprs(CambridgeMono)[1:10,1:4]
```

```
             X4298650009_A X4298650009_B X4298650009_C
ILMN_1762337      5.968208      5.905357      6.053652
ILMN_2055271      6.232482      6.187500      6.271421
ILMN_2383229      5.878106      6.039864      6.042416
ILMN_1806310      6.034579      6.043188      6.042303
ILMN_1779670      6.165598      6.125860      6.101373
ILMN_2321282      6.509377      6.605070      6.542265
ILMN_1772582      6.094197      6.111526      6.017685
ILMN_1717783      6.126431      6.094651      5.937685
ILMN_1814316      6.098873      6.066038      6.033029
ILMN_2359168      6.236354      6.222032      6.156214
             X4298650009_D
ILMN_1762337      6.055006
```

```
ILMN_2055271      6.154614
ILMN_2383229      6.057834
ILMN_1806310      6.085551
ILMN_1779670      6.140921
ILMN_2321282      6.521472
ILMN_1772582      6.052058
ILMN_1717783      5.925881
ILMN_1814316      6.052256
ILMN_2359168      6.202998
```

```
> # Exporting gene expression matrix
> ExpressMatrix <- exprs(CambridgeMono)
> dim(ExpressMatrix)
```

```
[1] 24526   435
```

```
> colnames(ExpressMatrix)[1:20]
```

```
 [1] "X4298650009_A" "X4298650009_B" "X4298650009_C"
 [4] "X4298650009_D" "X4298650009_E" "X4298650009_F"
 [7] "X4298650009_G" "X4298650017_A" "X4298650017_B"
[10] "X4298650017_C" "X4298650017_D" "X4298650017_E"
[13] "X4298650017_F" "X4298650017_G" "X4298650017_H"
[16] "X4298650022_A" "X4298650022_B" "X4298650022_C"
[19] "X4298650022_D" "X4298650022_E"
```

```
> head(ExpressMatrix[,1:4])
```

```
             X4298650009_A X4298650009_B X4298650009_C
ILMN_1762337      5.968208      5.905357      6.053652
ILMN_2055271      6.232482      6.187500      6.271421
ILMN_2383229      5.878106      6.039864      6.042416
ILMN_1806310      6.034579      6.043188      6.042303
ILMN_1779670      6.165598      6.125860      6.101373
ILMN_2321282      6.509377      6.605070      6.542265
             X4298650009_D
ILMN_1762337      6.055006
ILMN_2055271      6.154614
ILMN_2383229      6.057834
ILMN_1806310      6.085551
ILMN_1779670      6.140921
ILMN_2321282      6.521472
```

```
> # Extracting expression levels for a list of probes
> listProbe <- c("ILMN_1684873")
> expressILMN_1684873 <- CambridgeMono[which(as.character(featureNames
+                         (CambridgeMono)) %in% listProbe),]
```

```
> boxplot(t(exprs(expressILMN_1684873)) ~ expressILMN_1684873$gender,
+     data = expressILMN_1684873, col = "darkgrey",
+     ylab = "Normalized expression level", names = c("Male",
+         "Female"), main = "ARSD gene", cex = 1,
+     cex.lab = 1.2, lwd = 2, cex.axis = 1.8)
```

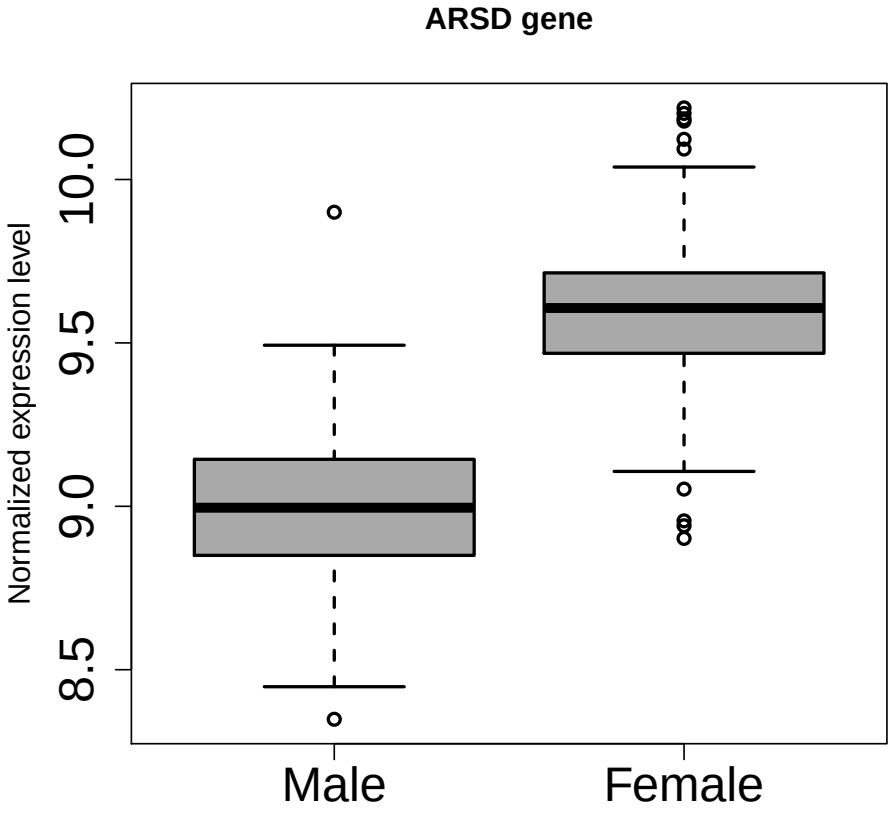

Figure 1: Gene expression-gender association

6.3 Accessing phenotypic data

Covariates describing the samples are saved to a phenoData object

```
> options(width=60)
> names(pData(CambridgeMono))
```

```
 [1] "PID"                     "Centre"
 [3] "Disease_Status"          "cell_Type"
 [5] "HasGenotypes"            "individual"
 [7] "Sample_Batch"            "hybridization_Date"
 [9] "dateHyb"                 "monoAndMacro"
[11] "contamination"           "age"
[13] "gender"                  "BMI"
[15] "height"                  "weight"
[17] "SBP"                     "DBP"
[19] "Diabetes"                "Hyperlipidemia"
[21] "hypertension"            "smoking"
[23] "smoke"                   "LDL"
[25] "HDL"                     "cholesterol"
[27] "glucose"                 "Hb"
[29] "Lymphocytes"             "Monocytes"
[31] "Eosinophils"             "Basophils"
[33] "Platelet_count"          "RBC"
[35] "Neutrophils"             "WBC"
[37] "CD14plusCD16Minus"       "CD14plusCD16plus"
[39] "RecoveryofCD14pluscells" "PercentofCD14pluscells"
[41] "CRP"                     "cellsT.NK"
[43] "cellsCD19"               "cellsCD66b"
[45] "cellsCD14"               "cellsEB"
[47] "cellsMK1"                "contaminated"
```

```
> head(pData(CambridgeMono))[,1:8]
```

```
                     PID Centre Disease_Status cell_Type
X4298650009_A CXMA-04-020      4              0      mono
X4298650009_B CXMA-04-029      4              0      mono
X4298650009_C CXMA-04-024      4              0      mono
X4298650009_D CXMA-04-037      4              0      mono
X4298650009_E CXMA-04-062      4              0      mono
X4298650009_F CXMA-04-065      4              0      mono
              HasGenotypes individual Sample_Batch
X4298650009_A            1        420           42
X4298650009_B            1        429           42
```

```
X4298650009_C            1        424          42
X4298650009_D            1        437          42
X4298650009_E            1        462          42
X4298650009_F            1        465          42
             hybridization_Date
X4298650009_A         2008/04/06
X4298650009_B         2008/04/06
X4298650009_C         2008/04/06
X4298650009_D         2008/04/06
X4298650009_E         2008/04/06
X4298650009_F         2008/04/06
```

```
> # Smoking status
> table(CambridgeMono$smoke)
```

```
          current exSmokerLess12wks exSmokerMore12wks
               26                 0                140
            never
              233
```

```
> # Summary BMI
> summary(CambridgeMono$age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  32.00   47.00   54.00   53.43   59.00   65.00
```

```
> # Meta-data on phenotypic data
> options(width=60)
> varMetadata(CambridgeMono)[1:15,]
```

```
 [1] "Patient ID (CXMA code)"
 [2] "Recruitment center(01:Leicester;02:Lubeck;03:Regensburg;04:Cambridge; 05:Paris"
 [3] "Disease status (0: Healthy; 1:MI/CAD patient"
 [4] "Cell ype (monocyte or macrophage)"
 [5] "Indicates whether this individual has SNP data"
 [6] "Individual code"
 [7] "Sample batch"
 [8] "Hybridization date"
 [9] "date of hybridization coded as factor"
[10] "Indicates whether the indiviudal have both mono and macro RNAs"
[11] "Presence of other blood cells contamination:1, absence of contamination:0"
[12] "Age"
[13] "Gender (1:male; 2:female)"
[14] "Body Mass Index"
[15] "Height"
```

```
> # Check whether phenotype data and gene expression are well matched
> all(rownames(pData(CambridgeMono))==colnames(exprs(CambridgeMono)))

[1] TRUE
```

6.4 Accessing contamination data

```
> options(width=40)
> contamEstimation <-  pData(CambridgeMono)[,c("cellsT.NK","cellsCD19","cellsCD66b",
+                             "cellsCD14","cellsEB","cellsMK1","contaminated")]
> head(contamEstimation)

                cellsT.NK   cellsCD19
X4298650009_A 0.000000000 0.000000000
X4298650009_B 0.000000000 0.000000000
X4298650009_C 0.004087730 0.000000000
X4298650009_D 0.000000000 0.060257394
X4298650009_E 0.007396152 0.000000000
X4298650009_F 0.013123395 0.001432861
               cellsCD66b cellsCD14
X4298650009_A 0.008040016 0.9919600
X4298650009_B 0.013618407 0.9863816
X4298650009_C 0.019307329 0.9712475
X4298650009_D 0.064212458 0.8755301
X4298650009_E 0.027033622 0.9526664
X4298650009_F 0.097079766 0.8742464
                  cellsEB cellsMK1
X4298650009_A 0.000000000        0
X4298650009_B 0.000000000        0
X4298650009_C 0.005357455        0
X4298650009_D 0.000000000        0
X4298650009_E 0.012903803        0
X4298650009_F 0.014117576        0
              contaminated
X4298650009_A            0
X4298650009_B            0
X4298650009_C            0
X4298650009_D            0
X4298650009_E            0
X4298650009_F            0

> ## CD14+/CD16+ versus CD14+/CD16-
> NeedToBeChecked  <- which(CambridgeMono$CD14plusCD16plus >
```

```
+                                CambridgeMono$CD14plusCD16Minus)
> CambridgeMono[,NeedToBeChecked]$PID

 [1] CXMA-04-118 CXMA-04-311 CXMA-04-307
 [4] CXMA-04-306 CXMA-04-333 CXMA-04-341
 [7] CXMA-04-233 CXMA-04-234 CXMA-04-354
[10] CXMA-04-436 CXMA-04-441 CXMA-04-440
[13] CXMA-04-443 CXMA-04-442 CXMA-04-452
[16] CXMA-04-315 CXMA-04-328 CXMA-04-370
[19] CXMA-04-317 CXMA-04-462 CXMA-04-461
[22] CXMA-04-310 CXMA-04-326 CXMA-04-321
[25] CXMA-04-237 CXMA-04-402
866 Levels: CXMA-01-001 ... CXMA-05-078
```

6.5 Accessing probes annotation

The probe level data is contained in the featureData object.

```
> options(width=60)
> featureNames(CambridgeMono)[1:10]

 [1] "ILMN_1762337" "ILMN_2055271" "ILMN_2383229"
 [4] "ILMN_1806310" "ILMN_1779670" "ILMN_2321282"
 [7] "ILMN_1772582" "ILMN_1717783" "ILMN_1814316"
[10] "ILMN_2359168"

> fvarLabels(CambridgeMono)

 [1] "Probe_Id"               "Entrez_Gene_ID"
 [3] "Illum_Transcript"       "Illum_RefSeqID"
 [5] "Illum_GI"               "Illum_Symbol"
 [7] "Illum_ProbeType"        "Illum_ProbeStart"
 [9] "Illum_Chrom"            "Illum_ChrOrientation"
[11] "Illum_ProbeCoordinates" "Illum_Cytoband"
[13] "NCBI_GeneLowPoint"      "NCBI_GeneHighPoint"
[15] "NCBI_Orientation"       "NCBI_Chromo"
[17] "NCBI_GeneSymbol"        "NCBI_Build"
[19] "NCBI_cytoband"          "LeicesterMono_150"
[21] "LubeckMono_90"          "RegensburgMono_113"
[23] "ParisMono_61"           "CambridgeMono_435"
[25] "MonoConco"              "LeicesterMacro_123"
[27] "LubeckMacro_70"         "RegensburgMacro_99"
```

```
> hist(CambridgeMono$cellsCD14, col = "blue")
```

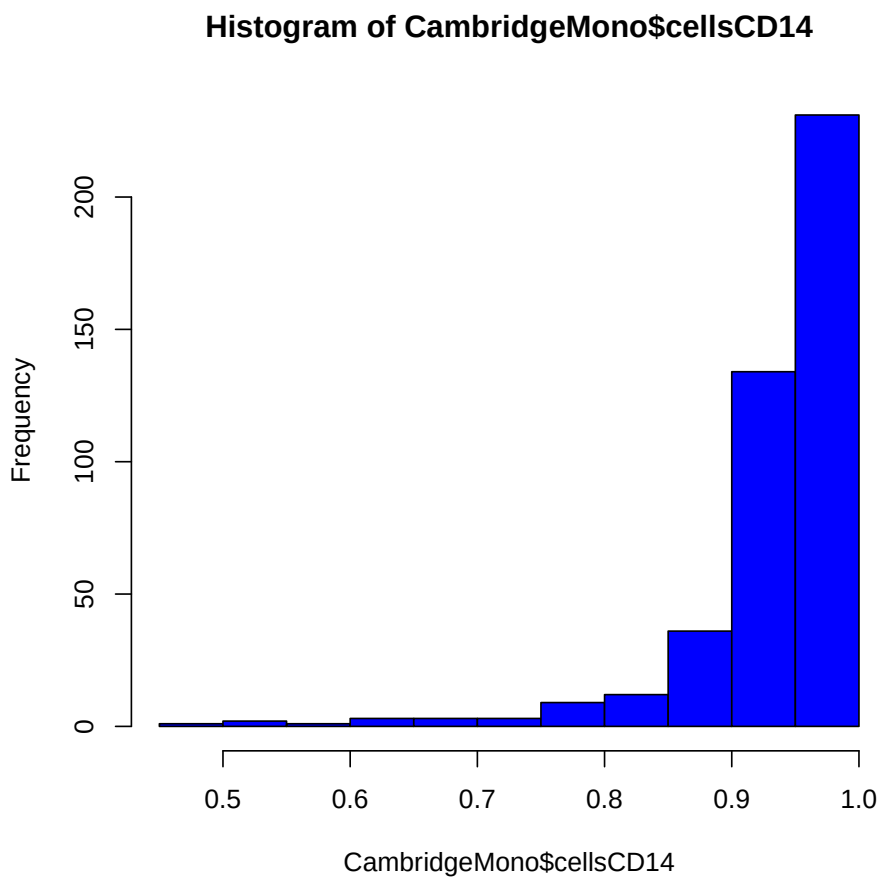

Figure 2: Percentage of blood cells

```
[29] "ParisMacro_56"          "CambridgeMacro_336"
[31] "MacroConco"             "CAD_mono414"
[33] "CAD_macro348"           "Healthy_Mono435"
[35] "Healthy_Macro336"       "ContamMono99"
[37] "ContamMacro"            "NotContamMono"
[39] "NotContamMacro"         "Male_mono541"
[41] "female_mono308"         "male_macro449"
[43] "Female_macro235"        "All_Mono849"
[45] "All_Macro684"           "AllSample_1533"
[47] "MonoPresenceAbsence"    "MacroPresenceAbsence"
[49] "All_PresenceAbsence"    "FilteringAll"
```

```
[51] "FilteringMono"          "FilteringMacro"
```

```
> # Meta-data on probe annotation data:
> fvarMetadata(CambridgeMono)[1:20,1]
```

```
 [1] "Illumina probe ID"
 [2] "Entrez_Gene_ID"
 [3] "Illum_Transcript"
 [4] "Illum_RefSeqID"
 [5] "Illum_Gene ID"
 [6] "Illum_Gene Symbol"
 [7] "Illum_ProbeType"
 [8] "Illumina ProbeStart"
 [9] "Illumina Chromosome"
[10] "Illum_ChrOrientation"
[11] "Illum_ProbeCoordinates"
[12] "Illum_Cytoband"
[13] "NCBI_GeneLowPoint"
[14] "NCBI Gene High Point"
[15] "NCBI Orientation"
[16] "NCBI Chromosome"
[17] "NCBI Gene Symbol"
[18] "NCBI Build"
[19] "NCBI cytoband"
[20] "Leicester Monocytes - Percentage of detection"
```

6.6 Probes filtering

Filtering on detection call

```
> options(width=40)
> presentProbes <- which(featureData(CambridgeMono)$FilteringMono=="Not Filtered")
> CambridgeMono.Filt <- CambridgeMono[presentProbes,]
> # Show the number of probe after filtering on detection call in monocytes
> dim(CambridgeMono.Filt)
```

```
Features  Samples
   11479      435
```

6.7 Subjects having GWAS data

Show which sample have genotype data

```
> HasGenotypes  <- which(as.character(CambridgeMono$HasGenotypes) ==1)
> HasGenotypes_all  <-  CambridgeMono[,HasGenotypes]
> dim(HasGenotypes_all)

Features  Samples
   24526      395
```

6.8 Example : preparing data for eQTL analyses

The script bellow provides an example of how this package could be used to prepare files for eQTL analyses. The script allows to read PED and MAP files and to perform filtering of gene expression and SNP data. In addition, data could be adjusted on gender, age and contamination and residuals could be exported for statistical analyses.

```
> library(CardioExpress)
> data(CambridgeMono)
> sampleNames(CambridgeMono) <- CambridgeMono$PID
> # read PED and MAP file  (Filtered on Call Rate (kept rs with CR > 0.95)
> #  and Intensity Only SNPs were removed
> PEDfile      <- 'Filt_CommonSNPChrom24.ped'
> setwd(PEDdir)
> pedFile      <- read.table(PEDfile, header = FALSE, sep = "\t")
> dim(pedFile)
> ## Filter on center
> IDgeno            <- grep('CXMA-04',pedFile[,1],value=TRUE)
> pedFile           <- subset(pedFile, pedFile[,1] %in% IDgeno)
> HasGenotypes      <- which(as.character(CambridgeMono$HasGenotypes) ==1)
> PhenoCamMono      <-  CambridgeMono[,HasGenotypes]
> dim(PhenoCamMono)
> ## Filter out absent probes in monocytes
> presentProbes    <- which(featureData(PhenoCamMono)$FilteringMacro=='Not Filtered')
> PhenoCamMono     <- PhenoCamMono[presentProbes,]
> dim(PhenoCamMono)
> table(PhenoCamMono$gender)
> ## Filter out genes associated with contamination
> contam          <- grep('contam',featureData(PhenoCamMono)$MonoConco,value=TRUE)
> contamGenes     <- which(!featureData(PhenoCamMono)$MonoConco %in% contam)
> PhenoCamMono    <- PhenoCamMono[contamGenes,]
> dim(PhenoCamMono)
> lm1 = lmPerGene(PhenoCamMono,~ age + gender)
> PhenoCamMacro_resid  = getResidPerGene(lm1)
> exprs(PhenoCamMono_resid)[1:10,1:4]
> # Filter PED file to remove individuals without expression data
```

```
> pedFile        <- subset(pedFile, pedFile[,1] %in% sampleNames(PhenoCamMono))
> dim(pedFile)
> # Construct pheno file
> temp             <- data.frame(PhenoCamMono$PID,PhenoCamMono$PID,
+                        PhenoCamMono$Centre,PhenoCamMono$gender,PhenoCamMono$age)
> colnames(temp)   <- c("sampleID","SubjectID","Center","Gender","Age")
> rownames(temp)   <- temp[,1]
> phenoCamMono     <- merge(temp,t(exprs(PhenoCamMono)),by.x="row.names",
+                                by.y="row.names")
> phenoCamMono_resid <- merge(temp,t(exprs(PhenoCamMono_resid)),by.x="row.names",
+                                by.y="row.names")
> phenoCamMono       <- phenoCamMono[,-1]
> phenoCamMono_resid <- phenoCamMono_resid[,-1]
> PEDorder           <- as.character(pedFile[,1])
> newOrder           =  match(PEDorder,phenoCamMono[,1])
> phenoCamMono       =  phenoCamMono[newOrder,]
> phenoCamMono_resid =  phenoCamMono_resid[newOrder,]
> all(as.character(pedFile[,1])==as.character(phenoCamMono[,1]))
> all(as.character(pedFile[,1])==as.character(phenoCamMono_resid[,1]))
> phenoCamMono       <- rbind(phenoCamMono[1,],phenoCamMono)
> phenoCamMono_resid <- rbind(phenoCamMono_resid[1,],phenoCamMono_resid)
> setwd(outDirGeneDetected)
> write.table(phenoCamMono, file = "phenoCamMono.txt", append = FALSE, quote = FALSE,
+             sep = "\t",row.names = FALSE,col.names = TRUE)
> write.table(phenoCamMono_resid, file = "phenoCamMono_resid.txt", append = FALSE,
+             quote = FALSE, sep = "\t",row.names = FALSE,col.names = TRUE)
> phenoCamMono <- phenoCamMono[-1,]
> for (i in 1:24) {
+ PEDfile          <- paste("Filt_CommonSNPChrom",i,".ped",sep="")
+ setwd(PEDdir)
+ pedFile          <- read.table(PEDfile, header = FALSE, sep = "\t")
+ dim(pedFile)
+ pedFile          <- subset(pedFile, pedFile[,1] %in% phenoCamMono[,1])
+ dim(pedFile)
+ PhenoOrder       <- as.character(phenoCamMono[,1])
+ newOrder         =  match(PhenoOrder,pedFile[,1])
+ pedFile          =  pedFile[newOrder,]
+ setwd(outDirGeneDetected)
+ write.table(pedFile, file =paste("CamMono_",PEDfile,sep=""), append = FALSE, quote
+ sep = "\t",eol= "\n", na = "NA", dec = ".", row.names = FALSE,col.names = FALSE)
+ }
```

7 What was used to create this document

This document was generated using Sweave (Leisch, 2002) which provides a convenient way of combining R and LATEX.

```
> sessionInfo()

R version 2.10.1 (2009-12-14)
i386-pc-mingw32

locale:
[1] LC_COLLATE=French_France.1252
[2] LC_CTYPE=French_France.1252
[3] LC_MONETARY=French_France.1252
[4] LC_NUMERIC=C
[5] LC_TIME=French_France.1252

attached base packages:
[1] stats     graphics  grDevices
[4] utils     datasets  methods
[7] base

other attached packages:
[1] CardioExpress_0.99.4
[2] GSEAlm_1.6.0
[3] Biobase_2.6.1

loaded via a namespace (and not attached):
[1] tools_2.10.1
```