Package 'BlockFeST'

May 7, 2018

Type Package

Title Bayesian Calculation of Region-Specific Fixation Index to Detect Local Adaptation

Version 1.6

Date 2018-05-6

Author Bastian Pfeifer

Maintainer Bastian Pfeifer <bastianxpfeifer@gmail.com>

Depends R (>= 2.14.2), BASIX

Imports methods

Suggests PopGenome

Description An R implementation of an extension of the 'BayeScan' software (Foll, 2008) <DOI:10.1534/genetics.108.092221> for codominant markers, adding the option to group individual SNPs into pre-defined blocks. A typical application of this new approach is the identification of genomic regions, genes, or gene sets containing one or more SNPs that evolved under directional selection.

License GPL-2

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-05-07 09:12:22

R topics documented:

BlockFeST *Bayesian calculation of region-specific Fixation Index (FST) to detect local adaptation*

Description

The method is based on the work from Beaumont and Balding (2004) where they introduce a F_{ST} based hierarchical Bayesian model to detect loci that are subject to selection. In this Bayesian approach they use a logistic regression model to distinguish between locus-specific effects like selection and population-specific effects which are shared by all loci (e.g effects caused by migration rates) (Riebler, 2008). Foll and Gaggiotti (2008) extended this work using a reversible jump MCMC (Green, 1995) which enables testing the hypothesis that a locus is subject to selection; a very similar approach was developed in parallel by Riebler & Stefan (2008). The method is implemented in a software named BayeScan (http://cmpg.unibe.ch/software/BayeScan/). The new method introduced here is a modification of BayeScan (see details).

Usage

```
BlockFeST(input,GROUP=FALSE,nb=20,runtime=500)
```

Arguments

input	textfile or an R-object returned from the function getBayes() provided by the R-package PopGenome
GROUP	SNP groups
nb	number of MCMC runs
runtime	length of MCMC runs

Details

BlockFeST considers all SNPs separately but generates exactly one region-specific alpha for each region (or group of SNPs). Example files can be found in the subdirectory "exdata".

Value

returned value is an object of class "BAYESRETURN"

Following Slots will be filled

---

alpha	alpha (α) effects

`beta`	beta (β) effects
`var_alpha`	variance of alphas
`fst`	Fe_{ST} values
`region.names`	names of region

References

[1] Foll M and OE Gaggiotti (2008). *A genome scan method to identify selected loci appropriate for both dominant and codominant markers: A Bayesian perspective.* Genetics 180: 977-993

[2] Beaumont M, Balding D. 2004. *Identifying adaptive genetic divergence among populations from genome scans*.Molecular Ecology. 13:969-980.

[3] Riebler A, Held L, Stephan W. 2008. *Bayesian variable selection for detecting adaptive genomic differences among populations*. Genetics 178: 1817-1829

[4] Green PJ. 1995. *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*. Biometrika 82: 711-732.

Examples

```
snps   <- system.file("extdata", "snps.txt", package="BlockFeST")
groups <- system.file("extdata", "groups.txt", package="BlockFeST")
BlockFeST.result <- BlockFeST(input=snps, GROUP=groups, nb=3, runtime=10)
P <- calcPval(BlockFeST.result)
```

`calcPval` *Calculates empirical P-values*

Description

We propose a simple sampling scheme to verify significant outlier loci subject to local adaptation based on the distribution of the α_I values observed after the MCMC iterations (approximated through a region-specific normal distribution $(N(\alpha_I))$ without using a time consuming reversible jump model for testing the null hypotheses:

1. For each region I, sample a single value $x_I \sim N(\alpha_I)$ and $y_I \sim N(\alpha_I)$, resulting in a distribution of sampled values D_x and D_y across regions.

2. For each region I, increment its counter c_I if y_I is above the $q - quantile$ for D_x.

3. repeat (1-2) 1000 times

The empirical P-value for each α_I is the number of times the sample x_I is greater than the user-defined significance level q (e.g., the 0.95 quantile) divided by the number of iterations (1000 times).

Usage

```
calcPval(BlockFeST.result, q=0.95)
```

Arguments

`BlockFeST.result`	an object returned from the function BlockFeST
`q`	quantile

Value

empirical P-values

Examples

```
snps   <- system.file("extdata", "snps.txt", package="BlockFeST")
groups <- system.file("extdata", "groups.txt", package="BlockFeST")
BlockFeST.result <- BlockFeST(input=snps, GROUP=groups, nb=3, runtime=10)
P <- calcPval(BlockFeST.result)
```

Index