Package 'popdemog'

September 20, 2017

Type Package

Title Plot population demographic hisotry

Version 1.0

Date 2017-06-29

Author Ying Zhou

Maintainer Ying Zhou <yz001@uw.edu>

Description Plot demographic graph for single/multiple populations based on the demographic description from scripts of simulation tools.

License GPL-2

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`popdemog-package` *Plot population demographic hisotry*

Description

Plot demographic graph for single/multiple populations based on the demographic description from scripts of simulation tools.

Details

This package contains three functions to visualize the population demographic history from simulation commands. Currently, `ms`, `msa`, `msHot`, `MaCS`, `msprime`, and `cosi` are supported. Users could copy the simulation script (for `ms(a)`, `msHot`, `MaCS`, and `msprime`(need a translation)) into one file or just use the parameter file (for `Cosi`) as the input command file. Each input file only contains one demographic model.

Once the command file is prepared, function `plotms` could be used to generate the demographic graph or to give the parameters to `plotmmig` plot the multiple migrations. More adjustments on migration graph would be achieved by using function `plotmig`, which is allowed to add the migration patterns to other plots, such as geological map.

Author(s)

Ying Zhou

Maintainer: Ying Zhou <yz001@uw.edu>

References

ms: Hudson, R. R. "Generating Samples under a Wright-Fisher Neutral Model of Genetic Variation." Bioinformatics 18.2 (2002): 337-38.

msHot: Hellenthal, G., and M. Stephens. "MsHOT: Modifying Hudson's Ms Simulator to Incorporate Crossover and Gene Conversion Hotspots." Bioinformatics 23.4 (2006): 520-21.

MaCS: Chen, G. K., P. Marjoram, and J. D. Wall. "Fast and Flexible Simulation of DNA Sequence Data." Genome Research 19.1 (2008): 136-42.

Cosi: Shlyakhter, Ilya, Pardis C. Sabeti, and Stephen F. Schaffner. "Cosi2: An Efficient Simulator of Exact and Approximate Coalescent with Selection | Bioinformatics | Oxford Academic." OUP Academic. Oxford University Press, 22 Aug. 2014.

msprime: Jerome Kelleher, Alison M Etheridge and Gilean McVean, "Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes", PLoS Comput Biol 12(5): e1004842. doi: 10.1371/journal.pcbi.1004842. 2016.

Examples

```
###Tennessen's standard model
cat("macs 2025 15000000 -i 10 -r 3.0e-04 -t 0.00069 -T -I 4 10 1006 1008 1 0
-n 4 0.205 -n 1 58.00274 -n 2 70.041 -n 3 187.55 -eg 0.9e-10 1 482.46
-eg 1.0e-10 2 570.18 -eg 1.1e-10 3 720.23 -em 1.2e-10 1 2 0.731
-em 1.3e-10 2 1 0.731 -em 1.4e-10 3 1 0.2281 -em 1.5e-10 1 3 0.2281
-em 1.6e-10 2 3 0.9094 -em 1.7e-10 3 2 0.9094 -eg 0.007 1 0
-en 0.007001 1 1.98 -eg 0.007002 2 89.7668 -eg 0.007003 3 113.3896
-eG 0.031456 0 -en 0.031457 2 0.1412 -en 0.031458 3 0.07579
-eM 0.031459 0 -ej 0.03146 3 2 -en 0.0314601 2 0.2546
-em 0.0314602 2 1 4.386 -em 0.0314603 1 2 4.386 -eM 0.0697669 0
-ej 0.069767 2 1 -en 0.0697671 1 1.98 -en 0.2025 1 1 -ej 0.9575923 4 1
-em 0.06765 2 4 32 -em 0.06840 2 4 0", file="std-model-Tennessen.cmd")
#plot the demographic graph
par(mfrow=c(1,2))
plotms(inputfile="std-model-Tennessen.cmd", type="macs", N4=10000, pop.scale="log",
```

```
log.base  =50, inpos = c(1,4,7,9), time.scale = "log10year",
col.pop=c("brown", "blue", "gold3", "forestgreen"),
pops=c("AFR", "EUR", "ASIA", "ARC"),
cex.lab=1, cex.axis = 1, xlab="", length.arrow=0.1)
title("Demographic histoy")
plotms(inputfile="std-model-Tennessen.cmd", type="macs", N4=10000,
time.scale = "log10year", plot=FALSE)->out;
plotmig(time_pt=2, demograph_out=out$demograph_out,mig_par=out$mig_par,
col.pop=c("brown", "blue", "gold3", "forestgreen"), pop.scale="topology");
legend("topleft", legend=c("AFR", "EUR", "ASIA", "ARC"),
col=c("brown", "blue", "gold3", "forestgreen"), pch=20, bty="n")
title("Migrations at 200 years ago");unlink("std-model-Tennessen.cmd")


###Archaic introgrssion model
cat("./ms 44 1 -r 20000 50000000 -t 30000 -I 6 20 20 1 1 1 1 -en 0 1 1
-en 0 2 1 -en 0 3 1e-10 -en 0 4 1e-10 -en 0 5 1e-10 -en 0 6 1e-10
-es 0.0125 2 0.97 -en 0.02500025 7 0.25 -en 0.02500025 2 1 -ej 0.05 4 3
-ej 0.05 6 5 -en 0.05000025 3 0.25 -en 0.05000025 5 0.25 -ej 0.0500025 5 3
-en 0.050005 3 0.25 -ej 0.075 2 1 -en 0.0750025 1 1 -ej 0.1 7 3
-en 0.1000025 3 0.25 -ej 0.3 3 1 -en 0.3000025 1 1", file="test.1.ms.cmd")
plotms(inputfile = "test.1.ms.cmd", type="ms", N4=10000,
time.scale = "kyear", length.arrow=0.1, inpos=c(1,2,5,4.5,5.5,6,3),
col.pop=c("brown", "blue", "forestgreen", rainbow(10)[6:9]));
unlink("test.1.ms.cmd")


###Migration model from ms.
cat("./ms 15 100 -t 3.0 -I 6 0 7 0 0 8 0 -m 1 2 2.5 -m 2 1 2.5 -m 2 3 2.5
-m 3 2 2.5 -m 4 5 2.5 -m 5 4 2.5 -m 5 6 2.5 -m 6 5 2.5 -em 2.0 3 4 2.5
-em 2.0 4 3 2.5", file="test.2.ms.cmd")
plotms(inputfile = "test.2.ms.cmd", type="ms", N4=10000, col.pop="gray",
col.arrow="black", length.arrow=0.1, lwd.arrow=2);unlink("test.2.ms.cmd")
```

`plotmig` *Plot single migration event*

Description

This function is used to plot single migration event for multiple populations based on the output of `plotms` with `plot`=FALSE. The parameters of `add` and `map.pos` allow the migration graph to be added to any other plots with specified positions for each population.

Usage

```
plotmig(time_pt = NULL, event = 1, mig_par, demograph_out,
  pop.scale = mig_par$pop.scale, linear.scale = mig_par$linear.scale,
  log.base = mig_par$log.base, col.pop = mig_par$col.pop,
  col.arrow = mig_par$col.arrow, xlim = mig_par$xlim, ylim = mig_par$ylim,
  lwd.arrow = mig_par$lwd.arrow, length.arrow = mig_par$length.arrow,
```

```
angle.arrow = mig_par$angle.arrow, topology.scale = 1, add = FALSE,
map.pos = NULL, m.adjust = 0)
```

Arguments

time_pt	Define the time to plot migrations. time_pt should be the same sacle as the setting of time.scale. For example, time_pt=3 when time.scale="log10year" gives us the migrations at 10^3 years ago.
event	Define index of an event in time order to be plotted. Every demographic event has an index, which is used to track the time when that event happens. Any demographic changes at the same time is defined as the same event.
mig_par	A list contained all settings for plot demographic graph, please check the return value decription of plotms.
demograph_out	A list contained all demographic information, please check the return value decription of plotms.
pop.scale	A string to define different ways to rescale population size, which is used for adjust the lineage width in the plot. "topology" ignores the population size so that only the topology structure will be presented and the time points will be as the integers of the orders of the demographic events; "linear" linearly rescales the population size, the scale magnitude is defined in variable linear.scale; "log" is to use logarithm to rescale the population size, the base is defined in log.base.
linear.scale	A numeric number used for rescale population size linearly. This value will be applied when pop.scale="linear".
log.base	A numeric value as the base of logarithm of the population size. This value will be applied when pop.scale="log".
col.pop	A specification for the default plotting color for each population.
col.arrow	A specification to define the color for each migration arrow.
xlim	Specify the range of x-axis.
ylim	Specify the range of y-axis.
lwd.arrow	The line width of arrow, which is determined by 0.5+migration strngth*lwd.arrow.
length.arrow	A positive number for the size of arrow ends
angle.arrow	The angle of the arrow end, between 0 and 90.
topology.scale	Used for rescale the size of circle when the pop.scale is "topology".
add	A logical value allowing to add the migration graph to other plot if it is TRUE and the positions for every population should be well define in variable map.pos.
map.pos	A npop*2 matrix of positions for each population.
m.adjust	Threshold for plotting the migration events. Only migration strength higher than m.adjust will be plotted. Default value is 0.
...	Please check the parameters' decription of plotms.

See Also

plotms

Examples

```
cat("./ms 15 100 -t 3.0 -I 6 0 7 0 0 8 0 -m 1 2 2.5 -m 2 1 2.5 -m 2 3 2.5
-m 3 2 2.5 -m 4 5 2.5 -m 5 4 2.5 -m 5 6 2.5 -m 6 5 2.5 -em 2.0 3 4 2.5
-em 2.0 4 3 2.5", file="test.mig.cmd")
out<-plotms(inputfile = "test.mig.cmd", type="ms", N4=10000, plot=FALSE);
#check all migration events
events<-out$mig_par$events
print(events)
#check the time for those migration events
timeofevents<-out$mig_par$time[events]
print(timeofevents)
#plot event by event
par(mfrow=c(1,2))
plotmig(event=1, demograph_out=out$demograph_out, mig_par=out$mig_par)
title("Event-1");
plotmig(event=2, demograph_out=out$demograph_out, mig_par=out$mig_par,
col.pop=1:6, xlim=c(-5,4))
title("Event-2", cex.main=3);
legend("topleft", col=1:6, pch=20, bty="n", cex=2,
legend=c("pop-1", "pop-2", "pop-3","pop-4","pop-5","pop-6"))
unlink("test.mig.cmd")
```

---

plotmmig *Plot mutiple migrations in one figure*

Description

This function is used to plot the migration history for multiple populations based on the output of plotms with plot=FALSE. All plot settings are in the function plotms. Please use function plotmig to customize each migration plot.

Usage

```
plotmmig(demograph_out, mig_par, m.adjust = 0)
```

Arguments

demograph_out	A list contained all demographic information, please check the return value decription of plotms.
mig_par	A list contained all settings for plotting demographic graph, please check the return value decription of plotms.
m.adjust	Threshold for plotting the migration events. Only migration strength higher than m.adjust will be shown. Default value is 0.

See Also

plotms

Examples

```
cat("./ms 15 100 -t 3.0 -I 6 0 7 0 0 8 0 -m 1 2 2.5 -m 2 1 2.5 -m 2 3 2.5
-m 3 2 2.5 -m 4 5 2.5 -m 5 4 2.5 -m 5 6 2.5 -m 6 5 2.5 -em 2.0 3 4 2.5
-em 2.0 4 3 2.5", file="test.mig.cmd")
out<-plotms(inputfile = "test.mig.cmd", type="ms", N4=10000, plot=FALSE,
col.pop=1:6, cex.lab=1.5);
plotmmig(out$demograph_out, out$mig_par)
unlink("test.mig.cmd")
```

---

plotms *Plot population demographic graph and generate plot parameters*

Description

This is the main function to plot demographic graph for single/multiple populations, which is named after Hudson's ms. Currently, it can read the simulation script from ms, msa, msHot, MaCS, and cosi.

The command inputfile and command type are required to give a quick plot of demographic history. Principally, every component in the output graph could be directly adjusted precisely.

In the demographic graph, each population has a lineage vertically stretch back in time (inpos for positions, col.pop for the colors, pops for the names, and time.scale for the time axis). The width of the lineage reflects the relative population size (pop.scale for lineage widths). Population splits and migrations are in arrows (col.arrow, length.arrow, lwd.arrow, angle.arrow).

Usage

```
plotms(inputfile, type, inpos = NULL, N4 = 10000, plot = T,
  pop.scale = "linear", linear.scale = 0.2, log.base = 10,
  time.scale = "4Ne", gen = 25, m.adjust = 0, col.pop = "gray45",
  col.arrow = col.pop, length.arrow = 0.15, lwd.arrow = 1,
  angle.arrow = 15, pops = NULL, xlab = "Population",
  ylab = paste("Time before present (", time.scale, ")", sep = ""),
  xlim = NULL, ylim = NULL, cex.lab = 1, cex.axis = 1, axes = T)
```

Arguments

inputfile	A file for demographic events. It can be simulation scripts (for ms, msa, MaCS) or parameter files (for cosi).
type	A string indicates the type of simlation command, currently supports "ms" for ms, "msa" for msa, "macs" for MaCS, and "cosi" for cosi.

inpos	A numeric vector of the population positions at time 0, used to avoid the intersections between the population lineages and migration arrows.
N4	A numeric for the effective population size of 4Ne, exact the same definition is the well-known simulation tool ms.
plot	A logical variable to give out the demographic plot or not. If TRUE, the demographic plot will be given; if FALSE, all parameters that used for the graph will output as a list.
pop.scale	A string to define different ways to rescale population size, which is used for adjust the lineage width in the plot. "topology" ignores the population size so that only the topology structure will be presented and the time will be as the integers of the orders of the demographic events; "linear" linearly rescales the population size, the scale magnitude is defined in variable linear.scale; "log" is to use logarithm to rescale the population size, the base is defined in log.base.
linear.scale	A numeric number used for rescale population size linearly. This value will be applied when pop.scale="linear".
log.base	A numeric value as the base of logarithm of the population size. This value will be applied when pop.scale="log".
time.scale	A string to define the time unit used in the plot, it can be "4Ne", "generation", "year", "kyear", and "log10year".
gen	A numeric number of years per generation. Default value is 25.
m.adjust	A numeric number to adjust the plot for migration events, only migration arrows with strength bigger than m.adjust are plotted. This value should be stay between 0 and 1.
col.pop	A specification for the default plotting color for each population.
col.arrow	A specification to define the color for each migration arrow
length.arrow	A positive number for the size of arrow ends
lwd.arrow	The line width of arrow, which is determined by 0.5+migration strngth*lwd.arrow.
angle.arrow	The angle of the arrow end, between 0 and 90.
pops	Population names in the plot. Usually as a vector of string, defaul as 1:total population number.
xlab	A title for the x-axis
ylab	A title for the y-axis
xlim	Specify the range of x-axis
ylim	Specify the range of y-axis
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex.
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex.
axes	A logical value to plot the axes or not.

Value

if the parameter plot=F/FALSE, following three lists will be returned:

demograph_out
: This list contains all demographic detaill from the input command file: time.series is a vector of time; Pos is a numeric matrix of positions for each population at every demographic event; N is a numeric matrix of population size for each population at every demographic event; m is 3-D numeric matrix of migration rates between populations at every demographic event; survive is a matrix recording the begin and end for each population according to the demographic events; g.rate is a matrix of exponential growth rates at every demographic event; pop.pos is a numeric vector of the population positions at time 0; pop.lab is a vector of population names; mscmd is the ms command for demographic plot. All simulation commands will turn to ms command for extract the demographic information; present.pop.num is the number of populations at present; total.pop.num is the number of total populations exist in the plot; N4 is 4Ne, please refers to Hudson's ms for more description; gen is the years per generation.

evo_par
: This list contains all parameters used to draw the demographic graph, including: pop.scale define the way to rescale population size; linear.scale is used for rescale population size linearly; log.base is the base of logarithm of the population size; time.scale defines the time unit used in the plot; time is the rescaled time vector according to the time.scale; col.pop specifies the plotting color for each population; col.arrow specifies the plotting color for each migration arrow; length.arrow is the size of arrow ends; lwd.arrow is the width of arrow; angle.arrow is the angle of the arrow end; lab.pop is a vector of population names; lab.pos is a vector of positions to plot the population names; xlim specifies the range of x-axis; ylim specifies the range of y-axis; xlab is the title of x-axis; ylab is the title of y-axis; cex.lab is the magnification to be used for x and y labels; cex.axis is the magnification to be used for axis annotation; axes is a logical value to plot the axes or not.

mig_par
: This list contains all parameters used to draw the migrations, including: pop.scale define the way to rescale population size; linear.scale is used for rescale population size linearly; log.base is the base of logarithm of the population size; time.scale defines the time unit used in the plot; time is the rescaled time vector according to the time.scale; lab.pop is a vector of population names; col.pop specifies the plotting color for each population; col.arrow specifies the plotting color for each migration arrow; length.arrow is the size of arrow ends; lwd.arrow is the width of arrow; angle.arrow is the angle of the arrow end; xlim specifies the range of x-axis; ylim specifies the range of y-axis; mfrow sets the layout of multiple migrations in one plot; events records the events when migration state changes; cex.lab is the magnification to be used for x and y labels;

References

4Ne: http://home.uchicago.edu/rhudson1/source/mksamples.html

Examples

```
#example 1
cat("./ms 44 1 -r 20000 50000000 -t 30000 -I 6 20 20 1 1 1 1 -en 0 1 1
-en 0 2 1 -en 0 3 1e-10 -en 0 4 1e-10 -en 0 5 1e-10 -en 0 6 1e-10
-es 0.0125 2 0.97 -en 0.02500025 7 0.25 -en 0.02500025 2 1 -ej 0.05 4 3
-ej 0.05 6 5 -en 0.05000025 3 0.25 -en 0.05000025 5 0.25 -ej 0.0500025 5 3
-en 0.050005 3 0.25 -ej 0.075 2 1 -en 0.0750025 1 1 -ej 0.1 7 3
-en 0.1000025 3 0.25 -ej 0.3 3 1 -en 0.3000025 1 1", file="test.1.ms.cmd")
plotms(inputfile = "test.1.ms.cmd", type="ms", N4=10000, time.scale = "kyear")
#adjust the population position
plotms(inputfile = "test.1.ms.cmd", type="ms", N4=10000, time.scale = "kyear",
inpos=c(1,2,5,4,6,7,3))
#add color for each population
plotms(inputfile = "test.1.ms.cmd", type="ms", N4=10000, time.scale = "kyear",
inpos=c(1,2,5,4,6,7,3), col.pop=rainbow(10)[3:9])
#add population names
unlink("test.1.ms.cmd")

#example 2
cat("./ms 15 100 -t 3.0 -I 6 0 7 0 0 8 0 -m 1 2 2.5 -m 2 1 2.5 -m 2 3 2.5
-m 3 2 2.5 -m 4 5 2.5 -m 5 4 2.5 -m 5 6 2.5 -m 6 5 2.5 -em 2.0 3 4 2.5
-em 2.0 4 3 2.5", file="test.2.ms.cmd")
plotms(inputfile = "test.2.ms.cmd", type="ms", N4=10000, col.pop=3,
col.arrow=1, lwd.arrow=1.5)
unlink("test.2.ms.cmd")

#example 3
cat("./ms 1 1 -t 1.0 -I 3 10 10 10 -n 1 1.682020 -n 2 3.736830
-n 3 7.292050 -eg 0 2 116.010723 -eg 0 3 160.246047
-ma 0 0.881098 0.561966 0.881098 0 2.797460 0.561966 2.797460 0
-ej 0.028985 3 2 -en 0.028985 2 0.287184
-ema 0.028985 3 0 7.293140 0 7.293140 0 0 0 0 0 -ej 0.197963 2 1
-en 0.303501 1 1", file="Ryan2009.cmd")
plotms(inputfile = "Ryan2009.cmd", type="ms", N4=10000, pop.scale="log",
log.base=200, pops=c("AFR", "EUR", "ESA"),
col.pop=c("brown", "blue", "yellow"));
unlink("Ryan2009.cmd")

#example 4
cat("ms 4 1 -t 7156.0000000 -r 2000.0000 10000000 -eN 0 5
-eG 0.000582262 1318.18 -eG 0.00232905 -329.546 -eG 0.00931619 82.3865
-eG 0.0372648 -20.5966 -eG 0.149059 5.14916 -eN 0.596236 0.5 -T",
file="zigzag.cmd")
par(mfrow=c(1,2))
plotms(inputfile = "zigzag.cmd", type="ms", N4=10000)
#change the time unit
plotms(inputfile = "zigzag.cmd", type="ms", N4=10000, time.scale="log10year")
unlink("zigzag.cmd")
```

Index