Intrudoction to caOmicsV

Hongen Zhang, Ph.D.
Genetics Branch, Center for Cancer Research,
National Cancer Institute, NIH

March 10, 2015

Contents

1 Introduction **1**

2 An Quick Demo **2**

3 Making Plot Data Set **4**

4 Plot bioMatrix Layout Manually **5**

5 Plot bioNetCircos Layout Manually **8**

6 sessionInfo **11**

1 Introduction

Translational genomics research in cancer, e.g., International Cancer Genome Consortium (ICGC) and The Cancer Genome Atlas (TCGA), has generated large multidimensional datasets from high-throughput technologies. Multidimensional data offers great promise to improve clinical applications of genomic information in diagnosis, prognosis and therapeutics of cancers. Tools to effectively visualize integrated multidimensional data are important for understanding and describing the relationship between genomic variations and cancers. The caOmicsV package provides methods to visualize multidimensional cancer genomic data in two layouts: a heatmap-like matrix layout, bioMatrix, and circular layouts superimposed on a biological network or graph, bioNetCircos.

The data that could be plotted with each layout is listed below:

- Clinical (phenotypes) data such as gender, tissue type, and diagnosis, plotted as colored rectangles
- Expression data such as RNASeq and miRNASeq, plotted as heatmap

- Category data such as DNA methylation status, plotted as colored box outlines on bioMatrix layout and bars on bioNetCircos layout
- Binary data such as mutation status and DNA copy number variations, plotted as colored points
- Text labelling, for gene names, sample names, summary in text format

In addition, link lines can also be plotted on bioNetCircos layout to show the relationship between two samples.

For bioNetCircos layout, igraph and bc3net packages must be installed first.

2 An Quick Demo

Following code will generate a bioMatrix layout image with the build-in demo data.

```
> library(caOmicsV)
> data(biomatrixPlotDemoData)
> plotBioMatrix(biomatrixPlotDemoData, summaryType="text")
> bioMatrixLegend(heatmapNames=c("RNASeq", "miRNASeq"),
+     categoryNames=c("Methyl H", "Methyl L"),
+     binaryNames=c("CN LOSS", "CN Gain"),
+     heatmapMin=-3, heatmapMax=3, colorType="BlueWhiteRed")
```

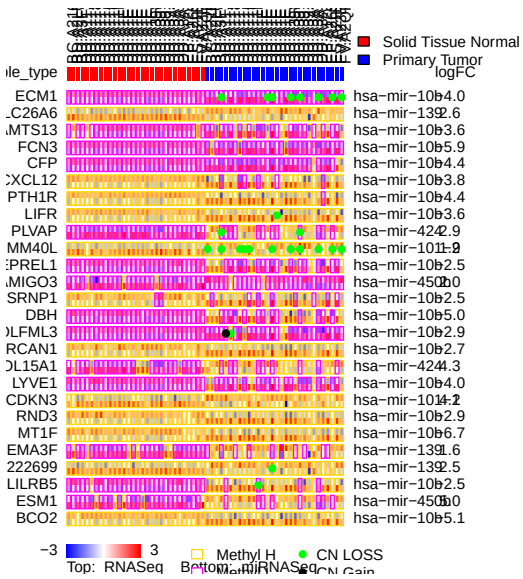

Figure 1. caOmicsV bioMatrix layout plot

Run the code below will get a bioNetCircos layout image with the build-in demo data.

```
> library(caOmicsV)
> data(bionetPlotDemoData)
> plotBioNetCircos(bionetPlotDemoData)

[1] "plot polygon"
[1] "plot heatmap"
[1] "plot heatmap"
[1] "plot bar"
[1] "plot points"

> dataNames <- c("Tissue Type", "RNASeq", "miRNASeq", "Methylation", "CNV")
> bioNetLegend(dataNames, heatmapMin=-3, heatmapMax=3)
```

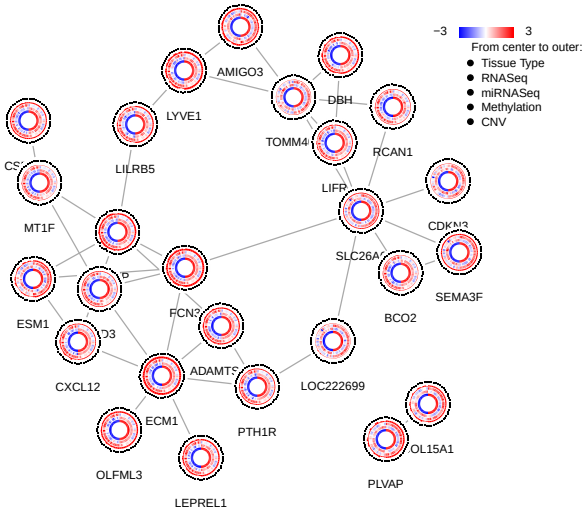

Figure 2. caOmicsV bioNetCircos layout plot

3 Making Plot Data Set

To use the default plot function shown as above, the first step is making an plot data set to hold all datasets in a list object. Two demo data sets are included in the package installation and could be explored with:

```
> library(caOmicsV)
> data(biomatrixPlotDemoData)
> names(biomatrixPlotDemoData)
```

```
[1] "sampleNames"     "geneNames"       "secondGeneNames" "sampleInfo"
[5] "heatmapData"     "categoryData"    "binaryData"      "summaryInfo"
```

```
> data(bionetPlotDemoData)
> names(bionetPlotDemoData)
```

```
[1] "sampleNames"     "geneNames"       "secondGeneNames" "sampleInfo"
[5] "heatmapData"     "categoryData"    "binaryData"      "summaryInfo"
```

caOmicsV package has a function getPlotDataSet() to make the plot data set as above. The input data to pass to the function are as below:

- sampleNames: required, character vector with names of samples to plot
- geneNames: required, character vector with names of genes to plot
- sampleData: required, data frame with rows for samples and columns for features. The first column must be sample names same as the sampleNames above in same order
- heatmapData: list of data frames (maximum 2), continue numeric data such as gene expression data.
- categoryData: list of data frames(maximum 2), categorical data such as methylation High, Low, NO.
- binaryData: list of data frames(maximum 3), binary data such as 0/1.
- summaryData: list of data frames (maximum 2), summarization for genes or for samples.
- secondGeneNames: names of second set of genes, e.g. miRNA names, to label genes on right side of matrix layout.

All genomic data must be held with data frame in the format of rows for genes and columns for samples. The first column of each data frame must be gene names same as the geneNames as above in same order. The column names of each data frame must be sample names same as sampleNames as above in same order. caOmicsV package contains functions to sort data frame for given order and several supplement functions are also included in the package to help extract required data set from big datasets by supplying required gene names and sample names in a given order.

To make caOmicsV plot, the plot data set must contains at least one genomic data (one of heatmap data, category data, or binary data)

4 Plot bioMatrix Layout Manually

The default plot method, plotBiomatrix(), is a convenient and efficient way to make bioMatirx plot. In case of necessary, users can follow procedures below to generate bioMatrix layout plot manually.

1. Demo data

```
> library(caOmicsV)
> data(biomatrixPlotDemoData)
> dataSet <- biomatrixPlotDemoData
> names(dataSet)
```

```
[1] "sampleNames"     "geneNames"       "secondGeneNames" "sampleInfo"
[5] "heatmapData"     "categoryData"    "binaryData"      "summaryInfo"
```

2. Initialize bioMatrix Layout

```
> numOfGenes <- length(dataSet$geneNames);
> numOfSamples <- length(dataSet$sampleNames);
> numOfPhenotypes <- nrow(dataSet$sampleInfo)-1;
> numOfHeatmap <- length(dataSet$heatmapData);
> numOfSummary <- length(dataSet$summaryData);
> phenotypes   <- rownames(dataSet$sampleInfo)[-1];
> sampleHeight <- 0.4;
> sampleWidth <- 0.1;
> samplePadding <- 0.025;
> geneNameWidth <- 1;
> sampleNameHeight <- 2.5;
> remarkWidth <- 2;
> summaryWidth <- 1;
> rowPadding <- 0.1;
> initializeBioMatrixPlot(numOfGenes, numOfSamples, numOfPhenotypes,
+     sampleHeight, sampleWidth, samplePadding, rowPadding,
+     geneNameWidth, remarkWidth, summaryWidth, sampleNameHeight)
> caOmicsVColors <- getCaOmicsVColors()
> png("caOmicsVbioMatrixLayoutDemo.png", height=8, width=12,
+         unit="in", res=300, type="cairo")
> par(cex=0.75)
> showBioMatrixPlotLayout(dataSet$geneNames,dataSet$sampleNames, phenotypes)
```

3. Plot tissue types on phenotype area

```
> head(dataSet$sampleInfo)[,1:3]
```

```
            TCGA.BC.A216.Normal   TCGA.BD.A2L6.Normal   TCGA.BD.A3EP.Normal
sampleID    "TCGA.BC.A216.Normal" "TCGA.BD.A2L6.Normal" "TCGA.BD.A3EP.Normal"
sample_type "Solid Tissue Normal" "Solid Tissue Normal" "Solid Tissue Normal"
```

```
> rowIndex <- 2;
> sampleGroup <- as.character(dataSet$sampleInfo[rowIndex,])
> sampleTypes <- unique(sampleGroup)
> sampleColors <- rep("blue", length(sampleGroup));
> sampleColors[grep("Tumor", sampleGroup)] <- "red"
> rowNumber <- 1
> areaName <- "phenotype"
> plotBioMatrixSampleData(rowNumber, areaName, sampleColors);
> geneLabelX <- getBioMatrixGeneLabelWidth()
> maxAreaX <- getBioMatrixDataAreaWidth()
> legendH <- getBioMatrixLegendHeight()
> plotAreaH <- getBioMatrixPlotAreaHeigth()
> sampleH<- getBioMatrixSampleHeight()
> sampleLegendX <- geneLabelX + maxAreaX
```

```
> sampleLegendY <- plotAreaH + legendH - length(sampleTypes)*sampleH
> colors <- c("blue", "red")
> legend(sampleLegendX, sampleLegendY, legend=sampleTypes,
+     fill=colors,  bty="n", xjust=0)
```

4. Heatmap plot

```
> heatmapData <- as.matrix(dataSet$heatmapData[[1]][,]);
> plotBioMatrixHeatmap(heatmapData, maxValue=3, minValue=-3)
> heatmapData <- as.matrix(dataSet$heatmapData[[2]][,])
> plotBioMatrixHeatmap(heatmapData, topAdjust=sampleH/2,
+     maxValue=3, minValue=-3);
> secondNames <- as.character(dataSet$secondGeneNames)
> textColors <- rep(caOmicsVColors[3], length(secondNames));
> plotBioMatrixRowNames(secondNames, "omicsData", textColors,
+     side="right", skipPlotColumns=0);
```

5. Draw outline for each samples to show methylation status.

```
> categoryData <- dataSet$categoryData[[1]]
> totalCategory <- length(unique(as.numeric(dataSet$categoryData[[1]])))
> plotColors <- rev(getCaOmicsVColors())
> plotBioMatrixCategoryData(categoryData, areaName="omicsData",
+     sampleColors=plotColors[1:totalCategory])
```

6. Binary data plot

```
> binaryData <- dataSet$binaryData[[1]];
> plotBioMatrixBinaryData(binaryData, sampleColor=caOmicsVColors[4]);
> binaryData <- dataSet$binaryData[[2]];
> plotBioMatrixBinaryData(binaryData, sampleColor=caOmicsVColors[3])
```

7. Plot summary data on right side of plot area

```
> summaryData  <- dataSet$summaryInfo[[1]][, 2];
> summaryTitle <- colnames(dataSet$summaryInfo[[1]])[2];
> remarkWidth <- getBioMatrixRemarkWidth();
> sampleWidth <- getBioMatrixSampleWidth();
> col2skip <- remarkWidth/2/sampleWidth + 2;
> plotBioMatrixRowNames(summaryTitle, areaName="phenotype",
+     colors="black", side="right", skipPlotColumns=col2skip);
> plotBioMatrixRowNames(summaryData, "omicsData",
+     colors=caOmicsVColors[3], side="right",
+     skipPlotColumns=col2skip)
```

8. Add legend

```
> bioMatrixLegend(heatmapNames=c("RNASeq", "miRNASeq"),
+     categoryNames=c("Methyl H", "Methyl L"),
+     binaryNames=c("CN LOSS", "CN Gain"),
+     heatmapMin=-3, heatmapMax=3,
+     colorType="BlueWhiteRed")
> dev.off()

null device
          1
```

Run code above should generate an image same as Figure 1.

5 Plot bioNetCircos Layout Manually

With default bioNetCircos layout plot method, the node layout and labelling rely on the igraph package and sometimes the node layout and labelling may not be in desired location. In that case, it is recommended to manually check out the layout first then plot each item.

Following are basic procedures to make a bioNetCircos plot:

1. Demo data

```
> library(caOmicsV)
> data(bionetPlotDemoData)
> dataSet <- bionetPlotDemoData
> sampleNames  <- dataSet$sampleNames
> geneNames    <- dataSet$geneNames
> numOfSamples <- length(sampleNames)
> numOfSampleInfo <- nrow(dataSet$sampleInfo) - 1
> numOfSummary <- ifelse(dataSet$summaryByRow, 0, col(dataSet$summaryInfo)-1)
> numOfHeatmap <- length(dataSet$heatmapData)
> numOfCategory <- length(dataSet$categoryData)
> numOfBinary <- length(dataSet$binaryData)
> expr <- dataSet$heatmapData[[1]]
> bioNet <- bc3net(expr)
```

2. Initialize bioNetCircos layout

```
> widthOfSample    <- 100
> widthBetweenNode <- 3
> lengthOfRadius   <- 10
> dataNum <- sum(numOfSampleInfo, numOfSummary, numOfHeatmap,
+     numOfCategory, numOfBinary)
> trackheight <- 1.5
> widthOfPlotArea  <- dataNum*2*trackheight
```

```
> initializeBioNetCircos(bioNet, numOfSamples, widthOfSample,
+     lengthOfRadius, widthBetweenNode, widthOfPlotArea)
> caOmicsVColors <- getCaOmicsVColors()
> supportedType <- getCaOmicsVPlotTypes()
> par(cex=0.75)
> showBioNetNodesLayout()
```

3. Manually label each node

At this point, each node has its index labelled. Manually check out the desired location for node name (gene) labelling then label each node (the node index here may be different from your graph).

```
> par(cex=0.6)
> onTop <- c(14, 15, 16, 9, 7, 20, 8, 24, 10, 25)
> labelBioNetNodeNames(nodeList=onTop,labelColor="blue",
+     labelLocation="top", labelOffset = 0.7)
> onBottom <- c(26, 22, 23, 18, 19, 3, 5)
> labelBioNetNodeNames(nodeList=onBottom,labelColor="black",
+     labelLocation="bottom", labelOffset = 0.7)
> onLeft <- c(2, 11, 21, 17)
> labelBioNetNodeNames(nodeList=onLeft,labelColor="red",
+     labelLocation="left", labelOffset = 0.7)
> onRight <- c(13, 12, 4, 1, 6)
> labelBioNetNodeNames(nodeList=onRight,labelColor="brown",
+     labelLocation="right", labelOffset = 0.7)
```

Once all node names are labelled correctly, plot area of each node could be erased for plotting.

```
> eraseBioNetNode()
```

4. Plot each data set

```
> inner <- lengthOfRadius/2
> outer <- inner +  trackheight
```

Plot tissue type for each node. Repeat this step if there are more than one clinical features.

```
> groupInfo <- as.character(dataSet$sampleInfo[2, ])
> sampleColors <- rep("blue", numOfSamples);
> sampleColors[grep("Tumor", groupInfo)] <- "red"
> plotType=supportedType[1]
> groupInfo <- matrix(groupInfo, nrow=1)
> bioNetCircosPlot(dataValues=groupInfo,
+     plotType, outer, inner, sampleColors)
> inner <- outer + 0.5
> outer <- inner +  trackheight
```

Heatmap plot for each node. Repeat this step if there are more heatmap data.

```
> exprData <- dataSet$heatmapData[[1]]
> plotType <- supportedType[4]
> bioNetCircosPlot(exprData, plotType, outer, inner,
+     plotColors="BlueWhiteRed", maxValue=3, minValue=-3)
> inner <- outer + 0.5
> outer <- inner +  trackheight
```

Category data plot for each node. Repeat this step if there are more category datasets.

```
> categoryData <- dataSet$categoryData[[1]]
> plotType <- supportedType[2];
> bioNetCircosPlot(categoryData, plotType,
+     outer, inner, plotColors="red")
```

```
length of data values and colors are different.
First color will be used for all samples.
```

```
> inner <- outer + 0.5
> outer <- inner +  trackheight
```

Binary data plot for each node. Repeat this step if there are more binary datasets

```
> binaryData <- dataSet$binaryData[[1]]
> plotType <- supportedType[3]
> plotColors <- rep(caOmicsVColors[1], ncol(binaryData))
> bioNetCircosPlot(binaryData, plotType,
+     outer, inner, plotColors)
> inner <- outer + 0.5
> outer <- inner +  trackheight
```

Link samples on a node. Repeat this step for each node when needed.

```
> outer <- 2.5
> bioNetGraph <- getBioNetGraph()
> nodeIndex <- which(V(bioNetGraph)$name=="PLVAP")
> fromSample <- 10
> toSample <- 50
> plotColors <- "red"
> linkBioNetSamples(nodeIndex, fromSample,
+     toSample, outer, plotColors)
> fromSample <- 40
> toSample <- 20
> plotColors <- "blue"
> linkBioNetSamples(nodeIndex, fromSample,
+     toSample, outer, plotColors)
```

5. Add legend

```
> dataNames <- c("Tissue Type", "RNASeq", "Methylation", "CNV")
> bioNetLegend(dataNames, heatmapMin=-3, heatmapMax=3)
```

The output from the code above should be same as Figure 2.

6 sessionInfo

```
> sessionInfo()

R version 3.6.0 (2019-04-26)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows Server 2012 R2 x64 (build 9600)

Matrix products: default

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

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

other attached packages:
[1] caOmicsV_1.14.0 bc3net_1.0.4    lattice_0.20-38 Matrix_1.2-17
[5] infotheo_1.2.0  c3net_1.1.1     igraph_1.2.4.1

loaded via a namespace (and not attached):
[1] compiler_3.6.0  magrittr_1.5    tools_3.6.0     grid_3.6.0
[5] pkgconfig_2.0.2
```