PIXANT Manual

Rapid and accurate multi-phenotype imputation for millions of individuals

Version: 0.1.0

Date: August, 21th 2023

Author: Lin-Lin Gu

Maintainer:

Lin-Lin Gu: linlin-gu@outlook.com

Table of Contents

1 Brief introduction

Deep phenotype datasets can enhance the power of genetic analysis such as genome-wide association study (GWAS), but recurrence of missing phenotypes compromises the potentials of such resources. Here we address the central issue of missing phenotypes in studies with any level of relatedness between phenotypes. we propose a multi-phenotype imputation method that is scalable to large data over a million individuals. We call our method PIXANT (multi-**p**henotype **i**mputation method based on mi**x**ed f**a**st ra**n**dom fores**t**). PIXANT models nonlinear and linear effects across multi-phenotype correlation and higher-order interactions between predictive factors, and brings out unbiased imputation of much higher accuracy than state-of-the-art methods. Tested in a dataset of *n*=20,000 individuals and *p*=30 phenotypes, PIXANT is ~24.45 times faster and uses only about one ten thousandth memory compared with PHENIX(***Nature Genetics*** 2016 (48):466–472). Moreover, real data set analysis and biologically plausible results suggest that our method imputation can uncover new true positive results.

2 PIXANT function

PIXANT	Imputing missing values by using the mixed fast random forest.

2.1 Description

Imputing missing values by using the mixed fast random forest. The PIXANT method comprises two parts: 1) the estimation part: estimating parameters under the null model and applying the likelihood to correct for linear and nonlinear effects; 2) the imputation part: performing imputation for the missing values.

2.2 Usage

```
PIXANT(data, aimPhenName, maxIterations = 20, maxIterations0 = 20, num.trees =
            100, initialLinearEffects = 0, errorTolerance = 0.001, aimPhenMissingSize
            = 500, initialImputeType = 'random', refPhenThreshold = 0.3,
            maxNum.refPhen = 10, SC.Threshold = 0.6, seed = 123, decreasing =
            TRUE, verbose = TRUE, xtrue = NA)
```

2.3 Arguments

data	A data frame with missing values.
aimPhenName	The column name for the imputed phenotype.
maxIterations	Stop after how many iterations (default = 20).
maxIterations0	The maximum iteration times of mixed fast random forest (default = 20).
num.trees	How many trees are grown in the mixed fast random forest (default = 100).
initialLinearEffects	The initial values for linear effects (default = 0).
errorTolerance	The tolerance for log-likelihood (default = 0.001).
indPhen	The columns of the target phenotype, and used only for cross-validation operation (default = NULL).
aimPhenMissingSize	The missing values size of settings in the imputed phenotype, and must be less than number of observed values of the imputed phenotype (default = 500).
initialImputeType	Initial imputation method for missing values in a data frame. Currently support: "random", "average", "median".
refPhenThreshold	Relevance threshold for the selection of reference phenotypes, the range from 0 to 1 (default = 0.3).
minNum.refPhen	Maximum number of reference phenotypes to be selected (default = 10).
SC.Threshold	Individual's imputation quality threshold (default = 0.6).
seed	Random seed. Default is 123, which generates the seed from R. Set to 0 to ignore the R seed.
decreasing	(boolean) If TRUE the columns are sorted with decreasing amount of missing values.
verbose	(boolean) If TRUE then PIXANT returns error estimates, runtime and if available true error during iterations.
xtrue	The complete true data (a vector, matrix or data frame).

2.4 Value

Return a list, the list contains:

ximp	The imputed data (a data frame).
imputePhen.accuracy	Imputation accuracy of imputed phenotype.
imputePhen.value	The missing data of settings in the imputed phenotype, including sample index, observed values and imputed values (a data frame).

imputePhen.refPhen	A list of reference phenotypes for impute phenotype.
imputePhen.missingRate	Original missing rate of the imputed phenotypes.
imputePhen	The impute phenotype, including imputed values and SC for each samples (a data frame).
imputePhen.filter.missingRate	Missing rate after SC quality control of the imputed phenotype.
imputePhen.r2	The r square of fitting between observed values and imputed values in the imputed phenotype.
imputePhen.pValue	The p value of fitting between observed values and imputed values in the imputed phenotype.

3 PIXANT.eval function

PIXANT.eval	PIXANT evaluation indicator.

3.1 Description

The function calculates correlation coefficient between imputed values and their true hidden values.

3.2 Usage

PIXANT.eval(ximp, xmis, xtrue)

3.3 Arguments

ximp	The imputed data (a vector, matrix or data frame).
xmis	The data with missing values.
xtrue	The complete true data.

3.4 Value

Return the correlation coefficient between the real values and the imputed values.

4 prodNA function

prodNA	Produce missing values.

4.1 Description

This R script contains the function to produce missing values in a given and data set completely at random.

4.2 Usage

prodNA(x, noNA, seed)

4.3 Arguments

x	A vector, matrix or data frame.
noNA	Proportion of missing values to add to x. In case x is a data frame, noNA can also be a vector of probabilities per column or a named vector (see examples).
seed	An integer seed.

5 simG function

simG	Simulated Genome Relationship Matrix.

5.1 Description

The function simulates the construction of a genome relational matrix.

5.2 Usage

simG(N, k, fam_size)

5.3 Arguments

N	The number of individuals and must be a positive integer.
k	Coefficient of kinship and the value ranges from 0 to 1.
fam_size	The size of the family, fam_size must be a positive integer and must divide N.

6 simPhen function

simPhen	Simulated phenotype.

6.1 Description

This function simulates the phenotypes for individuals.

6.2 Usage

```
simPhen(N = N, P = P, K = G, h2 = rep(0.6, P), B, E)
```

6.3 Arguments

N	The number of individuals.
P	The number of phenotypes.
K	A genome relational matrix.
h2	The heritability of each phenotype in individuals.
B	Genetic covariance. (allow the missing)
E	Environmental or residual covariance. (allow the missing)

7 imputeUnivariate function

imputeUnivariate	Univariate Imputation.

7.1 Description

Fills missing values of a vector, matrix or data frame by sampling with replacement from the non-missing values. For data frames, this sampling is done within column.

7.2 Usage

```
imputeUnivariate(x, initialImputeType, v = NULL, seed = NULL)
```

7.3 Arguments

x	A data frame with missing values.

initialImputeType	Initial imputation method for missing values in a data frame. Currently support: "random", "average", "median".
v	A character vector of column names to impute (only relevant if x is a data frame). The default NULL imputes all columns.
seed	An integer seed.

8 PhenAdj function

PhenAdj	Adjusted phenotypic values.

7.1 Description

Adjusted phenotypic values base on covariates.

7.2 Usage

PhenAdj(Phen, Cov)

7.3 Arguments

Phen	Phenotype file. The missing values should be denoted by NA.
Cov	A matrix of covariates. Each row is a sample and each column corresponds to one covariate. For example, age, gender.

7.4 Value

Return adjusted phenotype file.

8 sampleScore function

SC	QC.

8.1 Description

Estimating the SC (Sample Score) for each phenotype of each individual.

8.2 Usage

SC(Phen, aimPhenName,use="pairwise",method="pearson",adjust="fdr",alpha=.05)

8.3 Arguments

Phen
Phenotype file. The missing values should be denoted by NA.

aimPhenName
The column name for the imputed phenotype.

use
use="pairwise" is the default value and will do pairwise deletion of cases. use="complete" will select just complete cases.

method
method="pearson" is the default value. The alternatives to be passed to cor are "spearman" and "kendall". These last two are much slower, particularly for big data sets.

adjust
What adjustment for multiple tests should be used? ("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"). See p.adjust for details about why to use "holm" rather than "bonferroni").

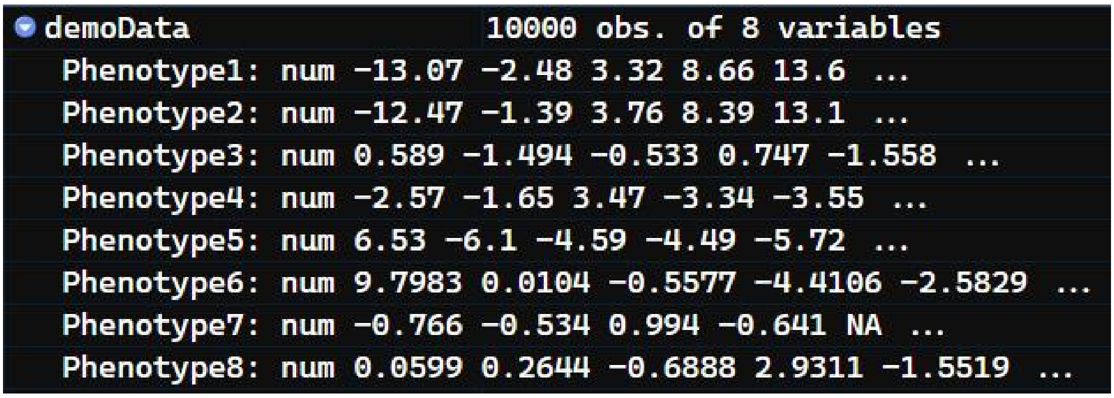
alpha
alpha level of confidence intervals.

8.4 Value

Return the SC (Sample Score) for imputed phenotype of each individual.

9 Build in data

An example dataset ‘demoData’ is the simulation data set. The data including a data frame (10000 * 8), each row represents 8 phenotypes information for an individual. The ‘demoData’ can be loaded with data(demoData).



```
demoData                    10000 obs. of 8 variables
  Phenotype1: num -13.07 -2.48 3.32 8.66 13.6 ...
  Phenotype2: num -12.47 -1.39 3.76 8.39 13.1 ...
  Phenotype3: num 0.589 -1.494 -0.533 0.747 -1.558 ...
  Phenotype4: num -2.57 -1.65 3.47 -3.34 -3.55 ...
  Phenotype5: num 6.53 -6.1 -4.59 -4.49 -5.72 ...
  Phenotype6: num 9.7983 0.0104 -0.5577 -4.4106 -2.5829 ...
  Phenotype7: num -0.766 -0.534 0.994 -0.641 NA ...
  Phenotype8: num 0.0599 0.2644 -0.6888 2.9311 -1.5519 ...
```

9.1 Running build-in data

```
library("PIXANT")
data(demoData)
system.time(PIXANT.imp <- PIXANT(demoData, aimPhenName = 'Phenotype7',
                          maxIterations = 20, maxIterations0 = 20,num.trees =
                          100, initialLinearEffects = 0, errorTolerance =
                          0.001,aimPhenMissingSize = 500, initialImputeType =
                          'random', refPhenThreshold = 0.3,maxNum.refPhen = 7,
                          SC.Threshold = 0.6, seed = 123))
```

10 Code availability

The source code of PIXANT is freely available.

11 How to access help

If users have any bugs or issues or any suggestions are available, feel free to contact:

Lin-Lin Gu: linlin-gu@outlook.com