

# Package ‘POPdemog’

December 6, 2017

**Type** Package  
**Version** 1.0.3  
**Title** Plot Population Demographic History  
**Date** 2017-12-05  
**Author** Ying Zhou  
**Maintainer** Ying Zhou <yz001@uw.edu>  
**Description** Plot demographic graph for single/multiple populations from coalescent simulation program input.  
**License** GPL-2  
**RoxygenNote** 6.0.1  
**Suggests** knitr, maps, rmarkdown  
**VignetteBuilder** knitr  
**NeedsCompilation** no

## R topics documented:

POPdemog-package . . . . .	1
NRuler . . . . .	3
PlotMig . . . . .	4
PlotMMig . . . . .	6
PlotMS . . . . .	6
<b>Index</b>	<b>11</b>

---

POPdemog-package	<i>Plot Population Demographic History</i>
------------------	--

---

## Description

Plot demographic graph for single/multiple populations from coalescent simulation program input. Currently, this package does not check the simulation program input for correctness, but assumes the simulation program input has been validated by the simulation program.

## Details

This package contains functions to visualize the population demographic history from coalescent simulation program input. This package supports ms, msa, msHot, MaCS, msprime, cosi, and scrm simulation programs. Please check the [online tutorial file](#) for the current list of supporting programs.

The input of the program is a file or string containing the simulation program commands(for ms, msa, msHot, MaCS, and msprime(need a translation)), or the parameter file (for Cosi), or a conversion of the program commands into ms format(for msprime). The input file or string should only contain one demographic model.

Once the input file or string is prepared, the [PlotMS](#) function can be used to generate the plot of the demographic model or to generate parameters to pass to the [PlotMMig](#) function for displaying migrations. The visualization of migrations can be customized using [PlotMig](#) function.

## Author(s)

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(input.file = "std-model-Tennessen.cmd", type = "macs", N4 = 10000, size.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.arrowtip = 0.1)
```

```

title("Demographic histoy")
PlotMS(input.file = "std-model-Tenessen.cmd", type = "macs", N4 = 10000,
time.scale = "log10year", plot.out = FALSE, demo.out = TRUE)->out;
PlotMig(time_pt = 2, demograph_out = out$demograph_out,mig_par = out$mig_par,
col.pop = c("brown", "blue", "gold3", "forestgreen"), size.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-Tenessen.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(input.file = "test.1.ms.cmd", type = "ms", N4 = 10000,
time.scale = "kyear", length.arrowtip = 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(input.file = "test.2.ms.cmd", type = "ms", N4 = 10000, col.pop = "gray",
col.arrow = "black", length.arrowtip = 0.1, lwd.arrow = 2);unlink("test.2.ms.cmd")

```

NRuler

*Add population size ruler*

## Description

Adds a ruler for the population size. This function works like the function legend,

## Usage

```
NRuler(x, y = NULL, Nsize, Nlab = Nsize, N4, size.scale,
linear.scale = 0.2, log.base = 10, ...)
```

## Arguments

x, y	Position of the population size ruler, if y does not have numeric value, x will support the keywords inputs from the list "bottomright", "bottom", "bottom-left", "left", "topleft", "top", "topright", "right" and "center".
Nsize	The population sizes of the ticks on the ruler.
Nlab	The labels on the ticks of the ruler. Default labels are the population index.
N4	Scalar to scale real population size.
size.scale	A keyword to define the way population size is scaled and displayed as lineage width. It supports "log" and "linear".

<code>linear.scale</code>	The scale factor applied to the population size when <code>size.scale = "linear"</code> .
<code>log.base</code>	The base of logarithm used when <code>size.scale = "log"</code> .
<code>...</code>	Additional arguments can be passed, such as <code>col</code> , <code>lwd</code> , <code>lty</code> .

---

PlotMig

*Plot migration event(s) at a particular time*


---

## Description

This function plots migration events at a particular time point based on the output of `PlotMS` with `demo.out = T` and `plot.out = F`. The `time_pt` or `event` determines the time points that is plotted. The `add` and `map.pos` parameters allow the migration graph to be added to other backgrounds such as maps.

## Usage

```
PlotMig(time_pt = NULL, event = 1, mig_par, demograph_out,
        size.scale = mig_par$size.scale, time.scale = mig_par$time.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$lwd.arrow,
        length.arrowtip = mig_par$length.arrowtip,
        angle.arrowtip = mig_par$angle.arrowtip, toposize.scale = 1,
        add = FALSE, map.pos = NULL, m.adjust = 0)
```

## Arguments

<code>time_pt</code>	A numeric value defining the time point for the migrations which will be plotted. <code>time_pt</code> should be in the scale defined by <code>time.scale</code> . For example, <code>time_pt = 3</code> when <code>time.scale = "log10year"</code> corresponds to the migrations $10^3$ years ago.
<code>event</code>	An index to define at which time to plot migration(s). Every demographic event has an index in the order of time. All demographic changes at the same time all defined as the same event and share the same index.
<code>mig_par</code>	A list contained all settings for plotting demographic graph, see <a href="#">PlotMS</a> .
<code>demograph_out</code>	A list contained all demographic information, see <a href="#">PlotMS</a> .
<code>size.scale</code>	A keyword to define the scaling of lineage width. "topology" returns only topology structure among simulated populations, ignoring both the population sizes and the length of time between any demographic events. "linear" linearly scales the lineage widths as a function of the population size, the scale factor is defined by the variable <code>linear.scale</code> ; "log" scales the lineage width logarithmically as a function of the population size. The logarithm base is defined by the variable <code>log.base</code> .
<code>time.scale</code>	A keyword to define the time scale used in the plot, it can be "4Ne", "generation", "year", "kyear", and "log10year". When the <code>size.scale = "topology"</code> , this parameter will be ignored.
<code>linear.scale</code>	Linear scale magnitude, be applied when <code>size.scale = "linear"</code> .
<code>log.base</code>	The base of logarithm, be applied when <code>size.scale = "log"</code> .

<code>col.pop</code>	Population lineage color.
<code>col.arrow</code>	Migration arrow color.
<code>xlim</code>	The range of x-axis.
<code>ylim</code>	The range of y-axis.
<code>lwd.arrow</code>	Control the line width of arrow representing a migration. The arrow width is defined by $0.5 + \text{migration strength} * \text{lwd.arrow}$ .
<code>length.arrowtip</code>	Length of the edges of the arrow tip.
<code>angle.arrowtip</code>	The angle of the arrow tip, between 0 and 90.
<code>toposize.scale</code>	Control the scaling the size of circle when the <code>size.scale = "topology"</code> .
<code>add</code>	A logical value allowing to add the migrations to other background (2D only). Positions for every population dime should be well defined in <code>map.pos</code> .
<code>map.pos</code>	A matrix with two columns, the <i>i</i> th row is the coordinate for the <i>i</i> th population.
<code>m.adjust</code>	Migration threshold for plotting migration events. Migration events with strength higher than <code>m.adjust</code> will be shown. The migration strength is defined as the proportion of the target population being replaced by the source population per generation. 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(input.file = "test.mig.cmd", type = "ms",
N4 = 10000, plot.out = FALSE, demo.out = TRUE);
#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 Multiple Migrations*

---

**Description**

This function is used to plot all the migration events based on the output of `PlotMS` with `plot.out = FALSE` and `demo.out = TRUE`. Plot settings should be customized in [PlotMS](#). Use function [PlotMig](#) to customize the plot of single migration.

**Usage**

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

**Arguments**

<code>demograph_out</code>	A list of all demographic information. See the return value description of <a href="#">PlotMS</a> .
<code>mig_par</code>	A list of all settings for plotting demographic graph. See the return value description of <a href="#">PlotMS</a> .
<code>m.adjust</code>	Migration threshold for plotting migration events. Migration event with strength higher than <code>m.adjust</code> will be shown. The migration strength is defined as the proportion of the target population being replaced by the source population per generation. 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(input.file = "test.mig.cmd", type = "ms", N4 = 10000,
plot.out = FALSE, demo.out = TRUE, col.pop = 1:6, cex.lab = 1.5);
PlotMMig(demograph_out = out$demograph_out, mig_par = out$mig_par)
unlink("test.mig.cmd")
```

---

**PlotMS***Plot population demographic graph and generate demographic parameters*

---

**Description**

This is the main function to plot demographic graph for single/multiple populations. The function is named after Hudson's `ms` program. It can read the simulation input data used for the `ms`, `msa`, `msHot`, `MacS`, `scrm`, and `cosi` programs.

The `input.file` or `input.cmd` and command type are required to plot the demographic history. The output graph can be customized according to need.

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 population size (size.scale for lineage widths). Population splits and migrations are represented by arrows (col.arrow, length.arrowtip, lwd.arrow, angle.arrowtip).

### Usage

```
PlotMS(input.cmd = NULL, input.file = NULL, type, inpos = NULL,
       N4 = 10000, size.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.arrowtip = 0.15, lwd.arrow = 1,
       angle.arrowtip = 15, pops = NULL, xlab = "Population",
       ylab = paste("Time before present (", time.scale, ")", sep = ""),
       xlim = NULL, ylim = NULL, plot.out = T, demo.out = F, cex.lab = 1,
       cex.axis = 1, axes = T)
```

### Arguments

input.cmd	A string input containing program input commands.
input.file	A file containing simulation program input commands (for ms, msa, MaCS) or parameter files (for cosi).
type	A keyword indicates the type of simulation commands: "ms" for ms, "msa" for msa, "macs" for MaCS, "scrm" for scrm, "cosi" for cosi, "msprime" for msprime. Please check the <a href="#">online tutorial file</a> to see more support simulators.
inpos	Population positions in the plot at time 0.
N4	Four times the effective population size. This parameter has the same definition as the 4N0 parameter for the ms simulation program.
size.scale	A keyword to define the scaling of lineage width. "topology" returns only topology structure among simulated populations, ignoring both the population sizes and the length of time between any demographic events. "linear" linearly scales the lineage widths as a function of the population size, the scale factor is defined by the variable linear.scale; "log" scales the lineage width logarithmically as a function of the population size. The logarithm base is defined by the variable log.base.
linear.scale	Linear scale factor, which will be applied when size.scale = "linear".
log.base	The base of logarithm, which will be applied when size.scale = "log".
time.scale	A keyword to define the time scale used in the plot, it can be "4Ne", "generation", "year", "kyear", and "log10year". When the size.scale = "topology", this parameter will be ignored.
gen	Years per generation. Default value is 25.
m.adjust	Migration threshold for plotting migration events. Migration events with strength higher than m.adjust will be shown. The migration strength is defined as the proportion of the target population being replaced by the source population per generation. Default value is 0. This value should be stay between 0 and 1.
col.pop	Color for each population.
col.arrow	Color for each migration arrow.
length.arrowtip	Size of arrow tips.

<code>lwd.arrow</code>	Controls the width of arrow representing a migration. The arrow width is determined by $0.5 + \text{migration strength} * \text{lwd.arrow}$ .
<code>angle.arrowtip</code>	Arrow end angle, between 0 and 90.
<code>pops</code>	Population name labels. Default as 1:number of populations.
<code>xlab</code>	Title for the x-axis.
<code>ylab</code>	Title for the y-axis.
<code>xlim</code>	Range of x-axis.
<code>ylim</code>	Range of y-axis.
<code>plot.out</code>	A logical variable that controls the production of the demographic plot. If TRUE, the demographic plot will be produced.
<code>demo.out</code>	A logical variable that controls the output of the demographic parameters. If TRUE, all demographic parameters that are used for the graph will be returned.
<code>cex.lab</code>	The magnification to be used for x and y labels relative to the current setting of <code>cex</code> .
<code>cex.axis</code>	The magnification to be used for axis annotation relative to the current setting of <code>cex</code> .
<code>axes</code>	A logical value to plot the axes or not.

### Value

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

<code>demograph_out</code>	<p>This list contains all demographic details from the input command file:</p> <ul style="list-style-type: none"> <li>• <code>time.series</code> is a vector of time;</li> <li>• <code>Pos</code> is a numeric matrix of positions for each population at every demographic event;</li> <li>• <code>N</code> is a numeric matrix of population size for each population at every demographic event;</li> <li>• <code>m</code> is 3-D numeric matrix of migration rates between populations at every demographic event;</li> <li>• <code>survive</code> is a matrix recording the begin and end for each population according to the demographic events;</li> <li>• <code>g.rate</code> is a matrix of exponential growth rates at every demographic event;</li> <li>• <code>pop.pos</code> is a numeric vector of the population positions at time 0;</li> <li>• <code>pop.lab</code> is a vector of population names;</li> <li>• <code>mscmd</code> is the <code>ms</code> command for the demographic plot. All simulation script will turn to <code>ms</code> command format for extracting the demographic information;</li> <li>• <code>present.pop.num</code> is the number of populations at present;</li> <li>• <code>total.pop.num</code> is the number of total populations exist in the plot;</li> <li>• <code>N4</code> is the <math>4N_e</math>;</li> <li>• <code>gen</code> is the years per generation.</li> </ul>
<code>evo_par</code>	<p>This list contains all parameters used to draw the demographic graph, including: <code>size.scale</code>, <code>linear.scale</code>, <code>log.base</code>, <code>time.scale</code>, <code>time</code>, <code>col.pop</code>, <code>col.arrow</code>, <code>length.arrowtip</code>, <code>lwd.arrow</code>, <code>angle.arrowtip</code>, <code>lab.pop</code>, <code>lab.pos</code>, <code>xlim</code>, <code>ylim</code>, <code>xlab</code>, <code>ylab</code>, <code>cex.lab</code>, <code>cex.axis</code>, <code>axes</code>. See the more details in the parameter description.</p>



`mig_par` This list contains all parameters used to draw the migrations, including: `size.scale`, `linear.scale`, `log.base`, `time.scale`, `time`, `lab.pop`, `col.pop`, `col.arrow`, `length.arrowtip`, `lwd.arrow`, `angle.arrowtip`, `xlim`, `ylim`, `events`, `cex.lab`. See the more details in the parameter description.

## 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(input.file = "test.1.ms.cmd", type = "ms", N4 = 10000, time.scale = "kyear")
#adjust the population position
PlotMS(input.file = "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(input.file = "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(input.file = "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 -ej 0.197963 2 1
-en 0.303501 1 1", file = "Ryan2009.cmd")
PlotMS(input.file = "Ryan2009.cmd", type = "ms", N4 = 10000, size.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(input.file = "zigzag.cmd", type = "ms", N4 = 10000)
#change the time scale
PlotMS(input.file = "zigzag.cmd", type = "ms", N4 = 10000, time.scale = "log10year")
```

```
unlink("zigzag.cmd")
```

# Index

## \*Topic **Population**

POPdemog-package, [1](#)

## \*Topic **demographic history**

POPdemog-package, [1](#)

NRuler, [3](#)

PlotMig, [2](#), [4](#), [6](#)

PlotMMig, [2](#), [6](#)

PlotMS, [2](#), [4–6](#), [6](#)

POPdemog (POPdemog-package), [1](#)

POPdemog-package, [1](#)