Getting started with **genoPlotR**

Lionel Guy

April 30, 2010

Contents

Introduction

The **genoPlotR** package is intended to produce publication-grade graphics of gene and genome maps. With the amazing speed of data production of new DNA sequencing techniques and the increase in the number of software available to compare these sequences, there is a great need to graphically represent these sequences and their comparisons. A number of packages already exist (Artemis, ACT, mauve), but none of them produces easily reproducible, publication-grade graphics. The goal of this package is to fill in that gap.

This document provides an introduction to **genoPlotR**, providing the user with examples of increasing complexity. It is not meant as a comprehensive

guide otto all the functions and options of the package, but rather as a first approach to the package.

To load the library in a R session, type:

```
> library(genoPlotR)
```

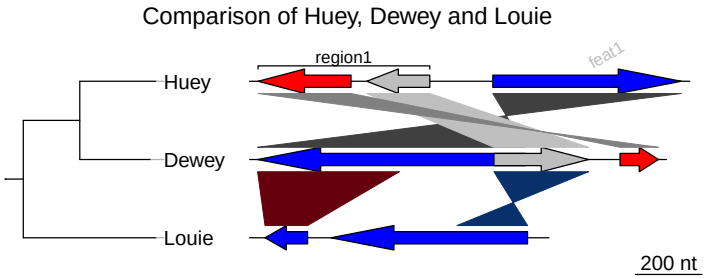
1 Quick start

Loading the simplest dataset, applying a color scheme and some limits to the plotting aread, adding a tree and some annotations.

```
> data(three_genes)
> comparisons[[1]]$col <- apply_color_scheme(c(0.6, 0.4, 0.5),
+     "grey")
> names <- c("Huey", "Dewey", "Louie")
> names(dna_segs) <- names
> tree <- newick2phylog("(((Huey:4.2,Dewey:3.9):3.1,Louie:7.3):1);")
> mid_pos <- middle(dna_segs[[1]])
> xlims <- list(c(Inf, -Inf), c(-Inf, Inf), c(1850, 2800))
> annot <- annotation(x1 = c(mid_pos[1], dna_segs[[1]]$end[2]),
+     x2 = c(NA, dna_segs[[1]]$end[3]), text = c(dna_segs[[1]]$name[1],
+         "region1"), rot = c(30, 0), col = c("grey", "black"))
```

Now plotting these three segments:

```
> plot_gene_map(dna_segs = dna_segs, comparisons = comparisons,
+     annotations = annot, annotation_height = 1.3, tree = tree,
+     tree_width = 2, xlims = xlims, main = "Comparison of Huey, Dewey and Louie")
```



2 Getting help

There are various ways to get help with **genoPlotR**. First, you can start the general help in a web browser, and then click on "packages" and find **genoPlotR** in the list. It will provide a list of the functions available. The first link in the list leads to a very general description of the package.

```
> help.start()
```

Another way of obtaining the list of functions present in the package is to run

```
> library(help = genoPlotR)
```

A lot of examples and help are available in the main functions, i.e. the reading functions (the various `read_dna_seg_from*` and `read_comparison_from*` functions) and the main plotting function, `plot_gene_map`.

```
> help("read_functions")
> help("plot_gene_map")
```

Finally, the web page on R-forge (`http://genoplotr.r-forge.r-project.org/`) provides ways to get in touch with the **genoPlotR** community and to submit bugs and feature requests.

3 Objects in genoPlotR

This section will give an overview of the different types of R objects in **genoPlotR**.

3.1 dna_seg

A `dna_seg` object are is a collection of genes or elements along a genome, to be represented on a map.

`dna_seg` objects need to have 4 columns, `name`, `start`, `end` and `strand`. Extra columns with names `col`, `lty`, `lwd`, `pch`, `cex`, `gene_type` will be used in the plotting process. Other extra columns will be kept in the object, but not used.

```
> names1 <- c("feat1", "feat2", "feat3")
> starts1 <- c(2, 1000, 1050)
> ends1 <- c(600, 800, 1345)
> strands1 <- c("-", -1, 1)
> cols1 <- c("blue", "grey", "red")
> df1 <- data.frame(name = names1, start = starts1, end = ends1,
+     strand = strands1, col = cols1)
> dna_seg1 <- dna_seg(df1)
> str(dna_seg1)

Classes âĂŸdna_segâĂŹ and 'data.frame':        3 obs. of  10 variables:
 $ name     : chr  "feat1" "feat2" "feat3"
 $ start    : num  2 1000 1050
 $ end      : num  600 800 1345
 $ strand   : num  -1 -1 1
 $ col      : chr  "blue" "grey" "red"
 $ lty      : num  1 1 1
 $ lwd      : num  1 1 1
 $ pch      : num  8 8 8
 $ cex      : num  1 1 1
 $ gene_type: chr  "arrows" "arrows" "arrows"
```

3.2 comparison

A `comparison` is a collection of similarities, representing the comparison between two DNA segments.

Objects (either data frames or lists) should have at least named elements `start1`, `end1`, `start2` and `end2`. In addition, it can use a `color` column to give. Additional numeric columns can be used for color-coding (via `apply_color_scheme`.

```
> starts1 <- c(2, 1000, 1050)
> ends1 <- c(600, 800, 1345)
> starts2 <- c(50, 800, 1200)
> ends2 <- c(900, 1100, 1322)
> comparison1 <- as.comparison(data.frame(start1 = starts1, end1 = ends1,
+     start2 = starts2, end2 = ends2))
> str(comparison1)

Classes âĂŸcomparisonâĂŹ and 'data.frame':        3 obs. of  5 variables:
 $ start1   : num  2 1000 1050
 $ end1     : num  600 800 1345
 $ start2   : num  50 800 1200
 $ end2     : num  900 1100 1322
 $ direction: num  1 -1 1
```

3.3 annotation

An `annotation` object is used to annotate a DNA segment. It has labels attached to positions. Each label can be attached to a single position or to a range.

```
> mid_pos <- middle(dna_segs[[1]])
> annot1 <- annotation(x1 = mid_pos, text = dna_segs[[1]]$name)
> str(annot1)

Classes âĂŸannotationâĂŹ and 'data.frame':        3 obs. of  5 variables:
 $ x1   : num  301 900 1198
 $ x2   : logi  NA NA NA
 $ text : chr  "feat1" "feat2" "feat3"
 $ color: chr  "black" "black" "black"
 $ rot  : num  0 0 0
```

3.4 tree

A tree description in Newick format can ba parsed using **ade4** package.

```
> tree <- newick2phylog("(((A_aaa:4.2,B_bbb:3.9):3.1,C_ccc:7.3):1);")
> str(tree$leaves)

 Named num [1:3] 4.2 3.9 7.3
 - attr(*, "names")= chr [1:3] "A_aaa" "B_bbb" "C_ccc"
```

4 Reading data

4.1 DNA segments

Several formats can be read by **genoPlotR** to produce `dna_seg` objects:

- EMBL files (`read_dna_seg_from_embl`)
- Genbank files (`read_dna_seg_from_genbank`)
- PTT (protein table) files (basically a tabular version of a Genbank file)(`read_dna_seg_from_ptt`)
- Tab files (user generated) (`read_dna_seg_from_ptt`)

The function `read_dna_seg_from_file` is a wrapper function, that will attempt to guess the correct format of the file.

The first three files are common biological formats and can be downloaded from major databases, such as the NCBI (`http://www.ncbi.nlm.nih.gov/`) and the EMBL (`http://www.ebi.ac.uk/embl/Access/index.html`). The definition of EMBL and Genbank files can be found at `http://www.ebi.ac.uk/embl/Documentation/FT_definitions/feature_table.html`.

4.2 Comparisons

genoPlotR can read tabular files, either user-generated tab files (`read_comparison_from_tab`), or from BLAST output (`read_comparison_from_blast`). To produce files that are readable by **genoPlotR**, the `-m 8` or `9` option should be used in `blastall`, or `-outfmt 6` or 7 with the BLAST+ suite.

4.3 Mauve output

The backbone output of the Mauve genome aligner (`http://asap.ahabs.wisc.edu/mauve/index.php`) can be parsed using `read_comparison_from_blast`[1].

The function will return a list consisting of a list of `dna_seg`s and the corresponding `comparisons`.

5 Plotting data

There is only one plotting function in **genoPlotR**, `plot_gene_map`. Many arguments are available, but here is a list of the most important. Check the documentation for a more thorough description.

`dna_segs` A list of DNA segment objects.

`comparisons` A list of comparisons. Should containt one element less than the previous.

`tree` An eventual phylogenetic tree to be plotted at the left of the figure.

`annotations` An annotation object, or a list of annotations. Will display annotations to the first, or to all DNA segments, respectively.

[1] Tested with Mauve 2.3.1

`xlims` A list of even-numbered numeric vectors, giving the borders of sub-segments to be plotted. The vector `c(1,5000,8000,6000` will display two sub-segments (1 to 5000 and 6000 to 8000), the second being in reverse orientation.

`main` A title to the plot.

`scale` Should a scale be displayed at the bottom right of the plot?

`dna_seg_scale` Allows to control the addition of scales to each segments. If simply `TRUE`, will display a scale on each segment. If a vector, a scale will be displayed for the corresponding `TRUE` element.

`global_color_scheme` Allows to recalcultate the colors of the comparisons, to have colors corresponding to the same scale for all comparisons.

`plot_new` Turn off to avoid creating a new plot. Escpecially useful to integrate a **genoPlotR** plot in a larger figure.

6 Other useful functions

`apply_color_scheme` Allows to apply a gray scale or hues of red and blue to a comparison.

`middle` Useful to get the middle of a gene, especially to create annotations.

Datasets Type `data(package="genoPlotR")` to get the full list.

7 Examples

7.1 A very simple example

7.2 Mauve alignment of four *Bartonella* genomes

7.3 Several sub-segments of four *Bartonella* genomes

7.4 Two segments of the Y chromsome in human and chimp

7.5 Generating data online