Riparian Plot Guide

JTLovell

10/25/2021

1. Run GENESPACE

See the GENESPACE overview for pipeline details. The full run through syntenic block construction is here:

```r
if (!requireNamespace("devtools", quietly = TRUE))
    install.packages("devtools")
if (!requireNamespace("GENESPACE", quietly = TRUE))
    devtools::install_github("jtlovell/GENESPACE", upgrade = F)


if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
if (!requireNamespace("Biostrings", quietly = TRUE))
  BiocManager::install("Biostrings")
if (!requireNamespace("rtracklayer", quietly = TRUE))
   BiocManager::install("rtracklayer")


library(GENESPACE)
runwd <- file.path("~/Desktop/testGenespace")

make_exampleDataDir(writeDir = runwd)
```

```
FALSE Done! Run GENESPACE with rawGenomeRepo = ~/Desktop/testGenespace/rawGenomes
```

```r
gpar <- init_genespace(
  genomeIDs = c("human","chimp","rhesus"),
  speciesIDs = c("human","chimp","rhesus"),
  versionIDs = c("human","chimp","rhesus"),
  ploidy = rep(1,3),
  diamondMode = "fast",
  orthofinderMethod = "fast",
  wd = runwd,
  nCores = 4,
  minPepLen = 50,
  gffString = "gff",
  pepString = "pep",
  path2orthofinder = "orthofinder",
  path2mcscanx = "~/MCScanX",
  rawGenomeDir = file.path(runwd, "rawGenomes"))
```

```
FALSE set working directory to /Users/jlovell/Desktop/testGenespace
FALSE
```

```
FALSE found raw gff files:
FALSE    /Users/jlovell/Desktop/testGenespace/rawGenomes/human/human/annotation/human_GRCh38.p13_gene.g
FALSE   /Users/jlovell/Desktop/testGenespace/rawGenomes/chimp/chimp/annotation/chimp_Clint_PTRv2_gene.g
FALSE   /Users/jlovell/Desktop/testGenespace/rawGenomes/rhesus/rhesus/annotation/rhesus_Mmul_10_gene.gff
FALSE
FALSE found raw peptide files:
FALSE    /Users/jlovell/Desktop/testGenespace/rawGenomes/human/human/annotation/human_GRCh38.p13_pep.fa
FALSE   /Users/jlovell/Desktop/testGenespace/rawGenomes/chimp/chimp/annotation/chimp_Clint_PTRv2_pep.fa
FALSE   /Users/jlovell/Desktop/testGenespace/rawGenomes/rhesus/rhesus/annotation/rhesus_Mmul_10_pep.fa.g
FALSE
FALSE
FALSE Can't find all parsed annotation files ... need to run parse_annotations, parse_ncbi or parse_phy
FALSE
FALSE GENESPACE run initialized:
FALSE   Initial orthofinder database generation method: fast
FALSE   Orthology graph method: global
```

```
parse_annotations(
  gsParam = gpar,
  gffEntryType = "gene",
  gffIdColumn = "locus",
  gffStripText = "locus=",
  headerEntryIndex = 1,
  headerSep = " ",
  headerStripText = "locus=")
```

```
FALSE Parsing annotation files ...
FALSE   human ...
FALSE       Importing gff ... found 1787 gff entires, and 1787 gene entries
FALSE       Importing fasta ... found 1787 fasta entires
FALSE       1786 gff-peptide matches
FALSE   Done!
FALSE   chimp ...
FALSE       Importing gff ... found 1927 gff entires, and 1927 gene entries
FALSE       Importing fasta ... found 1927 fasta entires
FALSE       1926 gff-peptide matches
FALSE   Done!
FALSE   rhesus ...
FALSE       Importing gff ... found 1906 gff entires, and 1906 gene entries
FALSE       Importing fasta ... found 1906 fasta entires
FALSE       1904 gff-peptide matches
FALSE   Done!
```

```
gpar <- run_orthofinder(
  gsParam = gpar)
```

```
FALSE Synteny Parameters have not been set! Setting to defaults
FALSE   Running 'draft' a.k.a 'fast' genespace orthofinder method
FALSE   ############################################################
FALSE   ***NOTE***
FALSE   This method should only be used for:
FALSE       (1) closely related diploid species,
FALSE       (2) visualization/genome QC purposes, or
FALSE       (3) inferring orthogroups WITHIN syntenic regions
FALSE   ############################################################
FALSE       Running 1/6 (chimp vs. chimp)
FALSE       Running 2/6 (chimp vs. rhesus)
FALSE       Running 3/6 (rhesus vs. rhesus)
FALSE       Running 4/6 (chimp vs. human)
FALSE       Running 5/6 (rhesus vs. human)
FALSE       Running 6/6 (human vs. human)
FALSE       Done!
FALSE   Inverting intergenomic files ... Done!
FALSE   Running orthofinder -og on pre-computed blast:
```

```
gpar <- synteny(gsParam = gpar)
```

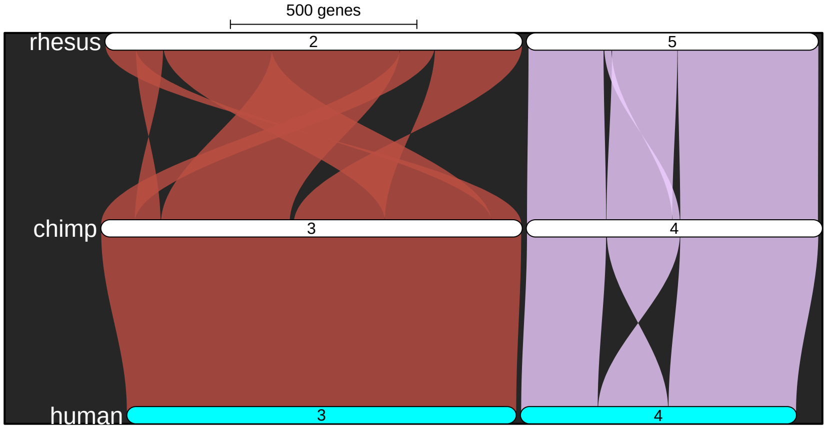
```
FALSE Loading annotations ...
FALSE   Indexing location of orthofinder results ... Done!
FALSE   Reading the gffs ... Done!
FALSE   Pulling gene lengths ... Done!
FALSE   Parsing global orthogroups ... Done!
FALSE Defining collinear orthogroup arrays ...
FALSE   Using collinear orthogroups for array identity:
FALSE   chimp: 102 genes in 37 collinear arrays
FALSE   human: 102 genes in 33 collinear arrays
FALSE   rhesus: 97 genes in 37 collinear arrays
FALSE   Choosing array representative genes ... Done!
FALSE Found 5616 genes, 1955 orthogroups and 107 arrays with 301 genes
FALSE Pulling within-genome synteny ...
FALSE   Genome: n raw hits / hits in (regions) / hits in (blks)
FALSE   human   (selfhit): 3834 / 3218 (2) / 3218 (2)

FALSE   chimp   (selfhit): 3834 / 3188 (2) / 3188 (2)

FALSE   rhesus  (selfhit): 3700 / 3056 (2) / 3056 (2)

FALSE Pulling intergenomic synteny ...
FALSE   human  -chimp   (primary): 4282 / 2771 (4) / 2761 (5)

FALSE   human  -rhesus  (primary): 4195 / 2624 (9) / 2613 (12)

FALSE   chimp  -rhesus  (primary): 4402 / 2712 (10) / 2695 (14)

FALSE   Synteny constraints - Done!
FALSE   Syntenic block coordinates written to /results/syntenicBlocks.txt.gz
FALSE Checking synteny-constrained global orthogroups for synOGs
FALSE   n. global OGs = 1955
FALSE   n. syntenic OGs = 2002
FALSE Combining synteny-constrained and inblock orthogroups ...
FALSE   syn OGs: 2002, inblk OGs: 0, combined OGs: 2002
FALSE   Wrote gff to file: /results/gffWithOgs.txt.gz
FALSE   Done!
```

2. Basic riparian functionality

2.1 Default specification Color chromosomes by the reference genome (and highlight the reference chromosomes in light yellow). Return the source data to construct the plots (for publications etc.). If you want to save the figure to file, see `?pdf`, `?png` or other graphic device writing functions.

```
ripSourceData <- plot_riparian(
  gpar,
  returnSourceData = T,
  highlightRef = "cyan")
```

```
##  Loading the gff ... Done!
##  Mapping genes against human chromosomes ... Done!
##  Projecting linear coordinate system ... Done!
##  Generating block coordinates ... Done!
##  Rendering plot ...
```

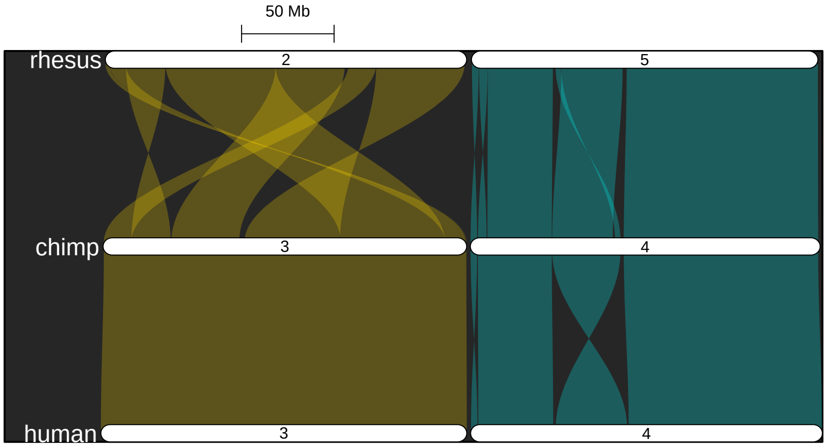


Chromosomes scaled by gene rank order

```
## Done!
```

2.2 change the input data Use physical positioninstead of gene rank order (`useOrder = F`) and plot the small syntenic blocks instead of the large syntenic regions (`plotRegions = F`). Also illustrating here that you can manually specify the colors (`colByChrscolByChrs`) and transparency (`braidAlpha`) of the syntenic polygons. If the number of colors specified is the same as the number of reference chrs, the colors will be respected exactly. Otherwise, the colors are used to make a ramp palette.

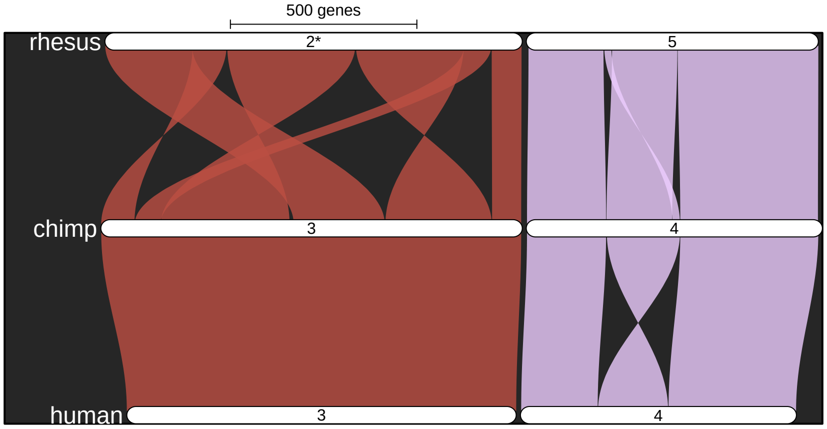
```
plot_riparian(
  gpar,
  plotRegions = F,
  useOrder = F,
  colByChrs = c("gold", "cyan"),
  braidAlpha = .25,
  verbose = F)
```



Chromosomes scaled by physical position

2.3 Invert a chromosome that is more syntenic if flipped Often some chromosomes are more syntenic if flipped. Here, we can specify which chromosomes to flip. The chromosome ID is then flagged with an "*".

```
invertThisGenomeChr <- data.table(genome = "rhesus", chr = "2")
plot_riparian(
  gpar,
  invertTheseChrs = invertThisGenomeChr,
  verbose = F)
```



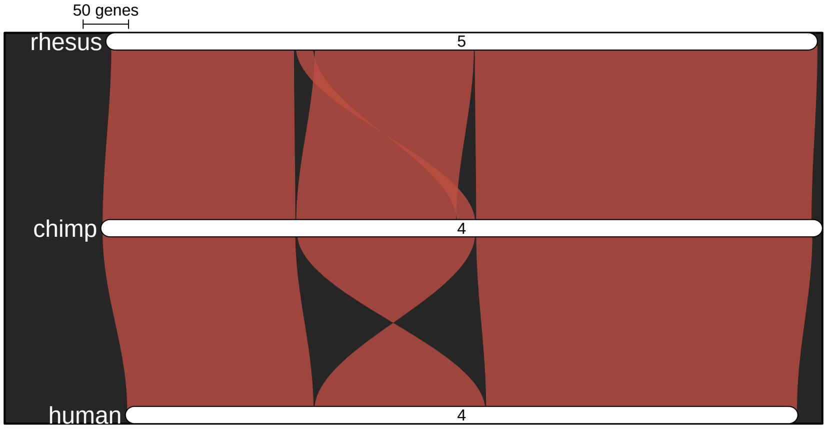
Chromosomes scaled by gene rank order

3. Using riparian plots to highlight specific regions

3.1 only plot specific chromosomes Often, the user wants to study a specific region, we can use riparian to just zoom in on specific chromosome. In this case, the chromosome has to be in the reference genome ... “human”, here.

```
plot_riparian(
  gpar,
  onlyTheseChrs = "4")
```

```
##  Loading the gff ... Done!
##  Mapping genes against human chromosomes ... Done!
##  Projecting linear coordinate system ... Done!
##  Generating block coordinates ... Done!
##  Rendering plot ...
```

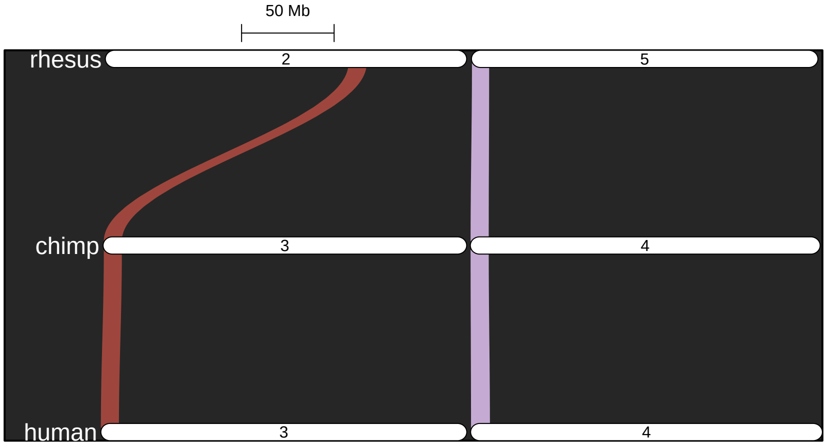


Chromosomes scaled by gene rank order

```
## Done!
```

3.2 only plot specific regions We can also zoom into just a couple regions. Here, the regions are specified by the physical position (bp), so we are best served setting useOrder = F. This is a nice way to combine multiple plots with multiple colors . . . make a full plot, then a separate plot for each region and overlay them in a vector graphics editor.

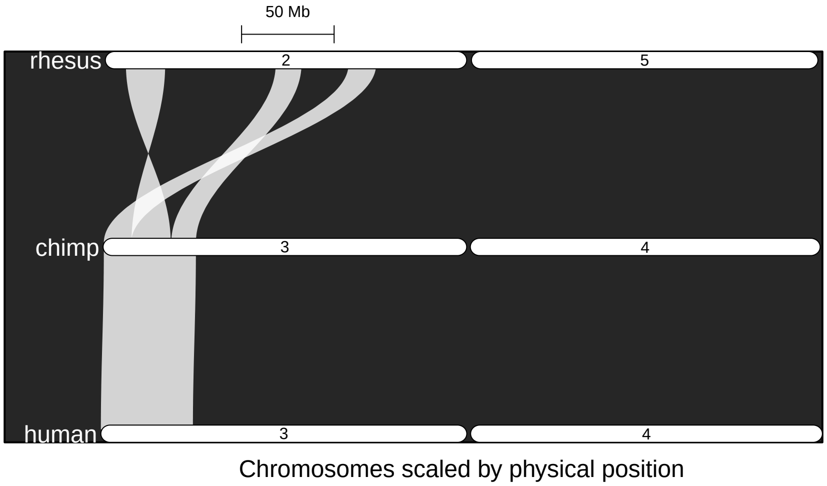
```
regs <- data.table(
  genome = c("human", "rhesus"),
  chr = c(3, 5),
  start = c(0, 0),
  end = c(1e7, 1e7))
plot_riparian(
  gpar,
  useOrder = F,
  onlyTheseRegions = regs,
  verbose = F)
```



Chromosomes scaled by physical position

we can also zoom in on just the highlighted regions.

```
regs <- data.table(
  genome = c("human"),
  chr = c(3),
  start = c(0),
  end = c(5e7))
plot_riparian(
  gpar,
  useOrder = F,
  onlyTheseRegions = regs,
  excludeChrOutOfRegion = F,
  colByChrs = "white",
  verbose = F)
```

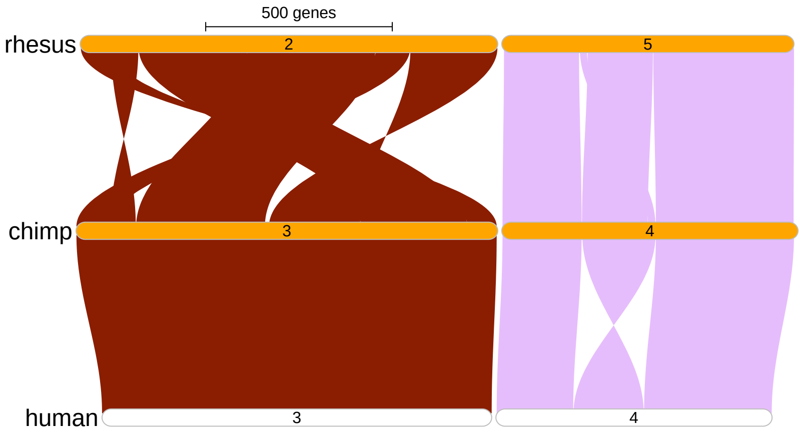


4 Adjust the general appearence

4.1 change the background and chr colors

Use a white background, orange chrs and completely opaque braids.

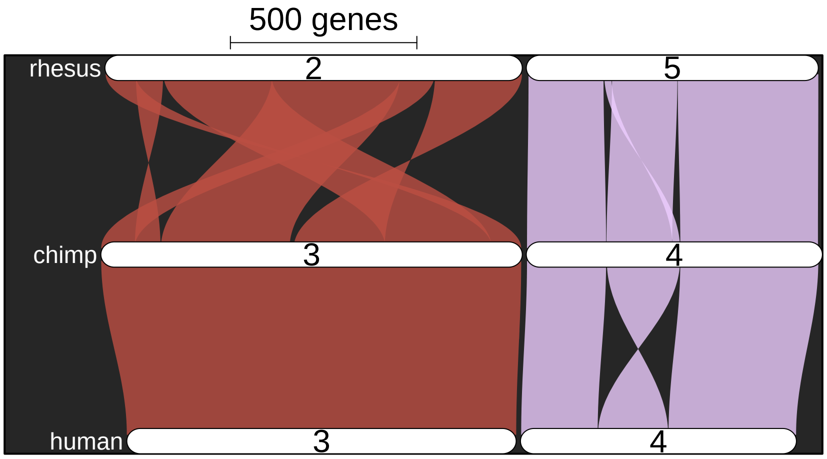
```
plot_riparian(
  gpar,
  blackBg = FALSE,
  chrFill = "orange",
  chrBorder = "grey",
  braidAlpha = 1,
  verbose = F)
```



Chromosomes scaled by gene rank order

4.2 change the size of chrs Increase the size of the chr links, but decrease the buffer size around the chrs

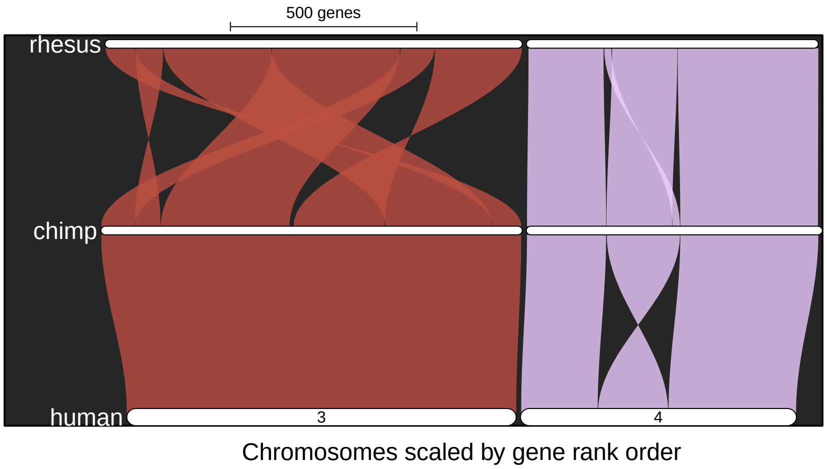
```
plot_riparian(
  gpar,
  chrLabCex = 1,
  chrRectBuffer = 1.1,
  verbose = F)
```



Chromosomes scaled by gene rank order

4.3 only label one genome In some cases, where there are lots of genomes or several genomes have tons of chromosomes, we might want to just label one or two genome chrs

```
plot_riparian(
  gpar,
  labelTheseGenomes = "human",
  verbose = F)
```



There is more functionality, but this is most of it.