Introduction to VisNAB: An Interactive Toolkit for Visualizing and Exploring Genomic Data

Tengfei Yin

March 22, 2011

Contents

1 Introduction

VisNAB(VisNAB is Not A Browser) is a package(called "visnab") developed in R, which is designed for visualization of genomic data, especially for next generation sequencing data. Besides traditional track-based views,VisNAB also aims to offer alternative views, including Bird-Eye overview, circular view, etc. At the same time, it also provides statistical cues and analytical tools for guiding scientists to find the information of interests. With support of QT library and related API in R, it's possible to make interactive graphics for visualizing and exploring huge amount of data with high performance. This toolkit with other packages in Bioconductor could form a powerful analytical pipeline for exploring the genomic data.

2 Infrastructure

We have two types of graphic parameters, one is individual graphic attributes, A *GraphicPars* class is defined to keep those attributes, such as background color, foreground color, fill color, stroke color, etc. So far, the *GraphicPars* is an environment with a set of accessors.

The other type of attributes is designed to specify observation oriented behavious. Each observation has some extra attributes that indicate some specific status, such as selected or not, brushed or not, highlighted or not. To achieve tihs goal, mutable data structure is preferred, so any signal could be attached to it conveniently. For genomic data, we use *MutableRanges* package to handle this basic structure, which is a `MutableGRanges` object, this object has common accessors for `GRanges` object, and what's more, it's mutable. So signal could be attached to this data structure. For common data structure, we use `mutalist` and `mutaframe` in *plumbr* package. For more details about these mutable data structure and signal handling method, please refer to the documentation of package *MutableRanges* and *plumbr*.

visnab is a pacakge which provides sets of views for visualization of genomic data, especially the next generation data. For fast and interactive graphics in R, we use QT API, please refer to documentation of package *qtbase* and *qtpaint* for more details.

VisnabView is a virtual class which define a top level view object in *visnab*, it contains basic information like graphic parameters, show status, etc. *QtVisnabView* is a virtual class which contains *VisnabView* and has extra information about Qt device, this object will store scene, view and rootLayer object and also some slots to control `row, col, rowSpan, colSpan` This feature may have potential use for integration of multiple views later.

Then for other individual views, we define specific view class, such as *IntervalView, CircularView, StackedView*,etc, which will be described in details in following sections.

3 Individule View

All the individual views inherits from *QtVisnabView*, which contains common slots for storing Qt device, like scene, view, rootLayer object, and a set of graphic parameters controling the layout of the view and a `MutableGRanges` object called viewrange to control the viewed object. This design aims to update associated scene automatically and handle `Signal` in an easy way.

Constructor which is consistency with class name, usually return a view object, then a generic function `print` could draw nessessary elements and show the view. It's possible to pass specific scene to view object and implement them in a user defined layout.

The design and infrastructure of *visnab* make writing a generic GUI function to each view object possible and easy. GUI development may not be part of *visnab* package. A new package called *visnabGUI* may need to be developed on seperate repos, `Qt` based GUI is preferred since this will make installation easy and make the whole package reply only on *qtbase*.

3.1 Circular View

Circular view aims to visualize list of `MutableGRanges` in a cicular layout, each `MutableGRanges` object occupy one circle in a specific graphic types, such as bar chart, rectangle, points. And inside the circle, a linking curve could be added if you have any kinds for translocation information. This idea is inspired by circos project. A zoomable view is build too.

A `CircularView` object could be constructed by related constructors easily.

Some common features is provided for this view:

- Associated zoom-in window
- Order change of chromosomes
- Types change of trakcs
- Linking lines highlight

examples goes here.

3.2 Interval View

Interval view aims to visualize single `MutableGRanges` object, we could consider it as a single track which is used for showing featues or components along the genomic coordinates, such CDS, exons, transcripts.

A `MutableGRanges` object may contains features about the whole genomes, which may have multiple chromosomes, the `viewrange` aim to control which range is going to be visualized.

Some common features is provided for this view:

- Mouse wheel controled zoom-in/out
- Tooltip popup when mouse move over one component
- idname could control which information you want to show in tooltips

Examples goes here.

3.3 Stacked View

Stacked view aims to visualize the whole genome, and stack all the chromosomes. When interaction happenedd along other views or filter, regions of interests will be highlighted.

When equiped with *rtracklayer*, it's possible to visualize over 80 genomes very easily.

Some common features is provided for this view:

- Mouse hover single chromosome showing coordinates
- Order change of chromosomes
- Region selection
- Segement highlight on the chromosomes

Examples go here.

3.4 Alignment View

Alignment view aims to provide shortread alignment with semantic zoom.

3.5 Seq View

Seq view aims to provide reference genome and coordinates, could be implemented in the tracks view, please see the Section 3.7.

3.6 Pileup View

Pileup view aims to show information about NGS data such as coverage, SNP statistics.

3.7 Tracks View

Tracks view aims to visualize multiple tracks at the same time, it's usually an integration of the following views:

- Interval View
- Seq View
- Alignment View
- Pile-up View

4 Linking for Multiple Views

5 Case Study

Appendix

```
> sessionInfo()

R version 2.13.0 Under development (unstable) (2011-03-07 r54691)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.utf8       LC_NUMERIC=C
 [3] LC_TIME=en_US.utf8        LC_COLLATE=en_US.utf8
 [5] LC_MONETARY=C             LC_MESSAGES=en_US.utf8
 [7] LC_PAPER=en_US.utf8       LC_NAME=C
 [9] LC_ADDRESS=C              LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.utf8 LC_IDENTIFICATION=C

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