**Quick tutorial**

This document is intended as a light and short alternative to the manual/vignette to get a quick overview of the functions in piano and a suggested workflow. For more details, check out the manual/vignette, function documentation and the help section on www.sysbio.se/piano.

**Installation or update**

Download the latest R version and run:

source('http://www.bioconductor.org/biocLite.R')  
biocLite('piano')

Optionally, to install *all* additional packages used by piano:  
  
biocLite('piano', dependencies=TRUE)

**Microarray analysis**

This example assumes you have the directory *C:/your/directory/with/cel-files/* with the following files:

*control\_1.cel control\_2.cel control\_3.cel   
mutantA\_1.cel mutantA\_2.cel mutantA\_3.cel   
mutantB\_1.cel mutantB\_2.cel mutantB\_3.cel   
mutantC\_1.cel mutantC\_2.cel mutantC\_3.cel   
setup.txt*

and that *setup.txt* looks like this:

file condition  
control\_1 ctrl  
control\_2 ctrl  
control\_3 ctrl  
mutantA\_1 mutA  
mutantA\_2 mutA  
mutantA\_3 mutA  
mutantB\_1 mutB  
mutantB\_2 mutB  
mutantB\_3 mutB  
mutantC\_1 mutC  
mutantC\_2 mutC  
mutantC\_3 mutC

Run the following workflow:

library(piano)   
setwd('C:/your/directory/with/cel-files/')  
d <- loadMAdata()  
d  
d$setup  
runQC(d, save=TRUE)  
extractFactors(d)  
pfc <- diffExp(d, c('mutA - ctrl', 'mutB - ctrl', 'mutC - ctrl'), save=TRUE)

This will save the resulting tables and figures in your directory (if you do not want this, omit save=TRUE).

**Gene set analysis**

This example assumes you have a file *gene\_sets.sif* giving all the gene-gene set associations (note that other alternative formats also are supported):

geneSet1 NA gene1  
geneSet1 NA gene2  
geneSet1 NA gene3  
geneSet2 NA gene1  
geneSet2 NA gene4  
geneSet3 NA gene5  
geneSet3

Further on, you also need a vector of gene level statistics and a vector of fold changes (here we will use the ones from the previous example on microarray analysis). Below is the workflow:

library(piano)   
p <- pfc$pValues['mutA - ctrl'] # or use your own statistics...  
fc <- pfc$foldChanges['mutA - ctrl'] # or use your own fold changes...  
geneSets <- loadGSC('gene\_sets.sif')  
gsaRes <- runGSA(p, fc, gsc=geneSets)  
networkPlot(gsaRes, class='non')  
GSAsummaryTable(gsaRes, save=TRUE, file='gsaResTab.xls')

This will run gene set analysis and return gene sets that are overrepresented by significant genes. The results can be visualized by the network plot and are summarized in the resulting file *gsaResTab.xls.*

**Consensus gene set analysis**

Assuming you have run multiple gene set analyses using runGSA with different methods/settings (but the same gene set collection), you can calculate a consensus result based on all these different runs:

library(piano)  
resList <- list(gsaRes1, gsaRes2, gsaRes3, gsaRes4, gsaRes5, gsaRes6, gsaRes7)  
consensusHeatmap(resList)