Principle Component Viewer

Lori Shepherd

August 20, 2007

Contents

One useful tool provided by the aCGHplus package is the interactive principle component viewer. This tool is designed as a quality control tool. It allows the user to view a sample's genomic profile and chip images for possible errors. It also allows for quick viewing of a color coded 3D principle component graph; this graph color codes by different factors listed in the given inventory file. If colors seperate it is an indication of a batch effect based on whatever factor is being displayed at the time. Any sample may be flagged, removed, scored, or commented. The results or a flag is added in the inventory file of an aCGH object for these actions.

1 Loading Principle Component Viewer

This application requires an aCGH object that has undergone circulary binary segmentation. Please see aCGHplus user manual for more information on creating this object. We have provided an example with the aCGHplus package. We begin by loading the package, loading the example, and loading the provided object.

```
library(aCGHplus)
Write.aCGH.ex2()
load("RData/aCGH.RData")
```

The Principle Component Analysis (PCA) Viewer application may be run with the function runPCA:

```
runPCA(aCGH,
       smpldx = NA,
       aCGHfileName = NA,
       factorList = NA,
       invName = NA,
       overwriteInv = FALSE)

or if all these default values are used
```

`runPCA(aCGH)`

The only required arugment is an aCGH object. The other arguments may be specified if the user desires setting other than the default.

- smpldx: smpldx controls which samples are used in the analysis. If smpldx is NA, all the samples of an aCGH object are used. If smpldx is specified it should be a numeric array listing the numeric index of the desired sample within the aCGH object. If the user, for example, only desired the first two sample and the sixth sample, smpldx = c(1,2,6). The user may find the which function provided with the R base package useful for determining indecies of desired sample (i.e. which(aCGH$inventory$<column.name>==<desired.field>) such as which(aCGH$inventory$sex ==”FEMALE” which will give the indicies of which samples are of the sex female)).

- aCGHfileName: The aCGHfileName argument determines the name of file that will save the initial aCGH object with principle component data. This file will be generated in the RData/ directory with a .RData extension. If aCGHfileName is NA, the intial aCGH file name will be PCAtest. The file RData/PCAtest.RData is generated. When utilizing the application there are opportunities to save the aCGH object at various points of the analysis, the user will be prompted for other aCGH file names to save the updated object.

- factorList: factorList determines the list of factors to color code the PCA graph. The factor list is made up of column names for the aCGH inventory. Any column name may be included in the factorList. If factorList is NA, a factor that includes all samples is created, and the factorList will include all the columns in the inventory. A NA may be included anywhere within the factorList to initilize a factor which encompasses all samples. The names function of the R base package may be used to see possible factorList items. (i.e. names(aCGH$inventory)). If the user, therefore, wanted the factors to include sex, orientation, the all inclusive, and User factorList=c(”sex”, ”orientation”, NA, ”User”).

- invName: As the PCA Viewer runs, items such as samples flagged, samples, removed, sample scores, and sample comments are stored. This information and other userful objects needed by the PCA application are saved in an inventory files.This is the name of the initial PCA inventory. It will be saved as an index in the aCGH object and a corresponding file is created in a directory PCA.inventory. If NA, the initial inventory is called origPCA; A file PCA.inventory/origPCA.RData is created. Just as there are opportunities for the user to save the aCGH object while running the application, there are opportunities to save the corresponding inventory files. The user will be prompted for a name for these files.

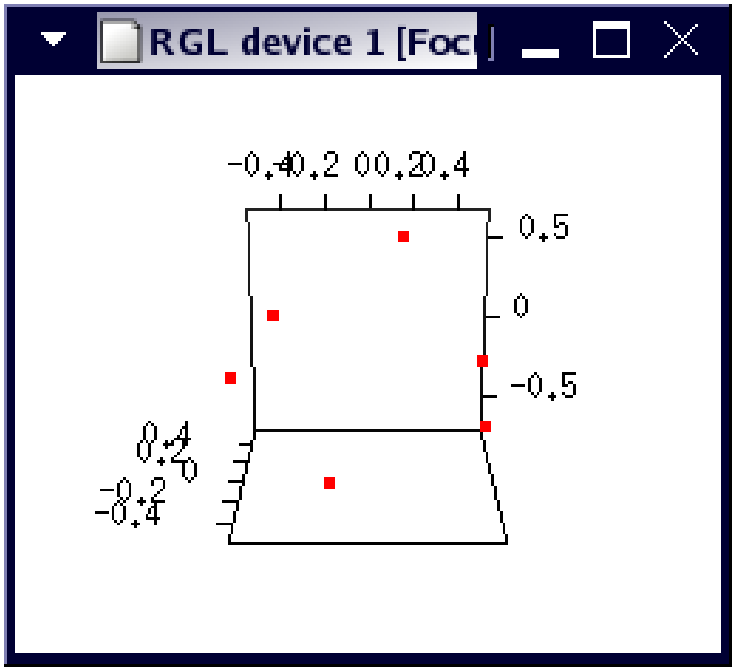
- overwriteInv: overwriteInv is a logical indicating if the last inventory entry recorded for the aCGH object should be overwritten with the initial inventory entry created.

Be patient when initially loading an aCGH object. If the aCGH object contains a large sample number it may take several minutes to calculate the principle components.

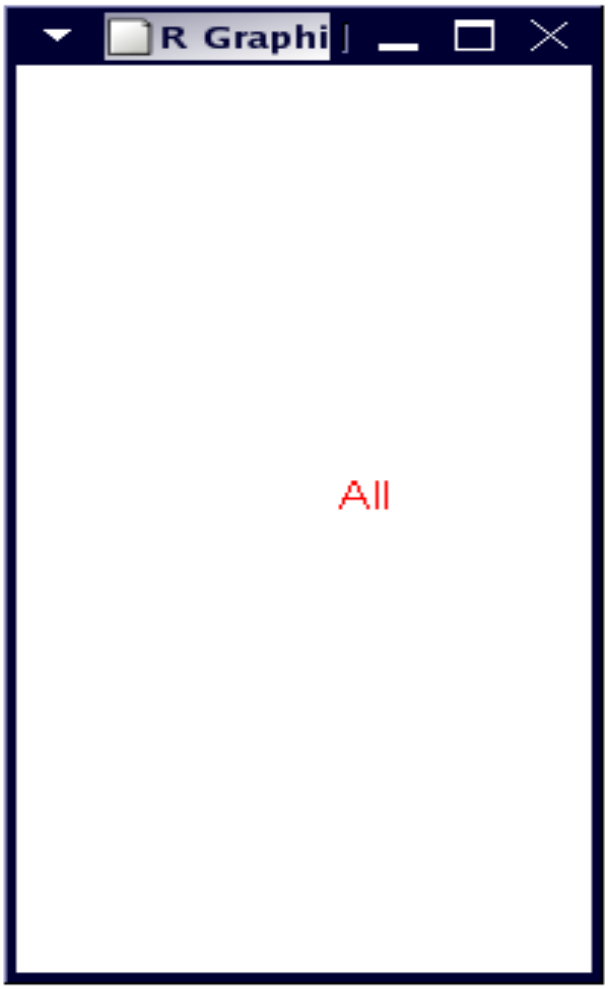
2 Brief Widow Descriptions

There are six windows that will appear on the screen each for a different purpose: 3-D PCA graph, factor legend, choice legend, PCA/sample genomic plot, and two for chip images.

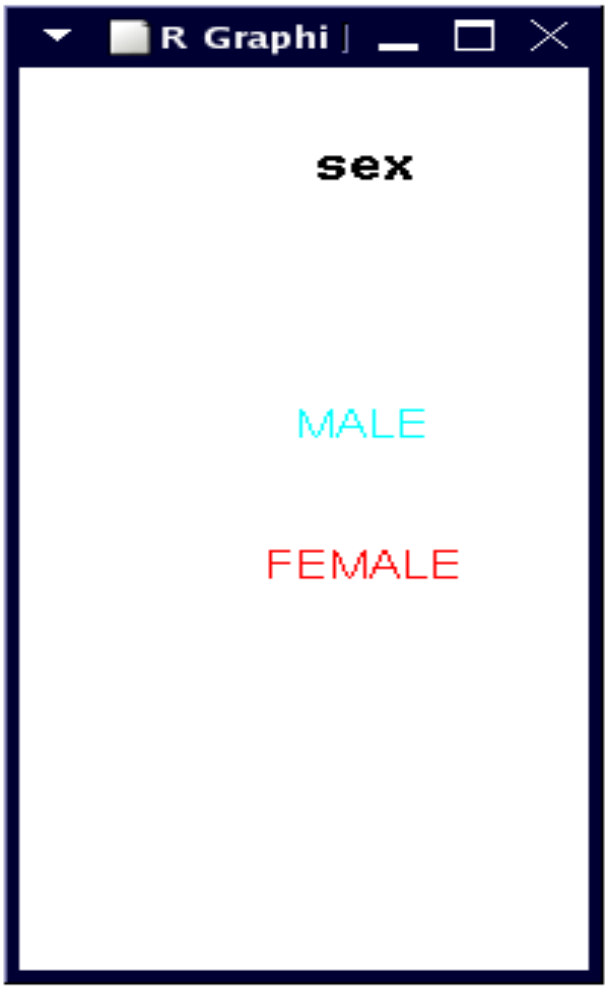
1. 3-D PCA Graph: This graph will appear upon running the runPCA function. This window is a user interactive window. If the factorList was initialized with a NA, an all inclusive color coded graph will appear such as the following:



2. factor legend: This legend gives a description of what the color codings mean on the 3-D PCA graph. They are determined by factoring the inventory. If the initial factorList is NA or begins with NA, the following graph will appear upon running the function:

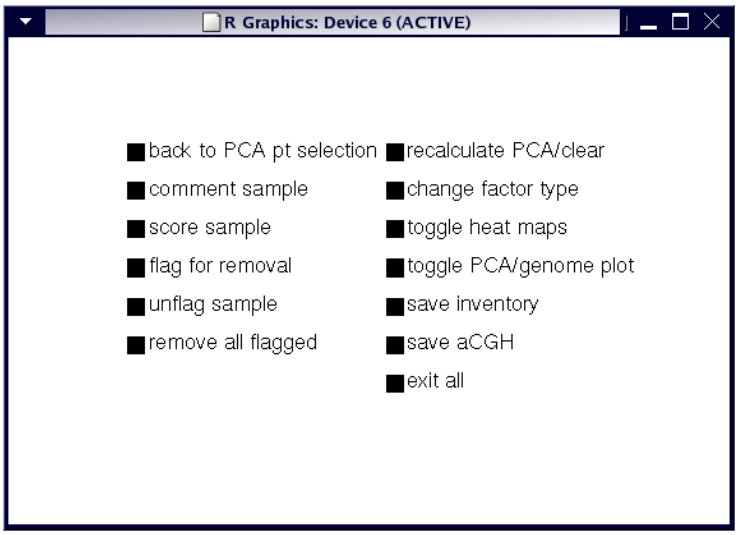


If a different factor such as sex is specified as the first item in the factorList a graph like the following would appear:



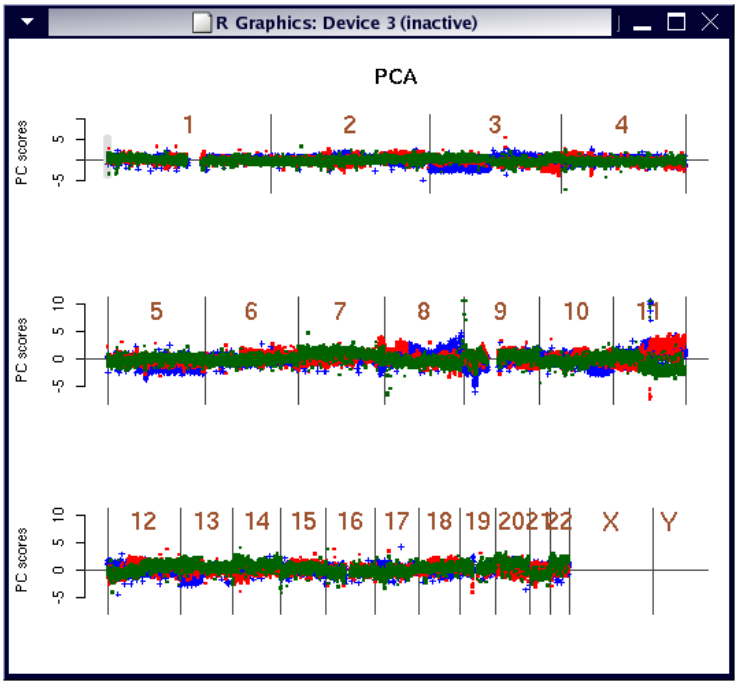
The title will be by what column factor has occured. The proceding entries will be determined by the factor levels and will be color coded accordingly.

3. choice legend: The choice legend also appears when first running the program. This window list different options that may be taken by the user to change the graph types or to take action on a sample. It looks like this:



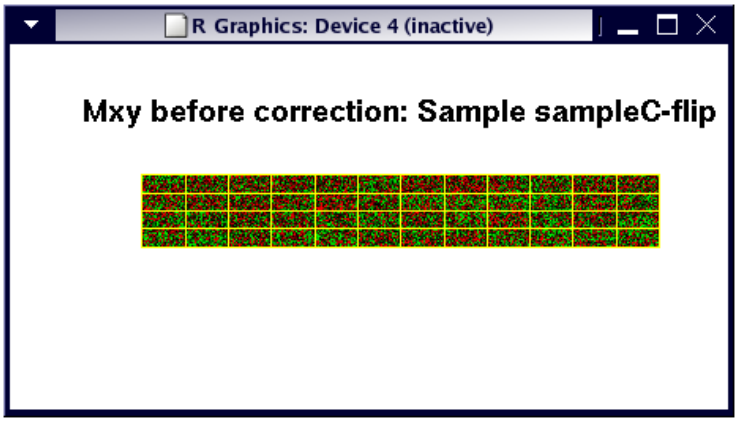
The options will be dicussed later on.

4. PCA/sample genomic plot: The initial version of this window will depict a genomic profile plot of the first three prinicple component values.



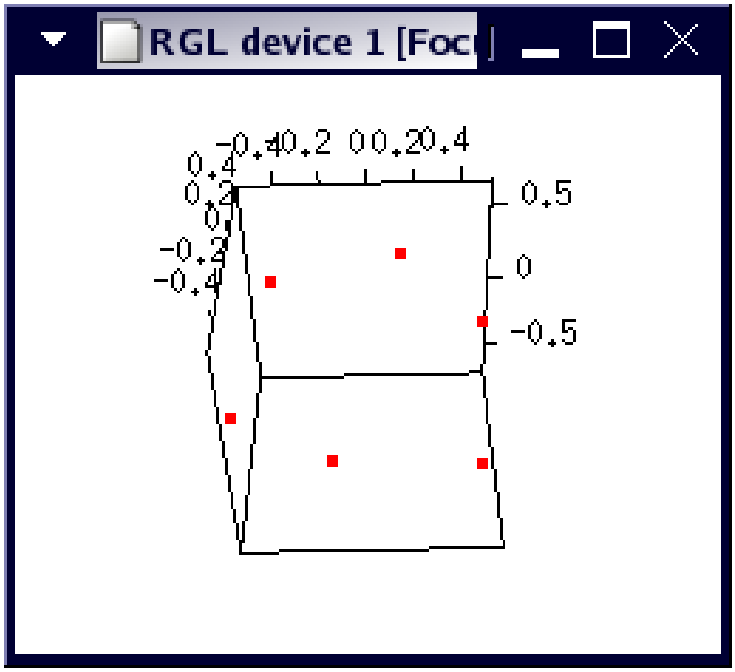
When the user selects a point on the 3-D graph this graph will change to the genomic profile of the sample's log2 ratio values. It is possible to return to the initial PCA genomic plot through the PCA choice legend. This will be dicussed further later on.

5. chip image: When the application is first run, these two window are blank. This windows will display a sample's chip image. It is possible to view three different sets of chip images: cy3/cy5, background cy3/background cy5, before correction/after correction. When the chip images are displayed it will appear similar to the following:

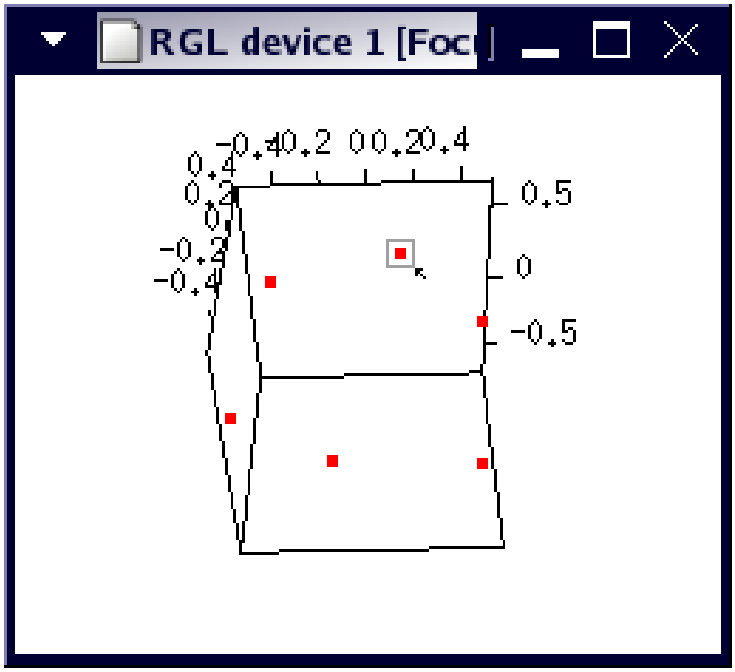


3 Instructions and Details

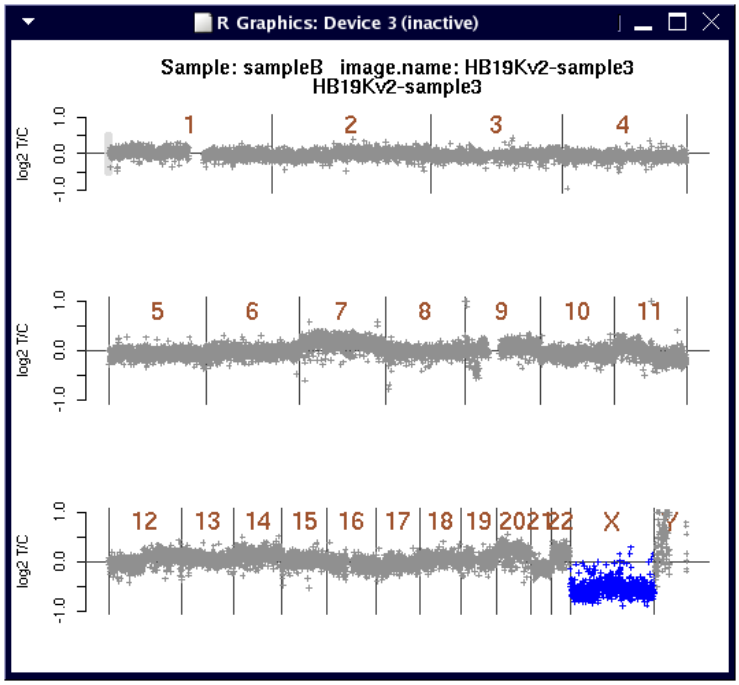
The 3-D PCA Graph window will be the first user interactive window. This graph may be rotated by click and dragging the right mouse button. The graph will rotate in the direction of the mouse. The following is an example when the graph is rotated downwards:



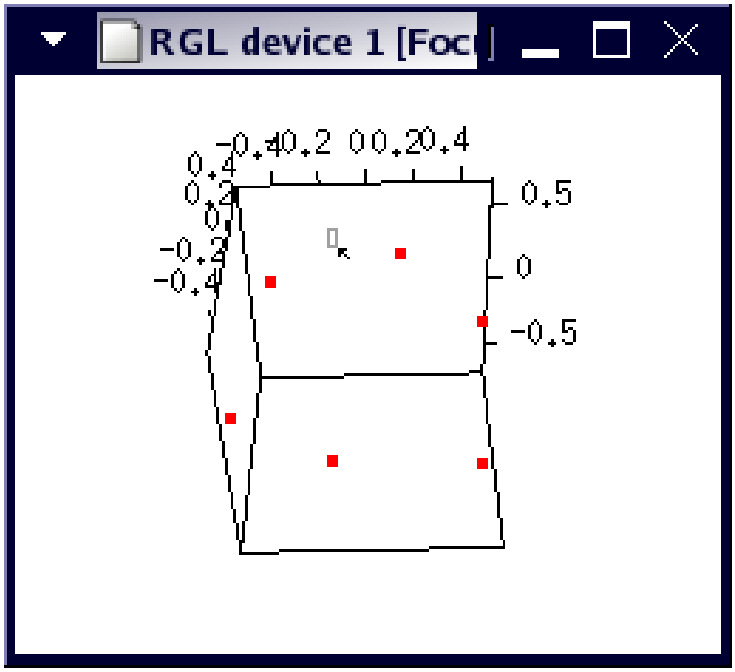
Any of the samples may be selected by clicking a dragging the right mouse button so that the box surrounds a point.



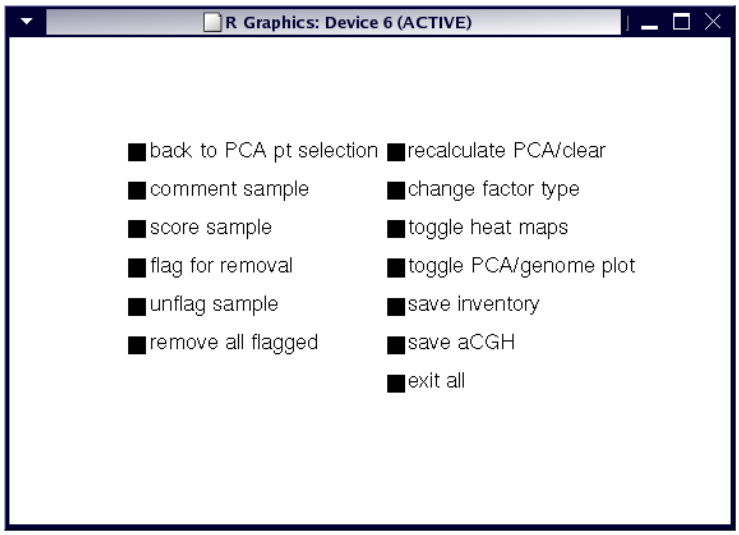
When a point is selected the other windows will be updated to display information for that specific sample. The PCA/sample genomic profile window will now have the selected sample's genomic profile.



The chip image windows will depict the chip images before and after a smoothing method was applied. The user may continue to select samples on the 3-D PCA graph to view genomic profiles and chip images. It is possible to see other chip images or to take other actions on the sample or the graphs depicted. This is achieved by selecting white space on the 3-D graph.



After white space is selected the choice legend becomes the user interactive window. There are thirteen options for the user:



1. back to PCA pt selection: Selecting this option will make the 3-D PCA graph the user interactive window again. This allows the user to return to selecting samples.

2. comment sample: The last selected point is the active sample. If the user wishes to make a comment about this sample based on the genomic profile, chip images, or PCA graph, the comment will be saved as part of the sample inventory. The comment should be entered at the command prompt. The comment will end when return is pressed.

3. score sample: The last selected point is the active sample. If the user wishes to score a sample based on the genomic profile, chip images, or PCA graph, the score will be saved as part of the sample inventory. The score should be entered at the command prompt. It should be a single numeric value. The score will be submitted when return is pressed.

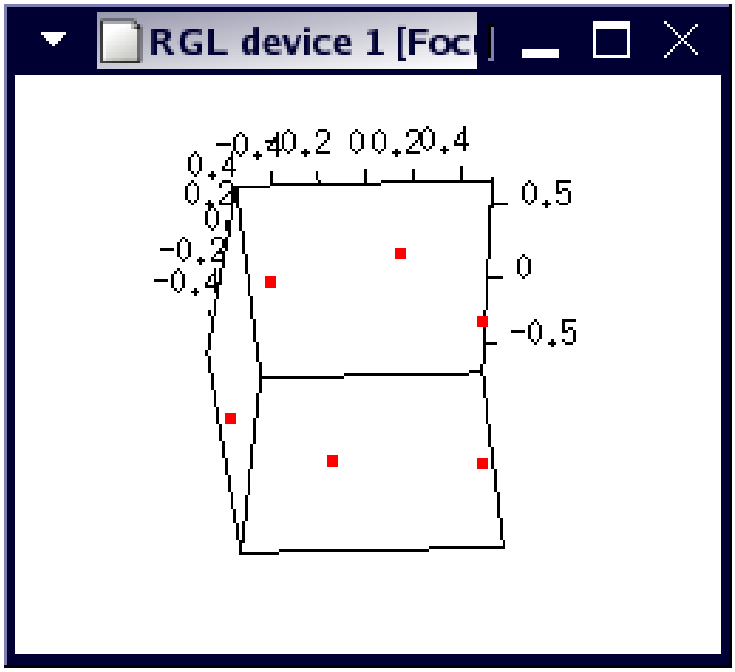
4. flag for removal: The last selected point is the active sample. If the user wishes to flag a sample for removal based on the genomic profile, chip images, or PCA graph, this sample's index will be added to a flagged list.

5. unflag sample: This allows the user to unflag a specific sample. They have the option of unflagging the current,or last selected point, sample or they may enter a sample.ID of a sample to unflag on the command prompt. If the sample's index is in the list of samples flagged it will be removed.

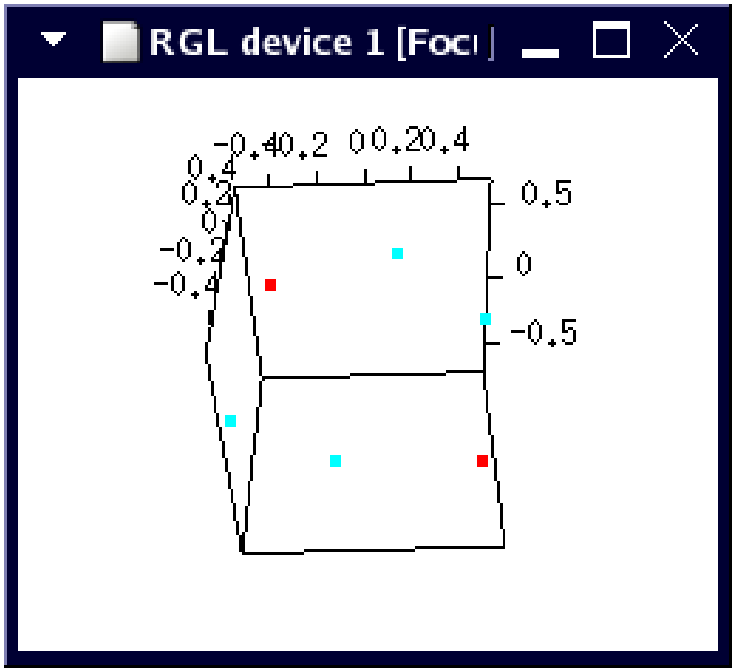
6. remove all flagged: This option will remove all the samples that are listed in the flagged samples list from the PCA graph. The 3-D PCA graph is updated with the remove of these points, however it does not recalculate the principle component values with these points removed. The principle component values will only be recalculated if the recalculate PCA option is selected.

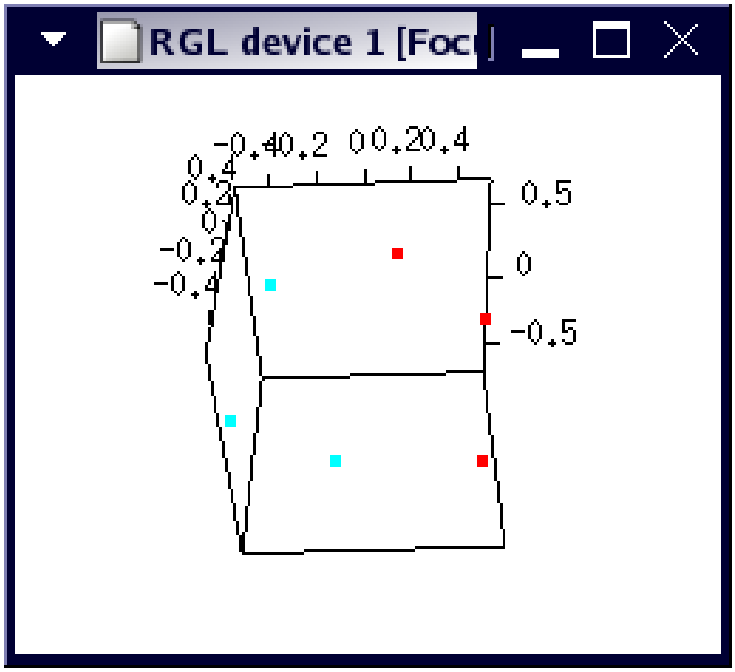
7. recalculate PCA/clear: This option will recalculate the principle component values and regraph the 3-D PCA graph. It will not include any samples that have been removed from analysis. The user will have the option to continue with the newly calculated PCA values or to return the the previous PCA values. When a sample is selected on the 3-D PCA graph its sample.ID is also places on the graph. It is possible to clear these labels by selecting this option and then to select FALSE to return to the old PCA values. If the user selects TRUE after the recalculation is made, the graph will be updated with the new values. The user will have the option to save the old values before continuing with the new values so that the old values may be returned if necessary. This is accomplished by saving the inventory file, so if the user chooses to save they will also be prompted for an inventory file name. This inventory name will be saved in a list in the aCGH object, as well as have a file written to the PCA.inventory directory.

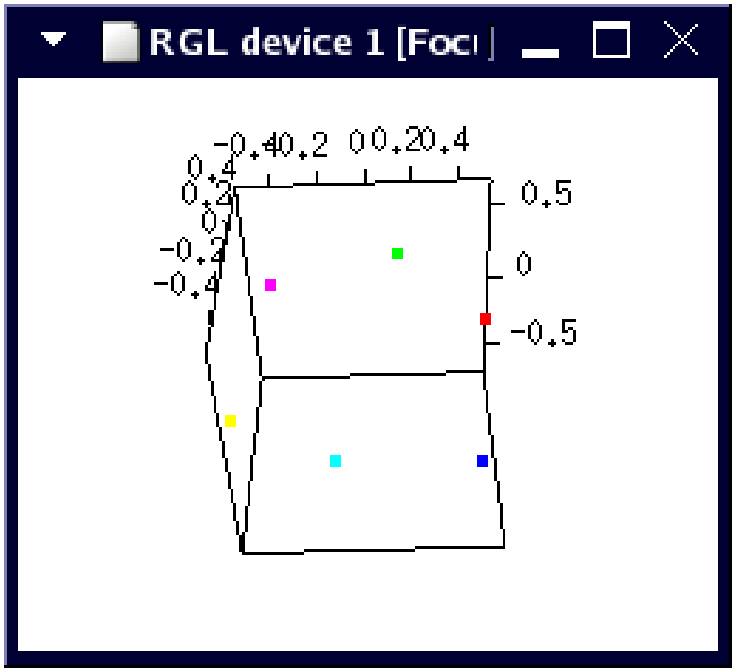
8. change factor type: As mentioned above, the user passes in a factorList of names of columns in the inventory file as an argument to the function. It is possible to see how the samples are grouped according to a factor within this factorList. This allows the user to visually see if groups are divided and a batch effect is present throughout the data. This option cycles through every factor listed in the factorList and changes the sample color on the PCA graph accordingly. The factor legend window will also be updated with the appropriate information. For example: say we passed a factorList = c(NA,"sex", "orientation","sample.ID"). The original graph that would appear would be:



If the user cycled through by clicking on this option three times the graphs that would appear would look similar to the following:


RGL device 1 [Foc
-0.4 -0.2 0 0.2 0.4
0.4
0.2
0
-0.2
-0.4
0.5
0
-0.5


RGL device 1 [Foc
-0.4
-0.2
0
0.2
0.4
0.4
0.2
0
-0.2
-0.4
0.5
0
-0.5



Again the factor legend would be updated appropriatly giving the name of the current factor being used as well as the color codings.

9. toggle heat maps: Once a sample is select the two chip image windows will show the before and after smoothing correction images. It is possible to see two other pairings of chip images: cy3/cy5 chip images and background cy3/background cy5 chip images. If the user activates the choice legend window and selects this toggle heat map option the heat maps will be updated appropriatly. The user may cycle through the chip images as desired.

10. toggle PCA/genome plot: Originally this window was a graph of the first three principle components that were used to graph the samples. When a point/sample is selected, it is updated with the sample's genomic profile. If the user selects this toggle button, the plot will toggle between the principle component values and the currently selected sample's genomic profile.

11. save inventory: The user may decide to save an inventory file at any point. The inventory file keeps track of samples that have been commented, scored, flagged, and removed, as well as the current principle component values used to make the PCA graph and the factor information. When the user chooses to save inventory, they will be prompted to

enter the name of the inventory file on the command prompt. A file will be created in the PCA.inventory directory with this name. If the user accidently enters a blank name, the current date and time will become the inventory name. This name will also be saved in a PCA.inv list within the aCGH object. After entering a file name the user will be asked if they wish to replace the last inventory. If the user selects this option, the last saved inventory in the aCGH object list will be replaced with this new inventory file. This list in the aCGH object therefore contains only paths to inventories associated with samples of that aCGH object.

12. save aCGH: Sample scores, comments, and flags are added to an aCGH object's invntory file when an aCGH object is saved. When this option is selected the user will first be asked if the previous aCGH object should be replaced. If TRUE is selected it will use the current aCGHfileName as the name of the .RData file that will be created in the RData directory. If FALSE is selected the user will be prompted to enter a file name on the command prompt.This name will be used as the file name. If a blank name is entered, the current date and time is used as the new aCGH file name.

13. exit all: When the user is finished with the current session they may exit through this option. First the user will be asked if they really wish to exit. If FALSE, the choice legend window is activated again. If TRUE, the user is then asked if they wish to save the inventory before exiting. If the inventory is not saved any information added after the initial set up or the last user opted save inventory will be lost. If TRUE for save inventory is selected the user will be asked to enter a name for the inventory file. Again if this is entered blank the date and time will be used as the file name. The user will also be asked if the new inventory file should replace the previous inventory. If true the last saved inventory will be replaced with the current inventory. After the user is asked about the inventory file, they are asked if they wish to save the aCGH object. Again if FALSE any information added to the aCGH object after the initial set up or the last save aCGH will be lost. If TRUE, the user will be asked if the new aCGH file should replace the previous file. If TRUE, the current aCGH fileName will be used as the file name. If FALSE, the user will be prompt to enter a file name at the command prompt. If a blank name is enters, the current date and time are used as the file name. The application will then exit and close all open windows.

All options when completed, except for select the back to pt selection and exit all, will return to the activated choice legend window. Remember the choice legend window remains active until the user selects the back to PCA pt selection or the exit all options. Also, the user will not be able to select a new point for viewing if they do not first select back to pt selection.

4 ReLoading PCA

After this application has been run once, it is possible to re-load PCA objects of a previous session. The user may re-load at any saved inventory point. This is achieved through the function loadPCAInv. In this function the user specifies an aCGHfileName; this is the name of the file in the RData directory containing an aCGH object with principle component data in the inventory and a list of possible inventory files in a PCAinv list. This file will also contain some of the prinicple component objects created throughout the application such as PCtest and sampleInfo. The user may also specify a given inventory start point. In is possible to save the inventory at any point while using the applications, these represent places that may be re-loaded. This will re-load information on flagged, removed, scored, and commented samples. It will also use which ever activated principle component values were used at the time of the save (in casse PCA values had been recalculated at any point). invFileName should be the name of the inventory file desired within the PCA.inventory directory. If this is left as NA, the most recently saved inventory for that aCGH object is loaded. The PCA.inventory file contains the following objects: factorInfo, PCtest, plotInfo, and sampleInfo.

5 Objects Created

- aCGH object: The aCGH object is updated to include an object PCAinv. This is a character vector containing the inventory path names associated with this aCGH object. Any file listed may be re-loaded with this aCGH object. The aCGH object's inventory also has been updated. The inventory data.frame will now include five additional columns: PCA.used, PCA.score, PCA.comment, PCA.excluded, and PCA.removed. PCA.used is a logicial indicating if the sample was included in the sample set for the original principle component analysis. PCA.score, PCA.comment, PCA.excluded, and PCA.removed will indicate values given while running the PCA application. PCA.scores give the score for any sample and PCA.comment the comment for any sample. PCA.excluded marks the samples that were flagged for removal and PCA.removed mark samples that were removed.

- PCtest: PCtest is the object created with the stats package's princomp function. This contains the principle component data. It is of class type princomp. see ?princomp in R for more details.

- sampleInfo: sampleInfo is a list. It contains the objects smpldx, scores, comments, exclude, and removed. sampleInfo is what keeps track of which samples have undergone any of these actions through the PCA application. smpldx is an integer array of the index of samples in the aCGH object used in the original PCA call. Scores is a numeric array. Any score that was given for a sample is stored; NAs are used if no score has been made.

Comments is a character array. Any comment that was given to a sample is stored; NAs are used if no comment has been made. Exclude is an integer array containing the index of any sample that was flagged for removal. Remove is an integer array containing the index of any sample that was removed from analysis.

1. sampleInfo: keeps track of samples used in analysis, sample scores, comments, flags, and removal.
   - smpldx: integer array of samples used; index within the aCGH object
   - scores: numeric array of length smpldx; keeps track of any scores given to samples
   - comments: character array of length smpldx; keeps track of any comments given to samples
   - exclude: interger array; keeps track of the index of any samples that were flagged for removal
   - removed: integer array; keeps track of the index of any samples that were removed from analysis

- factorInfo: factorInfo is a list. It contains information on factor types for the PCA application. These factor types will be used to group and color code the 3-D graph. factorInfo contains the following objects: facIdx, Ftype, iFtype, ntype, factorList, listType, and listInd. facIdx is a numeric indicating the index of the next factor to use within factorList. Ftype is of type factor. It gives the factor information for the current inventory column in factorList[listInd]. The only exception is if factorList[listInd] is NA. This is a automatically generated factor item that includes all samples. iFtype is an interger representation of Ftype; it is a numerical representation of the factor types. ntype is a numeric value indicating the total number of different factor levels in Ftype. factorList is a character array containing all user desired factor levels. factorList determines what columns in the inventory will be able to be viewed as a factor type and color code the 3-D PCA graph. This list should only contain items that are included in names(aCGH$inventory) or NA, or and index of such a column. listType may either be character or numeric. The user may specify inventory columns to use by listing the given column names directly, or by giving a numeric representation of the column of data to pull. (i.e if the names(aCGH$inventory) are [1] "sample.ID" [2] "image.name" [3] "sex" [4] "orientation" [5] "cancer type", and the user wanted to factor by image.name,orientation, and an all inclusive factor type, the factorList=c(2,4,NA) and listType would equal integer. If factorList=c("image.name", "orienation", NA) then listType would be character.) Finally, listInd indicates the current item in factorList being used to factor, group, and color code.

  1. factorInfo: keeps track of factor infomation usedd in analysis

- facIdx: numeric: index of next factor to use in factorList to group and factor samples
- Ftype: factor: current factor infomation
- iFtype: integer array: numeric represetation of Ftype
- ntype: integer: total number of different factor levels of current factor type
- factorList: either character or numeric array: list of all factors that may be used to group, factor, and color code samples
- listType: either character or numeric: indicates what class type is factorList. This aids in reading and pulling information from aCGH inventory
- listInd: numeric: index in factorList of current factor being used

- plotInfo: plotInfo contains information on the plotting windows and the colors to use for factoring.

  1. plotInfo: keeps track of graphing devices
     - clrs: character array: list of colors to use for factoring purposes. These will be the colors of points on the 3-D PCA graph
     - deviceInfo: list of device info. name and numeric indication of devices
       (a) legend.dev: integer: numeric indication of which window is equalivalent to the factor legend. This numeric representation should be able to be passed into dev.set to make window active.
       (b) genome.dev: integer: numeric indication of which window is equalivalent to the PCA/sample genomic profile. This numeric representation should be able to be passed into dev.set to make window active.
       (c) heatmap1.dev: integer: numeric indication of which window is equalivalent to the first chip image. This numeric representation should be able to be passed into dev.set to make window active.
       (d) heatmap2.dev: integer:numeric indication of which window is equalivalent to the second chip image. This numeric representation should be able to be passed into dev.set to make window active.
       (e) choice.dev: integer: numeric indication of which window is equalivalent to the choice legend containing user options. This numeric representation should be able to be passed into dev.set to make window active.