

# Discordant test

Stefanie Peschel

2023-04-12

## Discordant examples

### Example from the discordant help page

```
library(discordant)

# Load Data
data(TCGA_GBM_miRNA_microarray)
data(TCGA_GBM_transcript_microarray)
print(colnames(TCGA_GBM_transcript_microarray)) # look at groups

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"

groups <- c(rep(1,10), rep(2,20))

## DC analysis on only transcripts pairs

vectors <- createVectors(TCGA_GBM_transcript_microarray,
                        groups = groups)
result <- discordantRun(vectors$v1, vectors$v2,
                       TCGA_GBM_transcript_microarray)

classmat <- cbind(vectors[[1]], vectors[[2]], result$classVector)
colnames(classmat) <- c("corr1", "corr2", "class")
```

The class matrix as stated in the discordant paper, where rows are the classes of group 1 and columns the classes of group 2.

```
classtab <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)
dimnames(classtab) <- list(c("0", "-", "+"), c("0", "-", "+"))
classtab

##  0 - +
## 0 1 2 3
## - 4 5 6
## + 7 8 9
```

## Single steps

```
v1 <- vectors$v1
v2 <- vectors$v2
x <- TCGA_GBM_transcript_microarray

v1 <- fishersTrans(v1)
v2 <- fishersTrans(v2)

x <- Biobase::exprs(x)

pdata <- cbind(v1, v2)
param1 <- sd(v1)
param2 <- sd(v2)
class <- cbind(discordant:::.assignClass(v1, param1, 3),
               discordant:::.assignClass(v2, param2, 3))

pd <- discordant:::em.normal.partial.concordant(pdata, class, 3)
```

```
mu_sigma <- pd$mu_sigma
rownames(mu_sigma) <- c("mu", "sigma")
colnames(mu_sigma) <- c(0, "-", "+")
mu_sigma
```

### Estimated parameters for group 1

```
##           0           -           +
## mu    -0.03217065 -0.68171580 0.5956348
## sigma 0.06045809 0.03789554 0.1400244
```

```
nu_tau <- pd$nu_tau
rownames(nu_tau) <- c("nu", "tau")
colnames(nu_tau) <- c(0, "-", "+")
nu_tau
```

### Estimated parameters for group 2

```
##           0           -           +
## nu    0.03105002 -0.339605967 0.40579454
## tau 0.03156240 0.008719114 0.03565425
```

```
range(vectors$v1)
```

### Range of the correlations

```
## [1] -0.7939394 0.9151515
```

```
range(vectors$v2)
```

```
## [1] -0.4842105 0.6812030
```

### Assigned classes

The matrices contain the correlations in the two groups, the initially assigned mixture components (possible values: 0 = "0", 1 = "-", 2 = "+"), and the finally assigned class (values from 1 to 9).

```
classmat <- cbind(v1, v2, class, result$classVector)
colnames(classmat) <- c("corr1", "corr2", "comp1", "comp2", "class")
classmat <- as.data.frame(classmat)

for (i in 1:9) {
  print(head(classmat[classmat$class == i, ], 15))
  print("-----")
}
```

```
##                corr1      corr2 comp1 comp2 class
## A_23_P138644_A_23_P24296    0.38807205 0.220024438    0    0    1
## A_23_P138644_A_32_P82889   -0.04867084 -0.009022801    0    0    1
## A_23_P138644_A_23_P163455  -0.03031231 0.243816930    0    0    1
## A_23_P138644_A_23_P4561   -0.31953998 -0.052680258    0    0    1
## A_23_P138644_A_23_P78392  -0.16512084 0.079868643    0    0    1
## A_23_P138644_A_24_P936860  -0.36027308 0.101094880    0    0    1
## A_23_P138644_A_32_P131011 -0.30619291 0.095021801    0    0    1
## A_23_P138644_AK094413_1_1546 -0.17760164 0.240629474    0    0    1
## A_23_P138644_AK098212_1_3017 -0.53725737 0.064021139    1    0    1
## A_23_P138644_VIRS_10002627  0.00606068 -0.170041174    0    0    1
## A_23_P24296_A_24_P571870   -0.43088495 -0.057205176    0    0    1
## A_23_P24296_A_32_P71885   -0.04244972 -0.143841036    0    0    1
## A_23_P24296_A_23_P105264  -0.55335475 0.024064795    1    0    1
## A_23_P24296_A_23_P94009   -0.47524105 0.168493917    1    0    1
## A_23_P24296_A_24_P31583    0.40354779 0.003007528    0    0    1
## [1] "-----"
##                corr1      corr2 comp1 comp2 class
## A_23_P138644_A_23_P105264  -0.8613833 -0.1561439    1    0    2
## A_23_P24296_A_23_P4561   -1.0819995 0.2090062    1    0    2
## A_24_P571870_A_24_P936860  -0.6931472 0.2390375    1    0    2
## A_32_P82889_A_32_P131011  -0.5970637 0.1392388    1    0    2
## A_23_P163455_A_23_P4561   -0.6744184 -0.1561439    1    0    2
## AK098212_1_3017_VIRS_10008594 -0.6205562 -0.1981141    1    0    2
## [1] "-----"
##                corr1      corr2 comp1 comp2 class
## A_23_P24296_A_24_P345312  0.5058005 -0.063242074    2    0    3
## A_23_P24296_A_23_P163455  0.9626454 0.134642354    2    0    3
## A_24_P345312_A_32_P71885  0.7319263 0.061732435    2    0    3
## A_24_P345312_A_32_P82889  0.4693723 -0.036105907    2    0    3
## A_24_P345312_A_23_P163455  1.0195838 0.003007528    2    0    3
## A_32_P71885_VIRS_10002627  1.4884021 0.039117685    2    0    3
```

```

## A_32_P82889_AK098212_1_3017 0.6865288 -0.054208738 2 0 3
## A_23_P105264_A_24_P936860 0.7726795 -0.285725123 2 1 3
## A_23_P105264_A_32_P131011 0.8157084 -0.255011850 2 0 3
## A_23_P163455_A_23_P78392 0.7938882 -0.099575931 2 0 3
## A_23_P163455_A_24_P31583 0.8655211 0.021055743 2 0 3
## A_23_P163455_A_32_P204007 1.0195838 -0.052680258 2 0 3
## A_23_P414654_VIRS_10002627 0.6560932 0.072306198 2 0 3
## A_23_P4561_A_23_P94009 0.5533547 -0.018047072 2 0 3
## A_23_P78392_A_32_P204007 0.6205562 0.156143942 2 0 3
## [1] "-----"
##
## corr1 corr2 comp1 comp2 class
## A_23_P138644_A_24_P345312 -0.26675852 -0.4636849 0 1 4
## A_23_P138644_A_24_P571870 -0.14030745 -0.2598287 0 0 4
## A_23_P138644_A_32_P71885 -0.09116078 -0.2938933 0 1 4
## A_23_P138644_A_23_P414654 -0.29295083 -0.3761950 0 1 4
## A_23_P138644_A_24_P31583 0.12218555 -0.4111795 0 1 4
## A_23_P138644_A_32_P204007 -0.15269082 -0.5284698 0 1 4
## A_23_P138644_VIRS_10008594 -0.15269082 -0.2792172 0 1 4
## A_23_P24296_A_23_P414654 0.03031231 -0.2598287 0 0 4
## A_24_P345312_A_32_P131011 -0.15269082 -0.2873558 0 1 4
## A_24_P345312_AK094413_1_1546 0.31953998 -0.2694983 0 0 4
## A_32_P71885_A_32_P82889 0.01823919 -0.2971713 0 1 4
## A_32_P71885_A_23_P163455 0.10339721 -0.4326323 0 1 4
## A_32_P71885_A_32_P131011 -0.01818382 -0.3186334 0 1 4
## A_32_P71885_AK094413_1_1546 -0.31953998 -0.3120002 0 1 4
## A_32_P71885_AK098212_1_3017 -0.19013743 -0.3643085 0 1 4
## [1] "-----"
##
## corr1 corr2 comp1 comp2 class
## AK094413_1_1546_VIRS_10008594 -0.7520387 -0.3454462 1 1 5
## [1] "-----"
## [1] corr1 corr2 comp1 comp2 class
## <0 rows> (or 0-length row.names)
## [1] "-----"
##
## corr1 corr2 comp1 comp2 class
## A_24_P345312_A_23_P105264 0.06676570 0.542313 0 2 7
## A_24_P345312_VIRS_10008594 -0.00606068 0.542313 0 2 7
## [1] "-----"
##
## corr1 corr2 comp1 comp2 class
## A_23_P138644_A_23_P94009 -0.7726795 0.4129533 1 2 8
## A_23_P4561_A_23_P78392 -0.6205562 0.3471376 1 2 8
## A_23_P78392_VIRS_10008594 -0.5533547 0.3675875 1 2 8
## [1] "-----"
##
## corr1 corr2 comp1 comp2 class
## A_23_P24296_A_32_P82889 0.2546037 0.6039434 0 2 9
## A_23_P24296_A_23_P78392 0.7726795 0.5871178 2 2 9
## A_23_P24296_A_32_P204007 0.5533547 0.4182898 2 2 9
## A_23_P24296_AK094413_1_1546 0.5214105 0.8313552 2 2 9
## A_24_P345312_A_23_P414654 0.4308849 0.3454462 0 2 9
## A_24_P345312_A_24_P31583 0.9276292 0.4489708 2 2 9
## A_24_P345312_A_32_P204007 1.4884021 0.4526321 2 2 9
## A_24_P345312_VIRS_10002627 1.0195838 0.6745577 2 2 9
## A_24_P571870_A_23_P414654 0.4602685 0.3020998 2 2 9
## A_24_P571870_A_23_P94009 0.2281187 0.4599889 0 2 9
## A_32_P71885_A_23_P414654 0.6560932 0.3883360 2 2 9

```

```
## A_32_P71885_VIRS_10008594  0.2027326 0.3918248    0    2    9
## A_32_P82889_A_23_P78392   0.3823463 0.6676905    0    2    9
## A_32_P82889_A_32_P204007  0.7451086 0.2922567    2    2    9
## A_32_P82889_AK094413_1_1546 0.4693723 0.8011186    2    2    9
## [1] "-----"
```

Class assignments:

- Class 2: negative in group 1 and zero in group 2
- Class 3: positive in group 1 and zero in group 2
- Class 4: zero in group 1 and negative in group 2
- Class 7: zero in group 1 and positive in group 2
- ...

Thus, the columns of the class matrix represent Group 1 and the rows represent Group 2, which is contrary to the definition in the discordant manuscript.

## Differential network

### Network construction

```
library("NetCoMi")
```

```
## Loading required package: SpieEasi
```

```
##
```

```
# Load data sets from American Gut Project (from SpieEasi package)
data("amgut2.filt.phy")

# Split data into two groups: with and without seasonal allergies
amgut_season_yes <- phyloseq::subset_samples(amgut2.filt.phy,
                                             SEASONAL_ALLERGIES == "yes")
amgut_season_no  <- phyloseq::subset_samples(amgut2.filt.phy,
                                             SEASONAL_ALLERGIES == "no")

amgut_season_yes
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 138 taxa and 121 samples ]
## sample_data() Sample Data: [ 121 samples by 166 sample variables ]
## tax_table() Taxonomy Table: [ 138 taxa by 7 taxonomic ranks ]
```

```
amgut_season_no
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 138 taxa and 163 samples ]
## sample_data() Sample Data: [ 163 samples by 166 sample variables ]
## tax_table() Taxonomy Table: [ 138 taxa by 7 taxonomic ranks ]
```

```

# Filter the 121 samples (sample size of the smaller group) with highest
# frequency to make the sample sizes equal and thus ensure comparability.
n_yes <- phyloseq::nsamples(amgut_season_yes)

duo_net <- netConstruct(data = amgut_season_yes,
  data2 = amgut_season_no,
  filtTax = "highestFreq",
  filtTaxPar = list(highestFreq = n_yes),
  filtSamp = "totalReads",
  filtSampPar = list(totalReads = 1000),
  measure = "spring",
  measurePar = list(nlambda=10, rep.num=10, thresh = 0.15,
    Rmethod = "approx"),
  normMethod = "mclr",
  zeroMethod = "none",
  sparsMethod = "none",
  dissFunc = "signed",
  verbose = 3,
  seed = 123456)

```

```

## Checking input arguments ...

## Done.

## Infos about changed arguments:

## Normalization ignored for measure "spring".

## Data filtering ...

## 15 samples removed in data set 1.

## 16 samples removed in data set 2.

## 22 taxa removed in each data set.

## 116 taxa and 106 samples remaining in group 1.

## 116 taxa and 147 samples remaining in group 2.

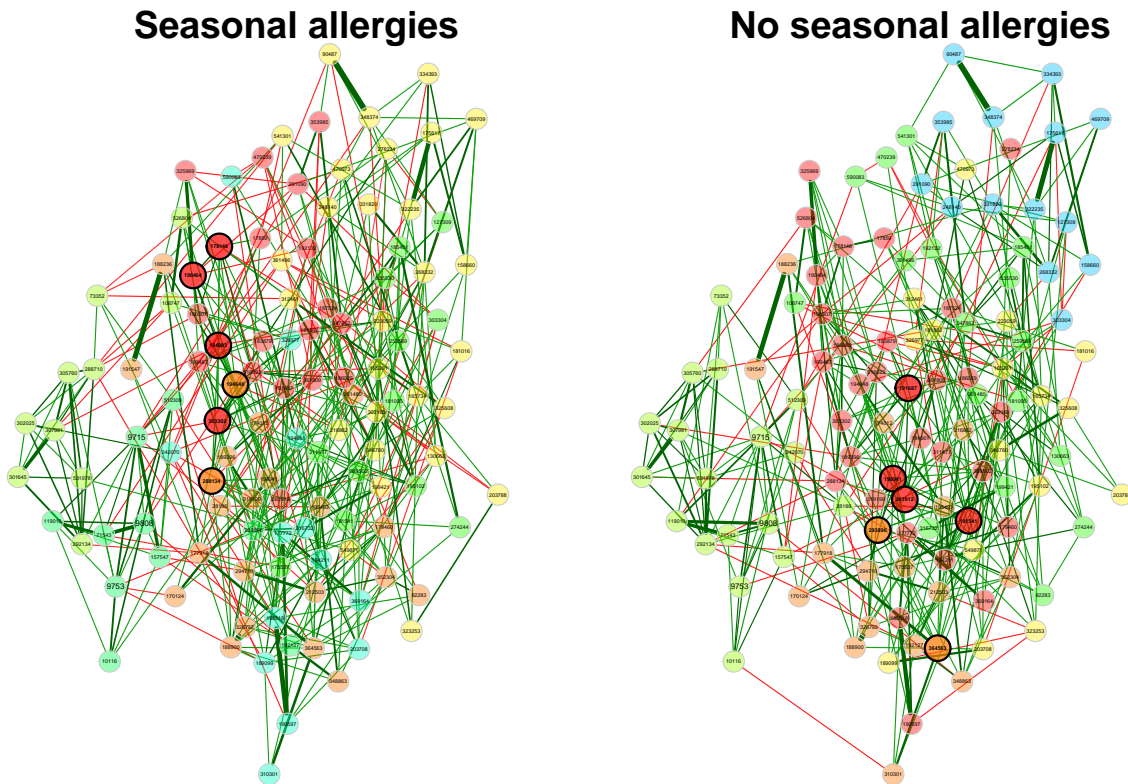
##
## Calculate 'spring' associations ...
## Registered S3 method overwritten by 'dendextend':
##   method      from
##   rev.hclust  vegan
## Registered S3 method overwritten by 'seriation':
##   method      from
##   reorder.hclust  vegan

```



## Network analysis and plot

```
# Network analysis  
# Note: Please zoom into the GCM plot or open a new window using:  
# x11(width = 10, height = 10)  
duo_props <- netAnalyze(duo_net, clustMethod = "cluster_fast_greedy",  
                        gcmHeat = FALSE)  
  
# Network plot  
plot(duo_props,  
     sameLayout = TRUE,  
     title1 = "Seasonal allergies",  
     title2 = "No seasonal allergies")
```



## Differential network

```
duo_diff <- diffnet(duo_net,  
                   diffMethod = "discordant",  
                   discordThresh = .9,  
                   adjust = "lfr")
```

```
## Checking input arguments ...
```



```
## Done.
```

```
table(duo_diff$classMat)
```

```
##  
##      1      2      3      4      5      7      9  
## 12004   550   66   378   70   150   238
```

### Inside the `diffnet()` function

```
x <- duo_net  
countMat1 <- x$countMat1  
countMat2 <- x$countMat2  
countsJoint <- x$countsJoint  
normCounts1 <- x$normCounts1  
normCounts2 <- x$normCounts2  
assoMat1 <- x$assoEst1  
assoMat2 <- x$assoEst2  
  
combMat <- rbind(countMat1, countMat2)  
  
x_expr <- Biobase::ExpressionSet(assayData = t(combMat))  
groups <- c(rep(1, nrow(normCounts1)), rep(2, nrow(normCounts2)))  
lowtri <- lower.tri(assoMat1, diag = FALSE)  
  
corrVector1 <- assoMat1[lowtri]  
corrVector2 <- assoMat2[lowtri]  
  
vector_names <- NetCoMi:::.getVecNames(t(assoMat1))  
names(corrVector1) <- vector_names  
names(corrVector2) <- vector_names  
  
discord <- discordant::discordantRun(corrVector1, corrVector2, x_expr)
```

### Inside `discordantRun()`

```
v1 <- corrVector1  
v2 <- corrVector2  
x <- x_expr  
  
v1 <- discordant:::fishersTrans(v1)  
v2 <- discordant:::fishersTrans(v2)  
  
x <- Biobase::exprs(x)  
  
pdata <- cbind(v1, v2)  
param1 <- sd(v1)  
param2 <- sd(v2)  
  
param1
```

```
## [1] 0.03374838
```

```
param2
```

```
## [1] 0.03244901
```

```
class <- cbind(discordant:::.assignClass(v1, param1, 3),  
              discordant:::.assignClass(v2, param2, 3))  
  
pd <- discordant:::em.normal.partial.concordant(pdata, class, 3)
```

```
mu_sigma <- pd$mu_sigma  
rownames(mu_sigma) <- c("mu", "sigma")  
colnames(mu_sigma) <- c(0, "-", "+")  
mu_sigma
```

#### Estimated parameters for group 1

```
##           0           -           +  
## mu  7.102831e-07 0.0157181344 0.12575031  
## sigma 1.514659e-08 0.0008289415 0.02579694
```

```
nu_tau <- pd$nu_tau  
rownames(nu_tau) <- c("nu", "tau")  
colnames(nu_tau) <- c(0, "-", "+")  
nu_tau
```

#### Estimated parameters for group 2

```
##           0           -           +  
## nu  7.440324e-07 0.017771564 0.11087457  
## tau 1.159816e-08 0.000447398 0.01971637
```

The means of the negative components are positive.

```
range(corrVector1)
```

#### Range of the correlations

```
## [1] -0.1634159  0.7334835
```

```
range(corrVector2)
```

```
## [1] -0.1183948 0.7135112
```

### Assigned classes

The matrices contain the correlations in the two groups, the initially assigned mixture components (possible values: 0 = "0", 1 = "-", 2 = "+"), and the finally assigned class (values from 1 to 9).

```
classmat <- cbind(v1, v2, class, discord$classVector)
colnames(classmat) <- c("corr1", "corr2", "comp1", "comp2", "class")
classmat <- as.data.frame(classmat)

for (i in 1:9) {
  print(head(classmat[classmat$class == i, ], 15))
  print("-----")
}
```

```
##           corr1 corr2 comp1 comp2 class
## 71543_307981    0    0    0    0    1
## 331820_307981    0    0    0    0    1
## 322235_307981    0    0    0    0    1
## 469709_307981    0    0    0    0    1
## 73352_307981    0    0    0    0    1
## 188236_307981    0    0    0    0    1
## 158660_307981    0    0    0    0    1
## 364563_307981    0    0    0    0    1
## 130663_307981    0    0    0    0    1
## 190597_307981    0    0    0    0    1
## 278234_307981    0    0    0    0    1
## 512309_307981    0    0    0    0    1
## 470239_307981    0    0    0    0    1
## 90487_307981    0    0    0    0    1
## 326792_307981    0    0    0    0    1
## [1] "-----"
##           corr1 corr2 comp1 comp2 class
## 184983_307981 -0.0420926626    0    1    0    2
## 108747_307981 -0.0260267516    0    0    0    2
## 361496_71543  -0.0024894984    0    0    0    2
## 175537_71543  -0.0429090374    0    1    0    2
## 9753_71543    0.0517102294    0    2    0    2
## 199421_331820 0.0108482818    0    0    0    2
## 347862_331820 0.0243009039    0    0    0    2
## 130663_322235 0.0282997700    0    0    0    2
## 541301_322235 0.0117609899    0    0    0    2
## 199421_322235 0.0009675611    0    0    0    2
## 28186_322235  -0.0025256899    0    0    0    2
## 158660_469709 0.0180448647    0    0    0    2
## 181016_469709 0.0149112603    0    0    0    2
## 348374_469709 0.0273543136    0    0    0    2
## 188236_73352  -0.0520540948    0    1    0    2
## [1] "-----"
```

```

##           corr1      corr2 comp1 comp2 class
## 470239_73352 -0.08443005 0.0000000000    1    0    3
## 194648_188236 0.08801727 0.0000000000    2    0    3
## 549871_130663 -0.06537320 0.0000000000    1    0    3
## 261912_130663 -0.06951029 0.0000000000    1    0    3
## 183879_278234 0.11862379 0.0000000000    2    0    3
## 216862_278234 -0.16489428 0.0000000000    1    0    3
## 541301_470239 -0.07921847 0.0000000000    1    0    3
## 190307_184983 0.08543390 0.0000000000    2    0    3
## 369502_127309 -0.09729221 0.0000000000    1    0    3
## 363302_516022 0.20393772 0.0000000000    2    0    3
## 178146_516022 0.08674166 0.0004176095    2    0    3
## 261912_516022 0.10132520 0.0000000000    2    0    3
## 186029_516022 0.08794545 0.0000000000    2    0    3
## 195102_181095 0.10405121 0.0000000000    2    0    3
## 175537_181095 0.11909876 0.0000000000    2    0    3
## [1] "-----"
##           corr1      corr2 comp1 comp2 class
## 28186_307981    0 0.010784894    0    0    4
## 9753_307981     0 0.002797957    0    0    4
## 10116_71543     0 0.023101735    0    0    4
## 322235_331820   0 0.034334330    0    2    4
## 188236_331820   0 0.006918135    0    0    4
## 353985_331820   0 0.004557863    0    0    4
## 174012_331820   0 0.020243669    0    0    4
## 165734_331820   0 0.011920579    0    0    4
## 165734_322235   0 0.046966187    0    2    4
## 127309_469709   0 0.001072556    0    0    4
## 119010_73352    0 0.015049541    0    0    4
## 9808_73352      0 0.016719116    0    0    4
## 531978_73352    0 0.004627399    0    0    4
## 184983_188236   0 0.029483667    0    0    4
## 165261_158660   0 0.039802388    0    2    4
## [1] "-----"
##           corr1      corr2 comp1 comp2 class
## 531978_307981 0.031744797 0.043284318    0    2    5
## 90487_322235 0.023547187 0.001958302    0    0    5
## 291090_469709 0.037117792 0.040874352    2    2    5
## 288710_73352 0.027364482 0.010655436    0    0    5
## 174012_188236 0.008533237 0.024039962    0    0    5
## 311477_364563 0.047211231 0.001711939    2    0    5
## 188900_364563 0.066079619 0.003227075    2    0    5
## 192132_130663 0.015187159 0.025895185    0    0    5
## 302160_278234 0.022183286 0.035939537    0    2    5
## 174012_326792 0.014823881 0.010855429    0    0    5
## 195102_311477 0.005013682 0.011835013    0    0    5
## 319199_311477 0.052168343 0.009076221    2    0    5
## 325969_516022 0.034881442 0.004995463    2    0    5
## 9808_9715      0.001084427 0.055607370    0    2    5
## 292134_9715    0.058261344 0.032696009    2    2    5
## [1] "-----"
## [1] corr1 corr2 comp1 comp2 class
## <0 rows> (or 0-length row.names)
## [1] "-----"

```

```

##          corr1      corr2 comp1 comp2 class
## 158660_331820    0 0.07094187    0    2    7
## 259569_322235    0 0.06622734    0    2    7
## 361480_130663    0 0.10070549    0    2    7
## 291090_512309    0 -0.04436540    0    1    7
## 293896_512309    0 -0.06059780    0    1    7
## 181095_470239    0 -0.11895265    0    1    7
## 334393_90487     0 0.05931386    0    2    7
## 516022_326792    0 0.07844021    0    2    7
## 157547_326792    0 -0.03833317    0    1    7
## 294710_326792    0 0.06384932    0    2    7
## 312461_311477    0 0.15213653    0    2    7
## 190307_311477    0 0.13820581    0    2    7
## 127309_248140    0 0.06737420    0    2    7
## 291090_248140    0 0.06490081    0    2    7
## 361480_541301    0 0.07039655    0    2    7
## [1] "-----"
## [1] corr1 corr2 comp1 comp2 class
## <0 rows> (or 0-length row.names)
## [1] "-----"
##          corr1      corr2 comp1 comp2 class
## 301645_307981 0.46465035 0.46047781    2    2    9
## 305760_307981 0.23839003 0.26850368    2    2    9
## 302025_307981 0.16013202 0.13393275    2    2    9
## 292134_307981 0.10987737 0.08568544    2    2    9
## 288710_307981 0.10631493 0.02073500    2    0    9
## 9715_71543     0.08640843 0.01845449    2    0    9
## 119010_71543  0.38607245 0.44589247    2    2    9
## 9808_71543     0.54777548 0.21429264    2    2    9
## 268332_331820 0.28220507 0.35020218    2    2    9
## 469709_322235 0.20547914 0.03977834    2    2    9
## 175617_322235 0.66918378 0.74656773    2    2    9
## 268332_469709 0.24677886 0.20923294    2    2    9
## 512309_73352   0.05320030 0.07867238    2    2    9
## 191547_188236 0.78945388 0.69591300    2    2    9
## 177918_188236 0.16078507 0.22977739    2    2    9
## [1] "-----"

```