Visualizing Data

BINF733 SPRING2006
Dr. Jeff Solka and Dr. Jennifer Weller
Overview

- Visualization is an essential part of exploratory data analysis.

- Visualization can be used to formulate models for the data, to confirm models for the data, to visually assess cluster structure, and to identify outliers.
Some of the R Visualization Packages

- arrayMagic
- arrayQuality
- aCGH
- glcus
- scatterplot3d
- vcd
References - I


References - II


- Brewer, Cynthia A., 1994, Color Use Guidelines for Mapping and Visualization, Chapter 7 (pp. 123-147) in *Visualization in Modern Cartography*, edited by A.M. MacEachren and D.R.F. Taylor, Elsevier Science, Tarrytown, NY.
Color Space

- The work of Cynthia Brewer (1994a, b) discussed the fact that distances in color space should reflect quantitative distances between data.

- RColorBrewer
Interactive Graphics in R

- tcltk
- RGtk
  - iSPlot
- GGobi
  - Rggobi
High-volume Scatterplots

High Volume Scatterplots in R

> library("affydata")
Loading required package: affy
Loading required package: Biobase
Loading required package: tools

Welcome to Bioconductor

Vignettes contain introductory material.
To view, simply type 'openVignette()' or start with 'help(Biobase)'.
For details on reading vignettes, see the openVignette help page.

> data("Dilution")
> x <- log2(exprs(Dilution)[, 1:2])
> x <- x %*% cbind(A=c(1,1), M=c(-1,1))
> plot(x, pch=".")

This is one array diff in intensities vs. sum of intensities?
Hexagonal Binning in R

R code:

```r
> library("hexbin")
Loading required package: grid
Loading required package: colorspace

Attaching package: 'colorspace'

The following object(s) are masked from package:grDevices :

hcl

> library("geneplotter")
Loading required package: annotate
KernSmooth 2.22 installed
Copyright M. P. Wand 1997
> hb <- hexbin(x, xbins=50)
> library("RColorBrewer")
> plot(hb,
  colramp=colorRampPalette(brewer.pal(9,"YlGnBu")[-1]))

-1 removes the first value in this color palette because it is too close to white.
```
library("prada")
smoothScatter(x, nrpoints=500)

nrpoints is the number of points to be superimposed on the density image.
densCols

plot(x, 
    col=densCols(x), pch=20)

Computes the local density for each point and returns a false color representation.
Side by Side Plot

BINF733 Spring06 Solka/Weller -
Visualizing Data
A Few Notes on Performance

- pdf or ps files involving large numbers of points in a scatterplot can take a great deal of time to render.
- Advantages of binning
  - Long drawing times reduced once the bin counts have been computed
- A careful choice of bins can be used to replace observations prior to smoothing.
  - One can obtain good fitted curve performance assuming a decent choice of the bin locations.
  - Smooth uses centers of bins as exemplars.
Agglomerative Algorithm (Bottom Up or Clumping)

Start: Clusters $C_1$, $C_2$, ..., $C_n$ each with 1 data point

1 - Find nearest pair $C_i$, $C_j$, merge $C_i$ and $C_j$, delete $C_j$, and decrement cluster count by 1

If number of clusters is greater than 1 then go back to step 1
Intercluster Dissimilarity Choices

- Furthest Neighbor (Complete Linkage)
- Nearest Neighbor (Single Linkage)
- Group Average
Dendrograms

Agglomerative

0 1 2 3 4
(1) (1,2) (1,2,3,4,5) (2) (3,4,5) (3) (4) (4,5)

Divisive

4 3 2 1 0
BINF733 Spring06 Solka/Weller - Visualizing Data
FIGURE 10.11. A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters. Here, at level 1 all eight points lie in singleton clusters; each point in a cluster is highly similar to itself, of course. Points $x_6$ and $x_7$ happen to be the most similar, and are merged at level 2, and so forth. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Heatmaps


- Semiology of graphics (Hardcover) by Jacques Bertin, 1983
Heat Map (Data Imaging)\nMotivation

- How does one identify structure in large data sets in high dimensional space?
  - Pairs plot
    - n-dimensional data implies n x n plots

- Given a proposed clustering of such a data set then can one devise a method to allow the human visual system to assess the clustering?
Scatterplot Matrices

- This is a way of examining plots all possible pairs of variates (variables)

```r
> data(iris)
> pairs(iris[1:4], main = "Anderson's Iris Data -- 3 species",
+       pch = 21, bg = c("red", "green3", "blue") [codes(iris$Species)])
```
Data Imaging Approach

- Start with a spreadsheet of data with n rows and p columns
  - Each row represents an observation and each column represents the range of values for a particular variable

- Normalize the data as desired

- Compute interpoint distances on the data

- Use your favorite clustering procedure to obtain an ordering of the data set

- Transpose the matrix and render it with low values in one color and high values in another
Compelling Data Image Example

90 observations of 100 dimensional mean 0 variance 1 data observations 1-30 shifted by 20 in their first dimension observations 31-60 shifted by 20 in their second dimension resultant matrix randomly transformed by multiplication by a 100 by 100 matrix of mean 0 variance 1 data resultant observations were randomly shuffled

scatter plot matrix of first first 5 variables
Data Images of 100 Dimensional Data

- Observations sorted by complete linkage
- Observations and variables sorted by complete linkage
library("ALL")

data("ALL")

# down select on two small subgroups
selSamples <- ALL$mol.biol %in% c("ALL1/AF4", "E2A/PBX1")

# here are the first three entries
> selSamples[1:3]
[1] FALSE FALSE FALSE FALSE

ALLs <- ALL[, selSamples]
heatmap in R - II

> ALLs$mol.biol <- factor(ALLs$mol.biol)

#Here is what they look like
> ALLs$mol.biol
[1] ALL1/AF4 E2A/PBX1 ALL1/AF4 ALL1/AF4 ALL1/AF4 ALL1/AF4
   ALL1/AF4 E2A/PBX1 ALL1/AF4
[9] E2A/PBX1 ALL1/AF4 ALL1/AF4 ALL1/AF4 ALL1/AF4 E2A/PBX1
   ALL1/AF4 E2A/PBX1
Levels: ALL1/AF4 E2A/PBX1

colnames(exprs(ALLs)) <- paste(ALLs$mol.biol, colnames(exprs(ALLs)))
library("genefilter")

meanThr <- log2(100)

g  <- ALLs$mol.biol

s1 <- rowMeans(exprs(ALLs)[, g==levels(g)[1]]) > meanThr
s2 <- rowMeans(exprs(ALLs)[, g==levels(g)[2]]) > meanThr
s3 <- rowttests(ALLs, g)$p.value < 0.0002

selProbes <- (s1 | s2) & s3

ALLhm <- ALLs[selProbes,]
heatmap in R - IV

```r
> exprs(ALLhm)[1:2,]

   ALL1/AF4 04006 E2A/PBX1 08018 ALL1/AF4 15004 ALL1/AF4 16004
1007_s_at  6.816397  7.151422  6.822427  6.709222
1044_s_at  4.570669  7.019295  4.892009  4.889920

   ALL1/AF4 19005 ALL1/AF4 24005 E2A/PBX1 24019 ALL1/AF4 26008
1007_s_at  6.798443  6.277473  8.554938  6.334291
1044_s_at  4.339371  5.358229  6.770408  4.191569

   E2A/PBX1 28003 ALL1/AF4 28028 ALL1/AF4 28032 ALL1/AF4 31007
1007_s_at  8.253946  6.622969  6.601746  6.358026
1044_s_at  7.015059  5.192337  5.103668  4.754665

   E2A/PBX1 36001 ALL1/AF4 63001 E2A/PBX1 LAL5
1007_s_at  7.460446  6.441714  7.477596
1044_s_at  6.556056  4.615990  7.303723

> dim(exprs(ALLhm)[,])
[1] 81 15

#81 probes by 15 cases
```
heatmap in R - V

```r
> hmcol <- colorRampPalette(brewer.pal(10, "RdBu"))(256)
> spcol <- ifelse(ALLhm$mol.biol=="ALL1/AF4", "goldenrod", "skyblue")
> heatmap(exprs(ALLhm), col=hmcol, ColSideColors=spcol)
```
Heatmaps of Residuals - I

- \( Y = f(x) + \varepsilon \)
  - \( Y \) is the observed data and \( x \) are the explanatory variables.
- An examination of the residuals can often leads to insights into the nature of the fit.

\[
\hat{\varepsilon} = Y - \hat{f}
\]
Heatmaps of Residuals - II

- estrogen data
  - 8 samples from an estrogen positive breast cancer cell line
  - After serum starvation, four samples were exposed to estrogen and then harvested for analysis with Affymetrix human genome U-95Av2 after 10 hours for two samples and 48 hours for the other two.
Heatmaps of Residuals - III

- For each probe set we have
  - 8 measurements
  - 4 coefficients
    - Overall baseline
    - Estrogen stimulation (+ = yes, - = no)
    - Time effect (10h, 48h)
  - Interaction between treatment and time
    - Difference in treatment effect between 10h and 48h
- This leaves four residual degrees of freedom for each probe set
So we wish to compare the expression values that are computed by our model against the actual expression values.
Heatmaps of Residuals - V

```r
> esEset <- cache("metaVisualize-esEset", {  
+   library("estrogen")
+   library("limma")
+   library("hgu95av2cdf")
+   datadir <- system.file("extdata", package="estrogen")
+   targets <- readTargets("phenoData.txt", path=datadir, sep="")
+   covdesc <- list("present or absent","10 or 48 hours")
+   names(covdesc) <- names(targets)[-1]
+   pdata <- new("phenoData", pData=targets[, -1], varLabels=covdesc)
+   rownames(pData(pdata)) <- targets[, 1]
+   esAB <- ReadAffy(filenames=file.path(datadir, targets$filename), phenoData=pdata)
+   esEset <- rma(esAB)
+ })
```
Heatmaps of Residuals - VI

Loading required package: hgu95av2
Loading required package: hgu95av2cdf
Loading required package: vsn
Background correcting
Normalizing
Calculating Expression
Heatmaps of Residuals - VII

```r
> fit <- cache("metaVisualize-fit", { +   pdat <- pData(esEset) +   design <-
+     model.matrix(~factor(estrogen)*factor(time.h),pdat) +   colnames(design) <- c("Intercept","ES","T48","ES:T48") +   lmFit(esEset,design) + })
> stopifnot(all(fit$df.residual==4)) $>
> colnames(exprs(esEset)) <- paste( +   c("-", "+")[match(esEset$estrogen, c("absent", "present")), esEset$time.h) +   predict.MArrayLM <- function(f, design=f$design) { +     return(f$coefficients %*% t(design)) +   }
```
Heatmaps of Residuals - VIII

```R
> esFit <- predict(fit)
> res <- exprs(esEset) - esFit
> sel <- order(fit$coefficients[, "ES:T48"], decreasing=TRUE)[1:50]
> four.groups <- as.integer(factor(colnames(exprs(esEset))))
> csc <- brewer.pal(4, "Paired")[four.groups]
> heatmap(exprs(esEset)[sel,], col=hmcol, ColSideColors=csc)
```
Heatmaps of Residuals - IX

> heatmap(res[sel,],
  col=hmcol,
  ColSideColors=csc)
Visualizing Distances

- We will have much more to say about distances and clustering/dendrograms later.
- Dendrograms impose an ordering on the data based on the sequence of merges in hierarchical agglomerative clustering.
- Cophenetic correlation can be used to measure the association between two different distance measures.
  - cophenetic in R
Visualizing Distances in the ALL Dataset - I

```r
standardize <- function(z) {
  rowmed <- apply(z, 1, median)
  rowmad <- apply(z, 1, mad)
  # This sweeps (subtracts) the median from each row entry.
  rv <- sweep(z, 1, rowmed)
  # This sweeps (divides) the mad from each row entry.
  rv <- sweep(rv, 1, rowmad, "/")
  return(rv)
}

ALLhme <- exprs(ALLhm)
# We want distances between the cases
# This uses Euclidean distance by default
ALLdist1 <- dist(t(standardize(ALLhme)))
# Apply hierarchical clustering
ALLhcl <- hclust(ALLdist1)
```
plot(ALLhc1, xlab="", sub="", main="ALLhc1")

#It is not surprising that down selecting based on t-statistic gives us good class separation
# In this section we are going to select probes based on just median absolute deviation (MAD) across all of the samples.

```r
# Select probes based on MAD
ALLsub2 <- exprs(ALLs[(s1 | s2), ])
rowMads <- apply(ALLsub2, 1, mad)
ALLsub2 <- ALLsub2[rowMads > 1.4, ]

# Compute distances
ALLdist2 <- dist(t(standardize(ALLsub2)))
ALLhc2 <- hclust(ALLdist2)
```
Visualizing Distances in the ALL Dataset - IV

We will have more to say about interpreting these when we discuss clustering.
Cophenetic Distance Calculation - I

> ALLcph1 <-
  cophenetic(ALLhc1)
> cor(ALLdist1, ALLcph1)
[1] 0.9901792
> library(RColorBrewer)
> hmcol <-
  colorRampPalette(brewer.pal(10, "RdBu"))(256)
> plot(ALLdist1, ALLcph1,
pch="|", col="blue")
Cophenetic Distance Calculation - II

```r
ALLcph2 <- cophenetic(ALLhc2)
cor(ALLdist2, ALLcph2)
# the value here is smaller
# i did not save it though
> plot(ALLdist2, ALLcph2, pch="|", col="blue")
```
Multi-dimensional Scaling (MDS)

- How can we evaluate the results of clustering high-dimensional observations?
  - The dissimilarity measure that we used may not be a metric.
  - We can’t draw a picture of the clustering in the high-dimensional space.
  - Can we project to a lower dimensional space while preserving the distance relationships among the observations.
    - This is the focus of multidimensional scaling (MDS)
The Setup

- \( x_1, x_2, \ldots, x_n \) are the original d-dimensional observations with associated distances \( \delta_{ij} \)

- \( y_i \) is the lower dimensional representation of \( x_i \) and the distance between \( y_i \) and \( y_j \) is given by \( d_{ij} \)

- We wish to find a configuration of the \( y_i \) such that the \( \delta_{ij} \) are as close as possible to the \( d_{ij} \)

- In general we can’t guarantee equality
FIGURE 10.26. The figure shows an example of points in a three-dimensional space being mapped to a two-dimensional space. The size and color of each point $x_i$ matches that of its image, $y_i$. Here we use simple Euclidean distance, that is, $\delta_{ij} = \|x_i - x_j\|$ and $d_{ij} = \|y_i - y_j\|$. In typical applications, the source space usually has high dimensionality, but to allow easy visualization the target space is only two- or three-dimensional. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
MDS Criteria Functions

\[ J_{ee} = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2} \]

\[ J_{ff} = \sum_{i<j} \left( \frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2 \]

\[ J_{ef} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}} \]

- All criteria are invariant to rigid body motions of the points
- Invariant to dilations of the points
- \( J_{ee} \) emphasizes errors regardless of the size of the \( \delta_{ij} \)
- \( J_{ff} \) emphasizes large fraction errors regardless of whether \( |\delta_{ij} - d_{ij}| \) is large or small
- \( J_{ef} \) is a compromise
MDS Procedure

- Choose Criteria

- Choose an initial configuration of the $y_i$'s
  - Randomly
  - Based on the $\hat{d}$ coordinates with the largest variance

- Move the points in the direction of the greatest decrease in the criteria function
Estimation of the Gradients

\[ \nabla y_k J_{ee} = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \left( d_{kj} - \delta_{kj} \right) \frac{y_k - y_j}{d_{kj}} \]

\[ \nabla y_k J_{ff} = 2J_{ff} = \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{y_k - y_j}{d_{kj}} \]

\[ \nabla y_k J_{ef} = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{y_k - y_j}{d_{kj}} \]
A Word About Gradient Descent

- Suppose we wish to solve $a^T y_i > 0$
- Attack the problem by defining a criteria function $J(a)$ that is minimized if $a$ is a solution vector

Algorithm 1. (Basic Gradient Descent)
1. Begin initialize $a$, threshold $\theta$, $\eta(.)$, $k=0$
2. Do $k=k+1$
3. $a \leftarrow a - \eta(k) \nabla J(a)$
4. Until $|\eta(k) \nabla J(a)| < \theta$
5. Return $a$
6. end
MDS From $R^3$ to $R^2$ - Example 2

**FIGURE 10.27.** Thirty points of the form $x = (\cos(k/\sqrt{2}), \sin(k/\sqrt{2}), k/\sqrt{2})^t$ for $k = 0, 1, \ldots, 29$ are shown at the left. Multidimensional scaling using the $J_{ef}$ criterion (Eq. 109) and a two-dimensional target space leads to the image points shown at the right. This lower-dimensional representation shows clearly the fundamental sequential nature of the points in the original source space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.
Classical Multidimensional Scaling in R - I

- `cmdscale` in `stats` package
  - Uses classical MDS with a least squares definition of energy $J_{ee}$
  - Computes a singular value decomposition of the double centered matrix of squared distances
  - Solutions are nested (first two dimensions in $k = 3$ match the $k = 2$ solutions)
Classical Multidimensional Scaling in R - II

- Goodness of fit (GOF)
  - For each dimension
    - Sum of the eigenvalues for the components S divided by the sum of the absolute value of all eigenvalues
    - Sum of the eigenvalues for the components S divided by the sum of all positive eigenvalues
  - Examine scree plot and look for the elbow in the curve
Non-Metric MDS in R - I

- isoMDS in MASS
  - Chooses a k-dimensional space to minimize

\[ s^2 = \frac{\sum_{i \neq j} |f(p_{ij}) - d_{ij}|^2}{\sum_{i \neq j} d_{ij}^2} \]

where \( p_{ij} \) is the original distance matrix, \( f \) is a monotonic transformation, and \( d_{ij} \) are the distances between the MDS points. \( s \) is also called the stress.
Non-Metric MDS in R - II

- `sammon` in `MASS`

- Uses a different loss function.
cmdscale and sammon in R

```r
> library(MASS)
> cm1 <- cmdscale(ALLdist1, eig=TRUE)
> cm1$GOF
[1] 0.9083166 0.9083166
> samm1 <- sammon(ALLdist1, trace=FALSE)
> cm2 <- cmdscale(ALLdist2, eig=TRUE)
> cm2$GOF
[1] 0.6456937 0.6456937
> samm2 <- sammon(ALLdist2, trace=FALSE)
```
Setting Up Colors for the MDS Plots

```r
> ALLscol <- c("goldenrod", "skyblue") [as.integer(ALLs$mol.biol)]
> ALLscol
[1] "goldenrod" "skyblue" "goldenrod" "goldenrod" "goldenrod" "goldenrod" "skyblue"
[8] "goldenrod" "skyblue" "goldenrod" "goldenrod" "goldenrod" "skyblue" "goldenrod"
[15] "skyblue"
> ALLs$mol.biol
[1] ALL1/AF4 E2A/PBX1 ALL1/AF4 ALL1/AF4 ALL1/AF4 ALL1/AF4 ALL1/AF4
     E2A/PBX1 ALL1/AF4 E2A/PBX1 ALL1/AF4 E2A/PBX1 ALL1/AF4 ALL1/AF4
[11] ALL1/AF4 E2A/PBX1 ALL1/AF4 E2A/PBX1 ALL1/AF4 E2A/PBX1
Levels: ALL1/AF4 E2A/PBX1
> as.integer(ALLs$mol.biol)
[1] 1 2 1 1 1 1 2 1 2 1 1 1 2 1 2
```
Creating the MDS Plots in R

```r
> myPlot <- function(x, ...) 
  plot(x$points, xlab="Component 1", 
       ylab="Component 2", pch=19, col=ALLscol, ...)

> par(mfrow=c(2,2))
> myPlot(cm1, main="a) metric / t-test")
> myPlot(samm1, main="b) Sammon / t-test")
> myPlot(cm2, main="c) metric / MAD")
> myPlot(samm2, main="d) Sammon / MAD")
```
The MDS Plots in R

Why the nice separation?
```r
> cm1.3 <- cmdscale(ALLdist1, eig=TRUE, k=3)
> cm1.3$GOF
[1] 0.9347868 0.9347868
> library(scatterplot3d)
> scatterplot3d(cm1.3$points, color=ALLscol, xlab = "Component 1", ylab = "Component 2", zlab = "Component 3")
```
The Parallel Coordinates Framework (Points Become Lines)

Parallel Coordinates Plot of Clustered Data
Salad Plot For All Four Classes
Parallel Coordinates Plot of the ALL Data

```r
> cm1.4 <- cmdscale(ALLdist1, eig=TRUE, k=4)
> cm1.4$GOF
[1] 0.9481212 0.9481212
```
Distance Measures on Manifolds

Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (A) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) The neighborhood graph \( G \) constructed in step one of Isomap (with \( K = 7 \) and \( N = 1000 \) data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in \( G \). (C) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).
The ISOMAP Algorithm

Table 1. The Isomap algorithm takes as input the distances $d_X(i,j)$ between all pairs $i,j$ from $N$ data points in the high-dimensional input space $X$, measured either in the standard Euclidean metric (as in Fig. 1A) or in some domain-specific metric (as in Fig. 1B). The algorithm outputs coordinate vectors $y_i$ in a $d$-dimensional Euclidean space $Y$ that (according to Eq. 1) best represent the intrinsic geometry of the data. The only free parameter ($\epsilon$ or $K$) appears in Step 1.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Construct neighborhood graph</td>
</tr>
<tr>
<td>2</td>
<td>Compute shortest paths</td>
</tr>
<tr>
<td>3</td>
<td>Construct $d$-dimensional embedding</td>
</tr>
</tbody>
</table>

ISOMAP in Action

- Ref. Bioinformatics ISOMAP paper and snapshot some pictures.
Plotting Along Genomic Coordinates - I

- These tools relate gene expression to chromosomal location

- DNA
  - Sense strand (Watson or + strand)
  - Antisense strand (Crick or – strand)
> library("geneplotter")
Loading required package: annotate
KernSmooth 2.22 installed
Copyright M. P. Wand 1997

> chrLoc <- buildChromLocation("hgu95av2")
Loading required required package: hgu95av2
Compute the mean expressions for each probe.

```r
ALLch <- ALLs[s1|s2, ]

m1  <- rowMeans(exprs(ALLch)[, ALLch$mol.biol=="ALL1/AF4"])

m2  <- rowMeans(exprs(ALLch)[, ALLch$mol.biol=="E2A/PBX1"])
```
Compute the deciles of the combined data

```r
> deciles <- quantile(c(m1,m2),
                    probs=seq(0,1,.1))
> s1dec <- cut(m1, deciles)
> s2dec <- cut(m2, deciles)
> gN <- names(s1dec) <- names(s2dec) <- geneNames(ALLch)
```
Plotting Along Genomic Coordinates - V

```r
> colors <- brewer.pal(10, "RdBu")
> layout(matrix(1:3, nr=1), widths=c(5,5,2))
> cPlot(chrLoc, main="ALL1/AF4")
> cColor(gN, colors[s1dec], chrLoc)
Warning message:
is.na() applied to non-(list or vector) in: is.na(locs)
> cPlot(chrLoc, main="E2A/PBX1")
> cColor(gN, colors[s2dec], chrLoc)
Warning message:
is.na() applied to non-(list or vector) in: is.na(locs)
> image(1,1:10,matrix(1:10,nc=10),col=colors,
        axes=FALSE,
        + xlab="", ylab="")
> axis(2, at=(1:10), labels=levels(s1dec), las=1)
```
Plotting Along Genomic Coordinates - VI
Plotting Along Genomic Coordinates - VII

```r
> par(mfrow=c(1,1))
> msobj <- Makesense(ALLs, "hgu95av2")
> plotChr("22", msobj,
+   col = ifelse(ALLs$mol.biol=="ALL1/AF4",="#EF8A62", "#67A9CF"), log=FALSE)
```