Package 'iMKT'

December 30, 2017

Title McDonald and Kreitman Test and its extensions calculation

Version 0.1.1

Date 2017-12-01

Description McDonald and Kreitman Test and its extensions.

Author Sergi Hervas <sergi.hervas@uab.cat>
Marta Coronado <marta.coronado@uab.cat>
Jesús Murga <jesus.murga@uab.cat>

Maintainer Jesús Murga <jesus.murga@uab.cat>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 3.3), ggplot2,

Imports knitr, utils, stats, cowplot, reshape2, nls2, MASS, ggthemes

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`asymptoticMK` *Asymptotic MKT method*

Description

MKT calculation using asymptoticMK method (Messer and Petrov 2012 PNAS; Haller and Messer 2017 G3)

Usage

```
asymptoticMK(daf, divergence, xlow, xhigh, seed)
```

Arguments

`daf`	data frame containing DAF, Pi and P0 values
`divergence`	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
`xlow`	lower limit for asymptotic alpha fit
`xhigh`	higher limit for asymptotic alpha fit
`seed`	seed value (optional). No seed by default

Details

In the standard McDonald and Kreitman test, the estimate of adaptive evolution (alpha) can be easily biased by the segregation of slightly deleterious non-synonymous substitutions. Specifically, slightly deleterious mutations contribute to polymorphism but not to divergence, and thus, lead to an underestimation of alpha. Messer and Petrov proposed a simple asymptotic extension of the MK test that yields accurate estimates of alpha. Briefly, this method first estimates alpha for each DAF category using its specific Pi and P0 values and then fits an exponential function to this values, of the form: alphaFit(x) = a + b exp(-cx). Finally, the asymptotic alpha estimate is obtained by extrapolating the value of this function to x = 1: alphaAsymptotic = alphaFit(x=1). The code of this function is adapted from Haller and Messer 2017 G3 (http://github.com/MesserLab/asymptoticMK).

Value

Estimation of asymptotic alpha and details about the model fit (function parameters, confidence intervals, etc.)

Examples

```
asymptoticMK(myDafData, myDivergenceData, xlow=0, xhigh=0.9)
```

checkInput *Check input data*

Description

Check input data and return detailed errors when it is malformed.

Usage

```
checkInput(daf, divergence, xlow, xhigh)
```

Arguments

daf	data frame containing DAF, Pi and P0 values
divergence	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
xlow	lower limit for asymptotic alpha fit
xhigh	higher limit for asymptotic alpha fit

Details

Check input data used in most package's functions (arguments daf, divergence, xlow and xhigh) and return a brief description of the error(s) found. This function is called within each analysis function (standardMK, FWW, DGRP, asymptoticMK, iMK) and if data does not pass checkInput() without errors, the requested analysis is not performed.

completeMK *Complete MK methodologies*

Description

MKT calculation using all methodologies included in the package: standardMK, FWW, DGRP, asymptoticMK, iMK.

Usage

```
completeMK(daf, divergence, xlow, xhigh, seed)
```

Arguments

daf	data frame containing DAF, Pi and P0 values
divergence	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
xlow	lower limit for asymptotic alpha fit
xhigh	higher limit for asymptotic alpha fit
seed	seed value (optional). No seed by default

Details

Perform all MKT derived methodologies at the same time using the same input parameters.

Value

List with all MKT results: standardMK, FWW, DGRP, asymptoticMK, iMK

Examples

```
completeMK(myDafData, myDivergenceData, 0, 0.9)
```

DGRP *DGRP correction method*

Description

MKT calculation corrected using DGRP method (Mackay et al. 2012 Nature).

Usage

```
DGRP(daf, divergence, list_cutoffs = c(0, 0.05, 0.1), plot = FALSE)
```

Arguments

daf	data frame containing DAF, Pi and P0 values
divergence	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
list_cutoffs	list of cutoffs to use (optional). Default cutoffs are: 0, 0.05, 0.1
plot	report plot (optional). Default is FALSE

Details

In the standard McDonald and Kreitman test, the estimate of adaptive evolution (alpha) can be easily biased by the segregation of slightly deleterious non-synonymous substitutions. Specifically, slightly deleterious mutations contribute to polymorphism but not to divergence, and thus, lead to an underestimation of alpha. Because adaptive mutations and weakly deleterious selection act in opposite directions on the MKT, alpha and the fraction of substitutions that are slighlty deleterious, b, will be both underestimated when both selection regimes occur. To take adaptive and slighlty deleterious mutations mutually into account, Pi, the count off segregatning sites in class i, should be separated into the number of neutral variants and the number of weakly deleterious variants, Pi = Pineutral + Pi weak del. Alpha is then estimated as 1-(Pineutral/P0)(D0/Di). As weakly deleterious mutations tend to segregate at low frequencies, neutral and weakly deleterious fractions from Pi can be estimated based on any frequency cutoff established.

Value

MKT corrected by the DGRP method. List with alpha results, graph (optional), divergence metrics, MKT tables and negative selection fractions

Examples

```
## Using default cutoffs
DGRP(myDafData, myDivergenceData)
## Using custom cutoffs and rendering plot
DGRP(myDafData, myDivergenceData, c(0.05, 0.1, 0.15), plot=TRUE)
```

FWW *FWW correction method*

Description

MKT calculation corrected using FWW method (Fay et al. 2001 Genetics).

Usage

```
FWW(daf, divergence, list_cutoffs = c(0, 0.05, 0.1), plot = FALSE)
```

Arguments

daf	data frame containing DAF, Pi and P0 values
divergence	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
list_cutoffs	list of cutoffs to use (optional). Default cutoffs are: 0, 0.05, 0.1
plot	report plot (optional). Default is FALSE

Details

In the standard McDonald and Kreitman test, the estimate of adaptive evolution (alpha) can be easily biased by the segregation of slightly deleterious non-synonymous substitutions. Specifically, slightly deleterious mutations contribute to polymorphism but not to divergence, and thus, lead to an underestimation of alpha. Because they tend to segregate at lower frequencies than do neutral mutations, they can be partially controled by removing low frequency polymorphisms from the analysis. This is known as the FWW method.

Value

MKT corrected by the FWW method. List with alpha results, graph (optional), divergence metrics, MKT tables and negative selection fractions

Examples

```
## Using default cutoffs
FWW(myDafData, myDivergenceData)
## Using custom cutoffs and rendering plot
FWW(myDafData, myDivergenceData, c(0.05, 0.1, 0.15), plot=TRUE)
```

---

iMK *integrative MKT method*

Description

iMK: MKT using asymptoticMK method and estimation of negative selection fractions (d, b, f)

Usage

```
iMK(daf, divergence, xlow, xhigh, seed, plot = FALSE)
```

Arguments

daf	data frame containing DAF, Pi and P0 values
divergence	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes
xlow	lower limit for asymptotic alpha fit
xhigh	higher limit for asymptotic alpha fit
seed	seed value (optional). No seed by default
plot	report plots of daf, alpha and negative selection fractions (optional). Default is FALSE

Details

The integrative MKT (iMKT) allows the estimation of the rate of adaptive evolution (alpha) and the diverse negative selection regimens. iMKT uses asymptotic MK method (Messer and Petrov 2012 PNAS; Haller and Messer 2017 G3) to estimate alpha and the diverse negative selection fractions (d: strongly deleterious, b: weakly deleterious, f: neutral), based on the assumption that weakly deleterious mutations usually do not reach high allele frequencies and therefore, produce the underestimation of alpha at low DAF categories. The fraction of strongly deleterious mutations is estimated as the difference between neutral (0) and selected (i) polymorphic sites relative to the number of analyzed sites: d = 1 - (P0/m0 / Pi/mi). The fraction of weakly deleterious sites (b) corresponds to the relative proportion of selected polymorphic sites that cause the underestimation of alpha at low DAF categories. Finally, the fraction of neutral sites (f) is estimated as: f = 1 - d - b.

Value

iMKT method. List with asymptotic MK table and values, fractions of sites and graphs of DAF, asymptotic alpha model and negative selection fractions (optional).

Examples

```
## Without plot
iMK(myDafData, myDivergenceData, xlow=0, xhigh=0.9)
## With plot
iMK(myDafData, myDivergenceData, xlow=0, xhigh=0.9, plot=TRUE)
```

`loadPopFly` *Load PopFly dataset*

Description

Load PopFly dataset with information regarding protein coding gene annotations.

Usage

```
loadPopFly()
```

Details

This function loads PopFly data (Hervas et al. 2017 Bioinformatics, http://popfly.uab.cat/) into the current workspace. Data is stored in a dataframe named PopFlyData, which includes population genetics estimates (nucleotide diversity, divergence, basic tests of neutrality, recombination rates, etc.) regarding each protein coding gene for 16 worldwide wild-derived Drosophila melanogaster populations from the Drosophila Genome Nexus project (Lack et al. 2015 Genetics, Lack et al. 2016 MBE).

Value

PopFlyData object loaded in the workspace

Examples

```
## Load PopFly data if necessary. This process may take several seconds to complete.
# loadPopFly()
```

loadPopHuman *Load PopHuman dataset*

Description

Load PopHuman dataset with information regarding protein coding gene annotations.

Usage

```
loadPopHuman()
```

Details

This function loads PopHuman data (Mulet et al. 2017 NAR, http://pophuman.uab.cat/) into the current workspace. Data is stored in a dataframe named PopHumanData, which includes population genetics estimates (nucleotide diversity, divergence, basic tests of neutrality, recombination rates, etc.) regarding each protein coding gene for 26 worldwide Homo sapiens populations from the 1000 Genomes Project (The 1000 Genomes Project Consortium 2012 Nature, The 1000 Genomes Project Consortium 2015 Nature).

Examples

```
## Load PopHuman data if necessary. This process may take several seconds to complete.
# loadPopHuman()
```

myDafData *Example data frames*

Description

Data frame containing polymorphism sample data

- daf. derived allele frequency (DAF) categories
- Pi. number of selected (i) polymorphic sites for each daf category
- P0. number of neutral (0) polymorphic sites for each daf category

Usage

```
myDafData
```

Format

A data frame containing polymorphic sites for selected (i) and neutral (0) classes at different DAF categories

myDivergenceData *Example data frames*

Description

Data frame containing divergence sample data

- mi. number of selected (i) analyzed sites
- Di. number of selected divergent sites
- m0. number of neutral (0) analyzed sites
- D0. number of neutral divergent sites

Usage

```
myDivergenceData
```

Format

A data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes

PopFlyAnalysis *PopFlyAnalysis*

Description

Perform any MK test using a subset of PopFly data defined by custom genes and populations lists

Usage

```
PopFlyAnalysis(genes = c("gene1", "gene2", "..."), pops = c("pop1", "pop2",
  "..."), recomb = TRUE/FALSE, bins = 0, test = c("standardMK", "DGRP",
  "FWW", "asymptoticMK", "iMK"), xlow = 0, xhigh = 1)
```

Arguments

genes	list of genes
pops	list of populations
recomb	group genes according to recombination values (must specify number of bins). TRUE/FALSE. Recomb values (cM/Mb) from Comeron et al. 2012.
bins	number of recombination bins to compute (mandatory if recomb = TRUE)
test	which test to perform. Options include: standardMK (default), DGRP, FWW, asymptoticMK, iMK
xlow	lower limit for asymptotic alpha fit (default=0)
xhigh	higher limit for asymptotic alpha fit (default=1)

Details

Recombination values (recomb=T) from Comeron et al. 2012 (reference!)

Value

None

Examples

```
## List of genes
# mygenes <- c("FBgn0053196", "FBgn0086906", "FBgn0261836", "FBgn0031617",
#              "FBgn0260965", "FBgn0028899", "FBgn0052580", "FBgn0036181",
#              "FBgn0263077", "FBgn0013733", "FBgn0031857", "FBgn0037836")
## Perform analyses
# PopFlyAnalysis(genes=mygenes , pops=c("RAL","ZI"), recomb=F, test="iMK", xlow=0, xhigh=0.9)
# PopFlyAnalysis(genes=mygenes , pops=c("RAL","ZI"), recomb=T, bins=3, test="DGRP")
```

`PopHumanAnalysis` *PopHumanAnalysis*

Description

Perform any MK test using a subset of PopHuman data defined by custom genes and populations lists

Usage

```
PopHumanAnalysis(genes = c("gene1", "gene2", "..."), pops = c("pop1",
  "pop2", "..."), recomb = TRUE/FALSE, bins = 0, test = c("standardMK",
  "DGRP", "FWW", "asymptoticMK", "iMK"), xlow = 0, xhigh = 1)
```

Arguments

`genes`	list of genes
`pops`	list of populations
`recomb`	group genes according to recombination values (must specify number of bins). TRUE/FALSE. Recomb values (cM/Mb) from Bheller et al.
`bins`	number of recombination bins to compute (mandatory if recomb = TRUE)
`test`	which test to perform. Options include: standardMK (default), DGRP, FWW, asymptoticMK, iMK
`xlow`	lower limit for asymptotic alpha fit (default=0)
`xhigh`	higher limit for asymptotic alpha fit (default=1)

Details

Recombination values (recomb=T) from Bheller et al. (reference!)

Value

None

Examples

```
## List of genes
# mygenes <- c("AHNAK2","MUC5B","MUC4","TTN","MUC16","PLIN4",
#              "OBSCN","PLEC","MUC12","PKD1","LAMA5","HELZ2")
## Perform analyses
# PopHumanAnalysis(genes=mygenes , pops=c("CEU","YRI"), recomb=F, test="standardMK")
# PopHumanAnalysis(genes=mygenes , pops=c("CEU"), recomb=T, bins=3, test="DGRP")
```

`standardMK` *Standard MKT*

Description

Standard MKT calculation (McDonald and Kreitman 1991 Nature).

Usage

```
standardMK(daf, divergence)
```

Arguments

`daf`	data frame containing DAF, Pi and P0 values
`divergence`	data frame containing divergent and analyzed sites for selected (i) and neutral (0) classes

Details

The standard McDonald and Kreitman test (MKT) is used to detect the signature of selection at the molecular level. The MKT compares the amount of variation within a species (polymorphism, P) to the divergence (D) between species at two types of sites, one of which is putatively netral and used as the reference to detect selection at the other type of site. In the standard MKT, these sites are synonymous (putatively neutral, 0) and non-synonymous sites (selected sites, i) in a coding region. Under strict neutrality, the ratio of the number of selected and neutral polymorphic sites (Pi/P0) is equal to the ratio of the number of selected and neutral divergence sites (Di/D0). The null hypothesis of neutrality is rejected in a MKT when Di/D0 > Pi/P0. The excess of divergence relative to polymorphism for class i, is interpreted as adaptive selection for a subset of sites i. The fraction of adaptive fixations (alpha) is estimated from 1-(Pi/P0)(Ds/Dn). The significance of the test can be assesed with a Fisher exact test.

Value

Standard MKT. List with alpha estimate, Fisher's exact test p-value, MKT table and divergence metrics.

Examples

```
standardMK(myDafData, myDivergenceData)
```

`themePublication` *ggplot Theme for publication ready plots*

Description

Theme with the configuration and parameters necessary to generate publication ready plots using ggplot

Usage

```
themePublication(base_size = 14, base_family = "sans")
```

Arguments

`base_size`	base size required from themePublication
`base_family`	font to load in themePublication

Details

Theme used for plot images developed by Koundinya Desiraju (04/07/2015). Code adapted from http://rpubs.com/Koundy/71792.

Value

plot theme

Index