**Beginner’s Guide to genoPlotR**

**By Deborah Buhlers**

**Contents**

**Background**

**Key decisions in using genoPlotR to make graphical representations of genes and genomic comparisons**

**Objects, important parameters and functions in genoPlotR**

**Obtaining genoPlotR software**

**How to get Help**

**Generating sequence data for GenoPlotR R**

**Reading in *Rhizobium leguminosarum* bv. *viciae* 3841 and VF39SM genomic data from NCBI into GenoPlotR R in R Studio**

**Comparison of VF39c and d scaffolds to pRL9 and 10 from BLAST results**

**Plotting of individual genes of VF39 c and d scaffolds**

**Annotation of gene segments**

**Genome level comparison: Comparison of four *Bartonella* genomes**

**References**

**Appendix with sample code and graphical outputs**

**Background**

With the advances in DNA sequencing, entire microbial genomes are rapidly and economically being sequenced. Sequence data is then deposited in publically available databases, including National Center for Biotechnology Information (NCBI) and the European Molecular Biology Laboratory (EMBL). Researchers search these databases and utilize genome sequence information from these databases for experiments involving a microorganism of interest, which could be used to produce a microbial based product for either an industrial, agricultural or bioremediation applications. A catabolic gene, or groups of catabolic genes that could serve the purposes of the previously mentioned applications are searched for in microbial genome databases, with the appropriate microorganism of interest being enriched for and isolated from an environmental sample before further genetic and metabolic capabilities are characterised. For example, a search of the NCBI database for the catabolic gene “cellulase” results in many microorganisms that contain this gene. A researcher might also be interested in the origin or mechanism by which this particular cellulase gene arose in the genome of the microorganism that they are studying. Another question that could be posed is what is the relationship of this particular microorganism to other cellulase containing microorganisms? Is there a common arrangement of cellulase genes and other genes to form catabolic gene operons or gene families that may suggest a mechanism by which this cellulase gene or genes arose in the microorganism of interest? These questions can be answered by plotting the catabolic gene region or other genes of interest from the microorganism being studied, and making graphical representations of these genes displaying regions of sequence similarities, based on sequence alignment results obtained from the NCBI BLAST webserver or user owned software such as DNAStar or DNAStryder. A tree can be constructed that displays the possible evolutionary relationships between the microorganism being studied and closely related microorganisms based on the sequence similarities between the catabolic gene(s) or other type of gene. Phylogenetic trees of bacterial species is usually based on alignments of 16S ribosomal RNA (rRNA) sequences, which display evolutionary relationships between bacterial genera and species when used to construct phylogenetic trees. Algorithms commonly used to construct a phylogenetic tree based on sequence data include the Unweighted Pair Group Method with Arithmetic Mean (UPGMA), alternatively, there is also the Neighbor Joining Method (Brock *et al.*, 2015). Optimality criteria is another method of constructing phylogenetic trees which includes parsimony, maximum likelihood and Bayesian analyses, which construct many possible phylogenetic trees based on the sequence data input, and then selects the tree with the best optimality score based on a model of molecular evolution (Brock *et al.*, 2015). However, many possible phylogenetic trees may fit a discrete evolutionary model equally, a major limitation of phylogenetic trees, but which can be corrected by randomly resampling the data, a method known as bootstrapping, which indicates the percentage of the time that a node in a tree is correct (Brock *et al.*, 2015).

GenoPlotR is an R package developed by Lionel Guy (Guy, *et al.*, 2010) for use in creating publication quality graphics of genomes, sub segments of genomes and their genes and comparisons between these genomes, genome sub segments and genes. The ability to graphically display genomes and comparisons from BLAST analysis, along with the ability to display relationships using phylogenetic trees in Newick format (follows a leaf format with the lengths of each branch corresponding to distance of that leaf to the parental node) is important for publishing genomic data analysis and disseminating this information to the scientific community in the field that study a particular microorganism.

(((One:0.2,Two:0.3):0.3,(Three:0.5,Four:0.3):0.2):0.3,Five:0.7):0.0;

+-+ One

+--+

| +--+ Two

+--+

| | +----+ Three

| +-+

| +--+ Four

+

+------+ Five

Example of Newick tree format (<http://evolution.genetics.washington.edu/phylip/newick_doc.html>)(Gary Olsen's Interpretation of the "Newick's 8:45" Tree Format Standard)

Genome viewer software packages such as Artemis and Mauve are capable of producing genome graphics, however, figures developed in these programs are not easily generated and not of publication quality. This beginner’s guide to genoPlotR provides a novice to R and the genoPlotR package, the steps necessary to download genoPlotR, how to use this program in R Studio, with examples from *Rhizobium leguminosarum* bv. *viciae* strains 3841 and VF39SM genomes from my research, along with sample genomes of four *Bartonella* sp. which come with this software package. In order to get started with genoPlot R, one must first have R and RStudio downloaded to their computer, which is freely available at the following website: <https://cran.r-project.org/> and select download R for your operating system and the CRAN mirror that is the server that is closest to your geographical location. To install R studio, use the following link: <https://www.rstudio.com/products/rstudio/download/> Click on the link on this page to download the version for your operating system.

**Key decisions in using genoPlotR to make graphical representations of genes and genomic comparisons**

In order to determine if genoPlotR is the best program to plot genes, or genomic segments and their comparisons, along with phylogenetic tree analysis, the following factors have to be considered before using this R package:

1. What is the intended purpose of the genomic comparisons or gene segments that will be plotted? Will these figures be used in a publication or presentation? If yes, then the genoPlotR package should be used. If the genomic similarities between microorganisms being compared does not have to be graphically represented, then all that is required is sequence alignments using either NCBI BLAST or user owned alignment software.

2. Is the genomic data collected and being compared or plotted in a format that can be read by the genoPlotR package? For example, the data must be in a Genbank or Genbank fasta format or PTT for protein table format or a text format from either NCBI or EMBL websites. User generated Excel spreadsheet files can also be read by genoPlotR.

3. Genes, and larger genomic sub segments and whole genome information that is to be analysed using genoPlotR should contain the following information: Gene names, start and end points of the genes within the larger DNA unit, and which strand the gene element occupies, either the coding sense strand of the DNA segment or the antisense negative DNA strand.

4. Are graphical parameters also included with the genomic data to be plotted or analysed in genoPlotR, such as col (color), lty (line type), lwd (line width), pch (point character type), cex (size of text or symbols in the graphical output), and gene\_type (side blocks, arrows, etc. ).

5. The user’s knowledge of the R language. If the user does not have previous experience with the R language, the project may have to be assigned to an individual, such as bio-informatics consultant, that does have knowledge of this programming language, which can increase costs of publishing the work.

**Objects, important parameters and functions in genoPlotR**

**This is a list of the most important objects and plotting parameters in the genoPlotR package**

1. **dna\_seg** - Genes or elements to be represented in a map. This can take on the form of a data frame if many genes or elements are to be labeled on a genome map. A minimum of four columns are required: name, start, end and strand. Other information that may be present and used to graph the gene or genes includes col (color), lty (line type), lwd (line width), pch (point character type), cex (size of text or symbols in the graphical output), and gene\_type (side blocks, arrows, etc. ).

2. **Comparison** - A data frame or list containing the similarities between DNA segments. As in the case of DNA seg, it must contain start1, start2, end1 and end2 for each DNA segment being compared. Also as in the case of DNA seg, there can be additional columns describing color, line type, width, point character type, and gene type. For a data frame containing comparison information of two or more genes, as.comparison is used to define the comparison.

3. **Annotation** - An object used to label a DNA segment. The user can determine where the label is inserted onto the DNA segment by using the command “middle” which will insert the label in the middle of the specified DNA segment range.

4. **Tree** – A phylogenetic tree in which the lengths of each branch demonstrates the degree of relatedness of organisms being compared (Newick format).

5. **xlims** – A list of numeric vectors which provides the coordinates of the sub segments plotted.

6. **main** – A character string used as the title for a plot.

7. **scale** – Bar in the right hand corner of a DNA segment plot that gives an indication of size of the DNA segment in the plot. Dna\_seg\_scale will put a scale on each dna\_seg.

8. **global\_color\_scheme**- Re-calculates colors of comparisons so that a color represents the same scale for all comparisons that are plotted.

9. **apply\_color\_scheme**- Applies grey scale or shades of blue and red to comparison plots between dna\_seg objects.

10. **plot\_gene\_map** – The function used to plot dna\_seg objects in genoPlotR. Usually takes the form of plot\_gene\_map(dna\_segs= , comparisons= , annotations= , annotation\_height= , tree= , tree\_width= , main= ).

**Obtaining genoPlotR software**

GenoPlotR software can be obtained by Google search for genoPlotR, which guides the user to the following url: <http://genoplotr.r-forge.r-project.org/> . On this page under instillation, there is the following link <http://cran.r-project.org/web/packages/genoPlotR/index.html> which leads the user to the page for downloading the genoPlotR software version required by their operating system (eg. Windows, or MacOS). The latest version of genoPlotR is the rdev version, as well, the ade4 package link at the top of this page must also be downloaded. Ade4 is required for plotting of phylogenetic trees in Newick format, and also for multiple variable statistical analysis (Dray *et al.* 2016). Once these two packages are downloaded into your R working directory, to load the genoPlotR library into a session in R studio, the following command must be typed

**library(genoPlotR)**

This loads both the genoPlotR and ade4 packages in order to begin working with this software.

**How to get help**

To get help with a specific function in genoPlotR, typing “**?function name**” will bring up a help page in the environment window of the R Studio session with instructions on how to use the function and general usage of the function.

To see a list of functions available in the genoPlotR package, type “**help.start()**” then select the genoPlotR package under **C:\Packages** available in the web browser window, this will bring you to a window with the PDF version of the package vignette with examples, as well, the Rcode link gives the code to the examples in the vignette. There are also help websites such as: <https://github.com/cran/genoPlotR/blob/master/inst/doc/genoPlotR.R> when searching for help with a question in a search engine window for genoPlotR.

**Generating sequence data for GenoPlotR R**

DNA nucleotide or protein amino acid files in one of the following formats: EMBL, Genbank, Genbank fasta (nucleotide file), Genbank PTT (protein amino acid sequence table) or Excel spreadsheet files generated by the user can be read into genoPlotR. These files are downloaded from EMBL (<http://www.ebi.ac.uk/>), or NCBI (<http://www.ncbi.nlm.nih.gov/>). From the NCBI website, DNA sequences can be downloaded by selecting the nucleotide database and then entering either an accession number for a gene of interest or name of the gene as a query of this database. Once the nucleotide sequence is located, click on the link to the sequence and select “send” in the upper right hand corner of the sequence information page in Genbank format, then select to “file” under choose destination, then select “FASTA” under format or for protein data, PTT ( protein table format), then select create file to download this information to your computer. User generated tables can be generated by the user researching DNA or protein amino acid sequence information from either of these websites and recording this information into a Microsoft Excel spreadsheet to customise the information obtained from these websites.

**Reading in *Rhizobium leguminosarum* bv. *viciae* 3841 and VF39SM genomic data from NCBI into GenoPlotR R in R Studio**

Once genomic data for *R. leguminosarum* bv. *viciae* 3841 and VF39SM strains had been downloaded from the NCBI Genbank website as fasta nucleotide files, they were aligned to each other using the NCBI blast analysis tool by selecting “align two or more sequences” option (<http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch&BLAST_SPEC=blast2seq&LINK_LOC=align2seq> ) and typing in the accession numbers for the downloaded sequence files in the blank windows on the Blast page or copying and pasting the nucleotide sequences into the two sequence windows and then clicking on the BLAST button at the bottom left hand corner of this web page. The similarities between the 3841 and VF39SM genomes, or genome comparison of any organisms, should be downloaded as a HitTable (text format) from the download link on the results page. The use of “try” in the “read\_dna\_seg\_from” command essentially tries to identify what format the data file is in so that it can be opened and read into an R Studio session and analysed by the genoPlotR package. The sequences and similarities between the VF39SM and 3841 genomes were “read” into genoPlotR R using the following code:

Entire plasmid sequences from strain 3841

**pRL10 <- try(read\_dna\_seg\_from\_fasta("pRL10.fasta"))**

**pRL9 <- try(read\_dna\_seg\_from\_fasta("pRL9.fasta")) # Reads in sequence data from the genome of Rhizobium leguminosarum bv. viciae strain 3841 plasmid pRL 9 and 10**

Corresponding scaffold sequences containing only the catabolic gene sub segments currently being studied on the c and d plasmids of VF39SM

**VF391213 <- try(read\_dna\_seg\_from\_fasta("VF39cscaffold.fasta")) #Reading in of R.leguminosarum bv. viciae strain VF39SM c plasmid catabolic gene scaffold**

**VF394041 <- try(read\_dna\_seg\_from\_fasta("VF39dscaffold.fasta")) #Reading in of R. leguminosarum bv. viciae strain VF39SM d plasmid catabolic gene scaffold sequence from NCBI database**

Then the following plots of the sizes of the catabolic gene regions on the c and d plasmids of strain VF39SM in relationship to the entire plasmids pRL9 and pRL10 in strain 3841 which contain the same catabolic genes, were produced with the following code which involves reading in the data obtained from NCBI BLAST results and then specifying the sizes of the two DNA segments to be plotted with the xlims command, and then plotting the gene map using the plot gene map command. Main specifies the title of the plot, gene type specifies the shape of the genes that will be mapped, and dna seg scale specifies that there will be a scale added to each DNA segment plotted, but no scale added to the bottom right hand corner of the final plot by specifying scale=FALSE:

**VF391213\_vs\_pRL9 <-try(read\_comparison\_from\_blast("VF39cscaffoldvspRL9-Alignment.txt"))** #Alignment of VF39c plasmid scaffold to pRL9 of 3841 in text format from the NCBI database.

**xlims <- list(c(1,352782),c(1,150833))** #sizes of the two DNA sequences being compared so that an appropriate x axis scale can be set in the plot.

**plot\_gene\_map(dna\_segs = list(VF391213,pRL9),**  #list because dna segment data frames being compared contain different classes of data, numeric and character.

**xlims=xlims,** #limit of the x axis in the final figure is as large as the largest dna segment in the figure.

**main="VF39cscaffold coverage of pRL9 of R.leg. 3841",** #Title of the graph

**gene\_type = "side\_blocks",** #The shape of the gene segment in the graphical output.

**dna\_seg\_scale = TRUE, scale=FALSE)** #scales put on each gene segment, not as a scale on the bottom of the figure.

**VF39dscaffold\_vs\_pRL10 <-try(read\_comparison\_from\_blast("VF39dscaffolvsdpRL10-Alignment.txt"))** #Alignment of VF39d plasmid scaffold to pRL10 of 3841 in text format from the NCBI database.

**xlims <- list(c(1,12453),c(1,488135))** #sizes of the two DNA sequences being compared so that an appropriate x axis scale can be set in the plot.

**plot\_gene\_map(dna\_segs = list(VF394041,pRL10),** #list because dna segment data frames being compared contain different classes of data, numeric and character.

**xlims=xlims,** #limit of the x axis in the final figure is as large as the largest dna segment in the figure.

**main="VF39dscaffold coverage of pRL10 of R.leg. 3841",** #Figure title

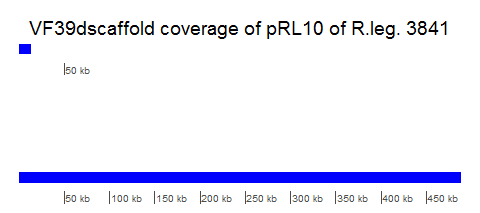
**gene\_type = "side\_blocks",** # Shape of the gene in the DNA segment in the figure

**dna\_seg\_scale = TRUE, scale=FALSE)** #scales put on each gene segment, not as a scale on the bottom of the figure.

The following plots being generated with titles specified by the “main” command:



**Figure 1. VF39c scaffold coverage of pRL 9 of *R. leguminosarum* bv. *viciae* strain 3841**



**Figure 2. VF39d scaffold coverage of pRL10 of *R. leguminosarum* bv. *viciae* strain 3841**

**Comparison of VF39c and d scaffolds to pRL9 and 10 from BLAST results**

Nucleotide blast results demonstrating similarities between sub segments of related genomes, which in this example, are the c and d scaffolds of *R. leguminosarum* strain VF39SM compared to the much larger plasmids pRL9 and pRL10 of the closely related *R. leguminosarum* strain 3841, can be graphically displayed using genoPlotR with the following code:

**VF391213 <- try(read\_dna\_seg\_from\_fasta("VF39cscaffold.fasta"))**

**pRL9 <- try(read\_dna\_seg\_from\_fasta("pRL9.fasta"))**

**VF391213\_vs\_pRL9 <-try(read\_comparison\_from\_blast("VF39cscaffoldvspRL9-Alignment.txt"))** #Reading in of data from BLAST alignment of the VF39SM c scaffold vs pRL9 plasmid from strain 3841

**xlims <- list(c(1, 400000), c(1, 400000))** #limits of the x axis in the final figure.

**plot\_gene\_map(dna\_segs = list(VF391213, pRL9),** #Asks R to plot a gene map of dna\_segments of the VF39SM c plasmid scaffold and plasmid pRL9 from strain 3841.

**comparisons=list(VF391213\_vs\_pRL9),** #The comparison of the VF39SM c scaffold to pRL9 from 3841.

**xlims = xlims,** # The limits of the x axis in the figure is equal to the x lims specified above.

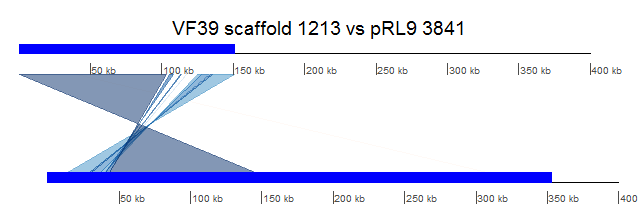
**main="VF39 scaffold 1213 vs pRL9 3841",** # Figure title

**gene\_type = "side\_blocks",** #gene\_type can be a block, a side\_block, or a triangle grob, or arrow.

**dna\_seg\_scale = TRUE, scale = FALSE)** # the scale is on the gene segment rather than a scale of base pair size at the bottom of the figure.

By specifying “comparisons in the above code, the comparisons found by BLAST on the NCBI website are plotted between the two DNA segments in shades of grey and blue.

**Which results in the following graphical comparison output:**

**Figure 3 Comparison of sequence of *R. leguminosarum* bv. *viciae* strain VF39SM c plasmid scaffold 1213 to the entire plasmid pRL9 sequence from *R. leguminosarum* bv. *viciae* strain 3841.**

With the grey color representing the most similar region and light blue representing regions of lower similarity. The dark blue lines between the two aligned genomic sequences represent small regions of sequence similarity between the two genomic sequences in this figure.

**Plotting of individual genes of VF39 c and d scaffolds**

In order to graphically represent the catabolic genes and their arrangement on the c and d plasmid scaffolds of *R. leguminosarum* bv. *viciae* strain VF39SM, the following information must be known: Gene names or symbols, the coordinates at which each gene starts and ends at, orientation of the gene fragment on the strand, either forward (+1) or reverse (-1), and color for each fragment in the order of the gene names given. This information is then used to build a data frame using the “data.frame” command, which was then subsequently designated as a dan\_seg containing object. The dna\_seg objects were then plotted using the “plot\_gene\_map” function. The following code was used to generate a plot of the scaffold genes:

Which resulted in the following output plot

**df1 <- data.frame(name=c("RlcK","RlcD","RlcB","RlcC","RlcH","RlcA","RlcO","RlcE","RlcF"),** #Gene symbols for each gene in this dna\_seg object

**start=c(40333, 116842, 124822, 135571, 127430, 133945, 113133, 105466, 292),** #The start coordinates of each corresponding gene in the dna\_seg object

**end=c(40373, 117376, 127423, 150883, 133347, 135513, 113649, 107870, 102982),** #The end coordinates of each corresponding gene in the dna\_seg object

**strand=c(1, -1, -1, -1, -1, -1, -1, -1, -1),** #The orientation or direction of each gene on the dna\_seg object.

**col=c("red", "blue", "yellow", "green", "black", "purple", "white", "pink", "orange"))** #color of each gene in the dna\_seg object.

**dna\_seg1 <- dna\_seg(df1)** #designating dna\_seg1 as containing information in data frame 1

**df1** #print contents of data frame 1

**dna\_seg1** #print contents of dna\_seg1

**df2 <- data.frame(name=c("RBTCK", "RBTCD", "RBTCB", "RBTCC", "RBTCH", "RBTCA", "RBTCO", "RBTCE", "RBTCF"),** #Gene symbols for each gene in this dna\_seg object

**start=c(307792, 40158, 39595, 29390, 36883, 30955, 40673, 43075, 145781),** #The start coordinates of each corresponding gene in the dna\_seg object

**end=c(307832, 39626, 36994, 36994, 14075, 30955, 29392, 40158, 40669),** #The end coordinates of each corresponding gene in the dna\_seg object

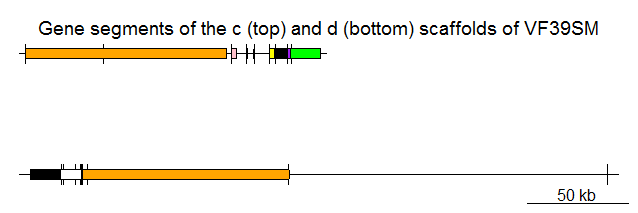
**strand=c(1, -1, -1, -1, -1, -1, -1, -1, -1),** #The orientation or direction of each gene on the dna\_seg object.

**col=c("red", "blue", "yellow", "green", "black", "purple", "white", "pink", "orange"))** #color of each gene in the dna\_seg object.

**dna\_seg2 <- dna\_seg(df2)** #designating dna\_seg2 as containing information in data frame 2

**plot\_gene\_map(dna\_segs=list(dna\_seg1, dna\_seg2),** #plot a gene map of dna\_seg1 and 2

**main = "Gene segments of the c (top) and d (bottom) scaffolds of VF39SM")** #With this title



**Figure 4. Catabolic genes on the c and d plasmid scaffolds of *R. leguminosarum* bv. *viciae* strain VF39SM**

**Annotation of gene segments**

It is also necessary to label genes with a gene symbol or name within a larger DNA segment in order to compare your genome of study with those of other microorganisms in order to establish evolutionary relationships by analysis of genes or gene clusters present in the genome of each microorganism. The following basic code example demonstrates the steps necessary to annotate genes in a data frame.

**Genes in a larger DNA segment-VF39SM c scaffold sequence genes and annotation of these genes with gene symbols**

To mark off specific genes in a large DNA segment, information about the gene names, start and end sites, strand direction either "+" (sense) or "-" (anti-sense) and colors for each gene segment mapped should be known.

Data frames are made with the gene information, and then designated as a dna\_seg object, which can then be plotted using the plot\_gene\_map function.

**name1 <- c("RlcK","RlcD","RlcB","RlcC","RlcH","RlcA","RlcO","RlcE","RlcF")**#Gene symbols for each gene in this dna\_seg object

**start1 <- c(40333, 116842, 124822, 135571, 127430, 133945, 113133, 105466, 292)**#The start coordinates of each corresponding gene in the dna\_seg object

**end1 <- c(40373, 117376, 127423, 150883, 133347, 135513, 113649, 107870, 102982)**#The end coordinates of each corresponding gene in the dna\_seg object

**strand1=c(1, -1, -1, -1, -1, -1, -1, -1, -1)**#The orientation or direction of each gene on the dna\_seg object.

**col1 <- c("red", "blue", "yellow", "green", "black", "purple", "white", "pink", "orange")**#color of each gene in the dna\_seg object.

**df1a <- data.frame(name=name1, start=start1, end=end1, strand=strand1, col=col1)**

**dna\_seg1a <- dna\_seg(df1a)** #designating dna\_seg1a as containing information in data frame 1a

**is.dna\_seg(dna\_seg1a)**

**df1a** #print contents of data frame 1a

**dna\_seg1a** #print contents of dna\_seg1a

#annotation of dna\_seg 1a

**mid <- middle(dna\_seg1a)** # mid calculates the middle of genes in dna\_seg1.

**annot1 <- annotation(x1=mid, text=dna\_seg1a$name, rot = 0, col = "black")** #Places text as an annotation from the name vector in data frame1a in the middle of genes on dna\_seg1a. The text is not rotated to any angle and the font color is black.

#Plotting DNA seg 1a

**plot\_gene\_map(dna\_segs = list(dna\_seg1a),** # Plot genes in dna\_seg 1a

**annotations= annot1, annotation\_height=1.0, annotation\_cex=0.3,** annotating genes in the middle, with a height of 1.0 and annotation text or symbol size of 0.3.

**limit\_to\_longest\_dna\_seg = F**, # Do not limit annotation to the longest dna\_seg.

**dna\_seg\_scale = TRUE)** # Add a dna scale to the dna\_seg.

The code resulted in the following graphical output: (click on pdf link below to view output)



**Figure 6 Annotated VF39SM c plasmid scaffold sequence**

**Simple annotation example adapted from the three genes dataset in genoPlotR package**

**names3 <- c("gene1", "gene2", "gene3")** #Vector of names for genes on the first DNA segment in this DNA segment comparison

**starts1 <- c(1, 1200, 1700)** #The start sites of genes 1-3 on DNA segment 1

**ends1 <- c(500, 1500, 2500)** # The end sites of genes 1-3 on DNA segment 1

**strands1 <- c("-", -1, 1)** #On DNA segment one, genes one and two go in the reverse orientation, and gene three is in the forward, sense orientation.

**cols1 <- c("red", "grey", "blue")** # Vector denoting the colors of genes 1, 2 and 3 on dna segment 1.

**df4 <- data.frame(name=names3, start=starts1, end=ends1,** #Create a dataframe named data frame 4 with vectors "names3", "starts1", "ends1", "strands1", "cols1".

**strand=strands1, col=cols1)**

**dna\_seg1 <- dna\_seg(df4)** #dna\_seg1 is equal to the dna\_seg specified in data frame 4.

**is.dna\_seg(dna\_seg1)** #Asks R if dna\_seg1 is interpreted by R as a dna\_seg. Returns a logical "True" or "False" statement.

**str(dna\_seg1)** #Asks R what the structure of dna\_seg1 is.

**#second dna seg**

**names2 <- c("gene4", "gene5", "gene6")** #Vector of gene names for dna\_seg2.

**starts2 <- c(50, 1400, 2000)** #Vector specifying the start sites of genes 4-6 on dna\_seg2

**ends2 <- c(600, 1700, 3000)** #Vector specifying the end sites of genes 4-6 on dna\_seg2.

**strands2 <- c("+", 1, -1)** #Vector specifying the orientation of genes 4-6 on dna\_seg2. Genes 4 and 5 are transcribed or oriented on dna\_seg2 in the positive or sense direction, while gene 6 is oriented in the negative direction.

**cols2 <- c("grey", "yellow", "blue")** #Vector specifying the colors of genes 4-6 on dna\_seg2. Gene 4 is colored grey, gene 5, yellow and gene 6 is blue in color.

**df5 <- data.frame(name=names2, start=starts2, end=ends2,** **strand=strands2, col=cols2)** #Create data frame 5 with vectors names 2, starts 2, ends 2, strands 2 and cols 2.

**dna\_seg2 <- dna\_seg(df5)** #dna\_seg2 is composed of the dna\_seg specified by data frame 5.

**is.dna\_seg(dna\_seg2)** #Asks R if dna\_seg2 is a dna\_seg. Returns a logical "True" or "False"

**str(dna\_seg2)** #Asks R what the structure of dna\_seg2 is.

**#Comaparison of dna\_segs1 and 2**

**comparison1 <- as.comparison(data.frame(start1=starts1, end1=ends1, start2=starts2, end2=ends2))** #Create a comparison between the start and end sites of genes 1-3 on dna\_seg1 to those of genes 4-6 on dna\_seg2.

**str(comparison1)** #Asks R what the structure of comparison 1 is.

**#annotation of dna\_segs 1 and 2**

**mid <- middle(dna\_seg1)** # mid calculates the middle of genes 1-3 in dna\_seg1.

**mid2 <- middle(dna\_seg2)**# mid2 calculates the middle of genes 4-6 in dna\_seg2.

**annot1 <- annotation(x1=mid, text=dna\_seg1$name, rot = 0, col = "black")** #Places text as an annotation from the name vector in data frame1 in the middle of genes 1-3 on dna\_seg1. The text is not rotated to any angle and the font color is black.

**annots <- annotation(x1=mid2, text=dna\_seg2$name, rot = 0, col = "black")** #Places text as an annotation from the name vector in data frame2 in the middle of genes 4-6 on dna\_seg2. The text is not rotated to any angle and the font color is black.

**#Plotting DNA segs 1 and 2**

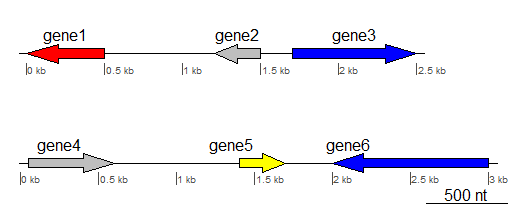
**plot\_gene\_map(dna\_segs = list(dna\_seg1, dna\_seg2),**

**annotations= list(annot1, annots), annotation\_height=1.8, annotation\_cex=1,** # Plot genes in dna\_seg 1 and 2, annotating genes 1-3 and 4-6 in the middle, with a height of 1.8 and annotation text or symbol size of 1.

**limit\_to\_longest\_dna\_seg = F,** # Do not limit annotation to the longest dna\_seg.

**dna\_seg\_scale = TRUE)** # Add a dna scale to the bottom right hand side of the gene map plot.

Basically, data frames containing information on each gene within a group of genes or cluster of genes is converted into a dna\_seg object, which are then compared for plotting purposes, annotated and then plotted using the “plot\_gene\_map function” to result in the following graphical representation of the two genomes



**Figure 7. Example of annotation of genes in two DNA segments**

**Genome level comparison: Comparison of four *Bartonella* genomes**

A more complex example is given below, which combines genome plotting with gene annotation and degree of genome relationship by gene sequence comparison and tree plotting of the four genomes being compared. This example also incorporates the use of a function which places the annotation in the middle of each gene, but does not label genes starting with the letter “B”, and also only labels every fourth gene.

**data(barto)** # Sample data set containing comparisons of the genomes of 4 Bartonella strains from the genoPlotR package.

**tree <- newick2phylog("(BB:3.0,(BG:1.0,(BH:0.5,BQ:0.2):2):3.5);")** # plots a phylogenetic tree of the four Bartonella genomes being compared. The decimals next to the abbreviations for the Bartonella genomes being compared indicate the lengths of the branches from the node that they are derived from. BH and BQ are 2.0 units away from the parental node and BB and BG are 3.5 units away from the parental node.

**xlims2 <- list(c(1445000, 1415000, 1380000, 1412000), c( 10000, 45000, 50000, 83000, 90000, 120000), c( 15000, 36000, 90000, 120000, 74000, 98000), c( 5000, 82000))** #These are the x axis limits for each gene in a dna segment being compared on the plot.

**annots <- lapply(barto$dna\_segs,** #lapply attaches an annotation to each DNA segment

**function(x)** #A funtion for applying annotations to function specified gene segments.

**{mid <- middle(x)** # defines mid as the middle of the gene segment

**annot <- annotation(x1=mid, text=x$name, rot=30)** #indicates to annotate in the middle of the gene with a character from the name column, with the annotation being rotated at a 30 degree angle.

**idx <- grep("^[^B]", annot$text, perl=TRUE)** #The vector idx contains gene names which match the pattern of not starting with the letter B in the text column, and are Perl-style regular expressions.

**annot[idx[idx %% 4 == 0],] })** # Annotate every fourth gene on the dna segment which does not start with the letter B.

**plot\_gene\_map(barto$dna\_segs, barto$comparisons, tree=tree,**  #Plot a map of the DNA segments of the comparisons of the four Bartonella genomes in Newick Tree format, as specified above,

**annotations=annots, xlims=xlims2, limit\_to\_longest\_dna\_seg=FALSE, dna\_seg\_scale=TRUE)**# annotating every fourth gene in the middle that does not start with the letter B, with x axis limits of each gene subsegment as specified by xlims2. Apply a dna\_seg scale to each dna segment on the plot.

**Click on the PDF link below to view graphical output of Figure 8 from the above code.**

****

**References**

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410. National Center of Biotechnology Information (NCBI) website URL https://www. www.ncbi.nlm.nih.gov (Source of VF39SM and 3841 strain genomic DNA sequences BLAST alignment data.)

Dray, S. and Dufour, A.B. and Chessel, D. (2007): The ade4 package-II: Two-table and K-table methods. RNews. 7(2): 47-52.

Guy, L. (2015). Getting started with genoPlotR (vignette). Pages 1-17.

Guy, L., Kultima, J.R., Andersson, G.E., (2010). genoPlotR: comparative gene and genome visualization in R. (Source of the sample “Barto” data set for the four genome comparison, along with use of this package for analysis and plotting of *R. leguminosarum* bv. *viciae* strains VF39SM and 3841 genomic DNA sequences and comparisons.) Bioinformatics 26 (18): 2334-2335.

The Genome Portal of the Department of Energy Joint Genome Institute

Grigoriev IV, Nordberg H, Shabalov I, Aerts A, Cantor M, Goodstein D, Kuo A, Minovitsky S, Nikitin R, Ohm RA, Otillar R, Poliakov A, Ratnere I, Riley R, Smirnova T, Rokhsar D, Dubchak I.

Nucleic Acids Res. 2012 Jan;40(Database issue):D26-32 (Source of *R. leguminosarum* bv. *viciae* strains VF39SM and 3841 genomic DNA sequences)

Madigan, M.T., Martinko, J.M., Bender, K.S., Buckley, D.H., Stahl, D.A., (2015). Biology of Microorganisms, Fourteenth Ed. Benjamin, Cummings Publishing. Unit 3, Chapter 12 Microbial Evolution and Systematics pages 355-361.

Olsen, G., (2016). Gary Olsen's Interpretation of the "Newick's 8:45" Tree Format Standard Example of Newick Tree Format. Accessed April 10, 2016. <http://evolution.genetics.washington.edu/phylip/newick_doc.html>

R Core Team (2015). R: A language and environment for statistical computing.

R Foundation for Statistical Computing, Vienna, Austria.

URL https://www.R-project.org/.

RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA

URL http://www.rstudio.com/.

European Molecular Biology Laboratory (EMBL) website http://www.embl.org/