visualization

Lori Shepherd and Jonathan Dare

July 11, 2007

Contents

Some standard visualizations of data include:

- heatmaps of statistics
- genomic plots for all samples
- mean and frequency across all samples

We also have some user interactive tools:

- principle component viewer
-

1 heatmaps

A common graphical representation of microarray data is through heatmaps of the chip images. We have created a few different options for viewing these heatmaps, see the R help file on image.aCGH. There is a built in function that will make heatmaps for samples within an aCGH object and save them all to a postscript and pdf file. This function is makeHeatmaps.

```
> library(aCGHplus)
> load("RData/aCGH.RData")
> makeHeatmaps(aCGH, smpls = NA, type = "correction", fileName = "correctionHeatmaps",
+     pdfFlag = T)
```

This function makes heatmaps of the before correction and after correction chip images for all samples and saves them all to file in the plots directory correctionHeatmaps.ps. This postscript file is also converted to a pdf file. If the user only wanted a specific sample or set of samples, smpls may be specified with the numeric index of the sample in the aCGH object.

For example if we wanted only the first and third samples' background images:

```
> makeHeatmaps(aCGH, smpls = c(1, 3), type = "background", fileName = "BG.1.3.Heatmaps",
+     pdfFlag = T)
```

The three values that are currently accepted for type are "background","correction", and "mean". If anything else is specified, the program defaults to mean.

2 genomic plots

Another common graphical representation of data are genomic plots. There are a few functions for building genomic plots including: create.GenomeOneRow, plotGenome, create.multiGenomePlot and GenomeImage. We have provided a built in function that will plot an aCGH object's samples' genomic plots and save them to a postscript file. This postscript file can also be converted to a pdf file. This function is called makeGenome3Row

```
> makeGenome3Row(aCGH, smpls = NA, fileName = "GenomeImages", pdfFlag = F,
+     fitFlag = T)
```

This will create a postscript file named GenomicImages in the plot directory. This file will contain genomic plots of the log2.ratios and, because fitFlag is tripped to true, the smoothed log2 ratios (log2.ratios.fitted) are used for all samples in the aCGH object. A smaller sample set may be specified, just like in the makeHeatmaps call, by listing a smpls index. Any additional arguments that may be included are passed into the plotGenome function. (see R help file for plotGenome). In this case because pdfFlag is F, only the postscript file is created.

3 genomic plots across all samples

Although it is nice to have each individual samples' genomic plots to scrutinize, it may be useful to get an idea of how the genome behaves across all samples. This overview of all samples is done through a mean log ratios across all samples plot and/or a frequency plot. We have included functions to make frequency and mean plots, as well as an interactive mean across samples plot: see pacakge help files on makeFqPlot, makeMeanGraph, getMean, and meanPlot. We have included a built in function plotMean.vs.freq that will graph both the mean across samples' and the frequency plot over the entire genome as well as subsequent chromosome plots for any number of specified chromosomes. All these

graphs will be saved to a file in the plots directory. Each page has a mean and a frequency graph.

```
> make.Mean.vs.Freq.plots(aCGH, smplDX = NA, maxNA = 0.25, rmNA = T,
+     fitVls = T, ylims = c(-1, 1), cutoffs = c(-0.15, 0.15), chrmList = 1:22,
+     fileName = "mean.freq.plots", recenter = T, recenterOpt = 2,
+     orientation.flipped = 1)
```

Let's examine the above call a little further. The smplDX in this case will be the set of samples checked to make sure a certain percentage of the sample data is present. This percentage is determined by maxNA. If a sample contains above the maxNA it will not be included in the analysis.The default, NA, will used all samples. fitVls and rmNA are used in the call to getMean. This call will retrieve the mean across samples. fitVls determines if log2 ratios or smoothed log2 ratios are used. To remove all missing values from calculating the means for each spot, rmNA is set to T. The default for both fitVls and rmNA are T. ylims and cutoffs are options for the frequency graph. ylims controls the y axis label; the lower and upper bound of axis is specified. Cutoffs is the bounding range, lower and upper, of what will be called a loss and a gain for values. Anything outside this region will be marked as an aberrant region. As mentioned before, after the graph over the entire genome is plotted, smaller graphs of individual chromosomal regions may also be graphed. This is controled by chrmList. chrmList is numeric (i.e. human chromsome X would be specified as 23), and only the chromosome numbers listed will be graphed. If no chromsome graphs are desired, chrmList may be set to NA. All graphs produced, genome and chromosome, will be saved in plots directory with the file names specified by fileName. The last arguments to mention are the recentering arguments. It is important to recenter the aCGH object. This may be done at any time so it may not be necessary to recenter the data if a recenter function has previously been called. This function does allow the aCGH object to be recentered. The package currently has three methods of recentering:

1. recenter on autosomes
2. recenter on autosomes removing any missing data (NA)
3. recenter on autosomes removing any missing data and then based on only a percentage of the most aberrant spots.

The recommened, default, method of recentering is the second option. If recentering is performed by the plot call, all default arguments are used. If other arguments besides the defaults are desired, recentering should be performed before the aCGH object is used in the plotting call and then recenter=F. See the help files on recenter for more details.

4 interrogate data through user interactive tools

coming soon see web example of PCA viewer chipSpotViewer