Package 'BIGf90'

August 8, 2024

Title R face front for running K-fold crossvalidation, estimating ebvs and variance component estimation with Blupf90 modules

Version 0.3.2

Maintainer Josue Chinchilla-Vargas <jc3635@cornell.edu>

Description

This package provides R functions to run several BLUPf90 modules. Along with an R function to run K-fold crossvalidation for univariate analyses through blupf90+.
The output table from the K-fold crossvalidation function calculates accuracy as cor(y,ebv^) and cor(y*,ebv^) along with bias of ebvs calculated as reg(y,ebv.)
You will need to create the .par file to feed into Renumf90 manually. Once this is done, there are functions to run Renumf90, Blupf90+, Predictf90, Gibbsf90+ and Postgibbsf90.
If you need to learn how to use the blupf90 suite of programs refer to nce.ads.uga.edu/wiki/doku.php?id=start. Please remember to cite Blupf90 appropriately along with this package when used for publications.
To run the functions in this package you will need to have a directory with all the blupf90 executables so that you can indicate the path for R to find the execs.
As of version 0.3.0, all functions have been tested in PC and Unix environments, we have noticed that PC environments tend to have issues about permissions, so it is advised to run RStudio as administrator when planning to use this package.

License Apache License (== 2.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat

Imports base (>= 4.3.1),
dplyr (>= 1.1.4),
utils (>= 4.3.1)

R topics documented:

`bf90_cv` *Run K-fold cross-validation analysis (CVA)*

Description

This function runs a K-fold cross-validation analysis (CVA) using blupf90 modules.

Usage

```
bf90_cv(
  missing_value_code = NULL,
  random_effect_col = NULL,
  h2 = NULL,
  num_runs = NULL,
  num_folds = NULL,
  path_2_execs = ".",
  input_files_dir = ".",
  output_files_dir = ".",
  output_table_name = NULL,
  renf90_ped_name = NULL,
  snp_file_name = NULL,
  seed = 101919,
  verbose = TRUE
)
```

Arguments

`missing_value_code`
: code used in the .par file after OPTION MISSING to indicate missing phenotype, if this option is no use, this value must be 0.

`random_effect_col`
: Column where random effect is located, found under RANDOM_GROUP in the renf90.par file.

`h2`
: estimate of narrow-sense heritabilty.This value is use to calculate accuracy of ebvs

`num_runs`
: Number of independent cross-validation runs to be performed.

`num_folds`
: Number of folds to be generated within each independent run.

`path_2_execs`
: path to a folder that holds all blupf90 executables that ill be used (blupf90+,predictf90). This field should be in quotes "".

`input_files_dir`
: directory containing files renf90.par renf90.fields renf90.inb renf90.tables renf90.dat

`output_files_dir`
: path to directory to store the output files

`output_table_name`
: Name of the final tab-separated out-up file. This field should be in quotes "".

`renf90_ped_name`
: Name of pedigree file generated by renumf90. This field should be in quotes "".

snp_file_name	the name of the genotype file to be used. If use, this field should be in quotes"". Default for the function is no genotype file.
seed	set seed for the stochastic process
verbose	logical defining if information will be printed on the console

Details

This function sets up and runs a K-fold cross-validation analysis (CVA) using blupf90+ and predictf90. The function run_renumf90 needs to be used beforehand to process a .par file created by the user. This function calculates 2 accuracy estimates: correlation between raw phenotypes and ebvs divided by the square-root of narrow sense heritabilty and correlation between corrected phenotypes and ebvs along with bias estimations calculated as the regression of the phenotypes on the ebvs.

Value

a tab-separated file that includes accuracy and bias estimates of ebvs.

Examples

```
## Example for a CVA with 5 independent runs dividing the data in 10 folds.


# bf90_cv(path_2_execs = "/Users/johndoe/Desktop/bf90_execs/",
#      missing_value_code = -999,
#      random_effect_col= 3,
#      h2 = 0.5,
#      num_runs = 5,
#      num_folds = 10,
#      output_table_name = "example_run",
#     renf90_ped_name = "renadd03.ped",
#     snp_file_name = "my_genos.geno" )
```

---

clean_ebvs *Clean ebvs*

Description

This function formats BLUP solutions.

Usage

```
clean_ebvs(random_effect_col, solutions_output_name)
```

Arguments

random_effect_col	Column where random effect is located, found under RANDOM_GROUP in the renf90.par file.
solutions_output_name	name for the output file. This field should be in quotes "".

Details

This function cleans and formats the raw solutions file produced by blupf90+ by removing the solutions to all effects other than the random effect (ebvs). I also matches the processed ID assigned by renumf90 to the original ID for each individual tested producing a interpretable output file with two columns: ID and EBV.

Value

a tab-separated file that includes the original id and ebv for all individuals for which an EBV was produced.

Examples

```
#clean_ebvs(3, "my_clean_ebvs")
```

`execute_command` *Function to run commands on the terminal and log output*

Description

Function to run commands on the terminal and log output

Usage

```
execute_command(command, logfile)
```

Arguments

`command`	comment line used to run executable file
`logfile`	logfile name

`run_blup` *Run blupf90+*

Description

This function runs blupf90+ using a pre-processed parameter file called renf90.par.

Usage

```
run_blup(path_2_execs)
```

Arguments

`path_2_execs`	path to a folder that holds the renumf90 executable. This field should be in quotes "".

Details

This function runs blupf90+ using a parameter file named renf90.par. Since this function depends only on the renf90.par parameter file, the only input needed from the user is a path where the blupf90+ executable is located. A log file called run_blup.log is also produced.

Examples

```
## Example

# run_blup(path_2_execs = "/Users/johndoe/Desktop/bf90_execs/")
```

`run_gibbs` *Run gibbsf90+*

Description

This function runs gibbsf90+.

Usage

```
run_gibbs(
  path_2_execs,
  input_files_dir = ".",
  output_files_dir = input_files_dir,
  gibbs_iter = 250000,
  gibbs_burn = 20000,
  gibbs_keep = 1,
  verbose = TRUE
)
```

Arguments

`path_2_execs`	path to a folder that holds the renumf90 executable. This field should be in quotes "".
`input_files_dir`	path to renf90.par file generated by run_renum function
`output_files_dir`	path to store the result files
`gibbs_iter`	number of samples in the Gibbs sampler.
`gibbs_burn`	number of samples to be discarded at the begining of the Gibbs sampler
`gibbs_keep`	the interval to save samples (thinning). Entering a 1 means all samples are kept.
`verbose`	logical if TRUE prints log information

Details

This function runs gibbsf90+ using input from the user and a renf90.par file. This function is written to use a renf90.par paramererfile, therefore the function run_renumf90 needs to be used beforehand to process a .par file created by the user. The user will have to enter the number of samples in the MCMC chain, the number of samples to burn and the number used to thin samples. A log file called run_gibbs.log is also produced.

Examples

```
## Example

# run_gibbs( path_2_execs = "/Users/johndoe/Desktop/bf90_execs/",
# gibbs_iter = 250000,
# gibbs_burn = 20000
# gibbs_keep = 1)
```

`run_postgibbs` *Run postgibbsf90*

Description

This function runs postgibbsf90 using the renf90.par file used to run gibbsf90+

Usage

```
run_postgibbs(
  path_2_execs,
  input_files_dir = ".",
  output_files_dir = input_files_dir,
  postgibbs_burn,
  postgibbs_keep,
  verbose = TRUE
)
```

Arguments

`path_2_execs`	path to a folder that holds the renumf90 executable. This field should be in quotes "".
`input_files_dir`	path to renf90.par file generated by run_renum function
`output_files_dir`	path to store the result files
`postgibbs_burn`	number of samples to be discarded at the begining of the Gibbs sampler
`postgibbs_keep`	the interval to save samples (thinning). Entering a 1 means all samples are kept.#'
`verbose`	logical if TRUE prints log information in the console

Details

This function runs postgibbsf90 to provide diagnostic statistics,posterior means and standard deviations for analyses performed through gibbsf90+. This function is written to run using the same renf90.par file used with run_gibbs and its output files are be the standard output files produced by postgibbsf90. The user will have to enter the number of samples to burn and the number used to thin samples. A log file called run_postgibbs.log is also produced.

Examples

```
## Example

# run_postgibbs( path_2_execs = "/Users/johndoe/Desktop/bf90_execs/",
# postgibbs_burn =1,
# postgibbs_keep = 100)
```

`run_predict` *Run predictf90*

Description

This function runs predict90.

Usage

```
run_predict(path_2_execs)
```

Arguments

`path_2_execs`	path to a folder that holds the renumf90 executable. This field should be in quotes "".

Details

This function runs predictf90 using a pre-processed parameter file called renf90.par to calculate adjusted phenotypes. This function is written to use a renf90.par file and a solutions file. Therefore run_renum and run_blup should be ran before using this function. The output files are be the standard output files produced by predictf90. A log file called run_predict.log is also produced.

Examples

```
## Example

# run_predict(path_2_execs = "/Users/johndoe/Desktop/bf90_execs/")
```

`run_renum` *Run renumf90*

Description

This function runs renumf90.

Usage

```
run_renum(
  path_2_execs = ".",
  raw_par_file = NULL,
  output_files_dir = NULL,
  verbose = TRUE
)
```

Arguments

path_2_execs	path to a folder that holds the renumf90 executable. This field should be in quotes "".
raw_par_file	name of the .par file that will be processed. This field should be in quotes "".
output_files_dir	path to the folder to store the output files renadd03.ped renf90.dat renf90.fields renf90.inb renf90.par renf90.tables run_renum.log
verbose	logical if TRUE prints log information

Details

This function runs renumf90 to process the raw parameter(.par) file to be used with the blupf90 suite of programs. The outputs will be the standard output files produced by renumf90. A log file called run_renum.log is also produced.

Examples

```
#run_renum(path_2_execs = "path/bf90_execs/",
#input_files_dir = "weight_2022_no_cov_cv.par")
```