Package 'smartsnp'

October 14, 2022

Type Package

Title Fast Multivariate Analyses of Big Genomic Data

Version 1.1.0

Date 2021-03-01

Maintainer Christian Huber <christian.domitian.huber@gmail.com>

Description Fast computation of multivariate analyses of small (10s to 100s markers) to big (1000s to 100000s) genotype data. Runs Principal Component Analysis allowing for centering, z-score standardization and scaling for genetic drift, projection of ancient samples to modern genetic space and multivariate tests for differences in group location (Permutation-Based Multivariate Analysis of Variance) and dispersion (Permutation-Based Multivariate Analysis of Dispersion).

Language en-GB

License MIT + file LICENSE

Depends R (>= 3.6.0)

Imports bootSVD, data.table, foreach, Rfast, RSpectra, vegan, vroom, Rcpp

RoxygenNote 7.1.1

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Salvador Herrando-Perez [aut] (<https://orcid.org/0000-0001-6052-6854>), Ray Tobler [ctb] (<https://orcid.org/0000-0002-4603-1473>), Christian Huber [ctb, cre] (<https://orcid.org/0000-0002-2267-2604>)

Repository CRAN

Date/Publication 2021-03-04 09:50:02 UTC

R topics documented:

`read_packedancestrymap` *Read Files in PACKEDANCESTRYMAP format*

Description

This function loads genotype data in `PACKEDANCESTRYMAP` format (binary or compressed).

Usage

```
read_packedancestrymap(pref)
```

Arguments

`pref`	The prefix of the file name that contains the genotype data (i.e., without the `*.geno`).

Value

Returns a list containing a single element:

- `geno` Genotype data as R matrix.

`smart_mva` *Smart Multivariate Analyses (wrapper of PCA, PERMANOVA and PERMDISP)*

Description

Computes Principal Component Analysis (PCA) for variable x sample genotype data, such as Single Nucleotide Polymorphisms (SNP), in combination with Permutational Multivariate Analysis of Variance (PERMANOVA) and Permutational Multivariate Analysis of Dispersion (PERMDISP). A wrapper of functions `smart_pca`, `smart_permanova` and `smart_permdisp`. Genetic markers such as SNPs can be scaled by `centering`, z-scores and genetic drift-based dispersion. The latter follows the SMARTPCA implementation of Patterson, Price and Reich (2006). Optimized to run fast computation for big datasets.

Arguments

snp_data	snp_dataFile name read from working directory. SNP = rows, samples = columns without row names or column headings. SNP values must be count data (no decimals allowed). File extension detected automatically whether text or EIGENSTRAT. See details.
packed_data	Logical value for EIGENSTRAT, irrelevant for text data. Default packed_data = FALSE assumes uncompressed EIGENSTRAT. packed_data = TRUE for compressed or binary EIGENSTRAT (PACKENDANCESTRYMAP).
sample_group	Character or numeric vector assigning samples to groups. Coerced to factor.
sample_remove	Logical FALSE or numeric vector indicating column numbers (samples) to be removed from computations. Default sample_remove = FALSE keeps all samples.
snp_remove	Logical FALSE or numeric vector indicating row numbers (SNPs) to be removed from computations. Default snp_remove = FALSE keeps all SNPs. See details.
pca	Logical indicating if PCA is computed. Default TRUE.
permanova	Logical indicating if PERMANOVA is computed. Default TRUE
permdisp	Logical indicating if PERMDISP is computed. Default TRUE.
missing_value	Number 9 or string NA indicating missing value. Default missing_value = 9 as in EIGENSTRAT. If no missing values present, no effect on computation.
missing_impute	String handling missing values. Default missing_impute = "mean" replaces missing values of each SNP by mean of non-missing values across samples. missing_impute = "remove" removes SNPs with at least one missing value. If no missing values present, no effect on computation.
scaling	String. Default scaling = "drift" scales SNPs to control for expected allele frequency dispersion caused by genetic drift (SMARTPCA). scaling = "center" for centering (covariance-based PCA). scaling = "sd" for centered SNPs divided by standard deviation (correlation-based PCA). scaling = "none" for no scaling. See details.
program_svd	String indicating R package computing single value decomposition (SVD). Default program_svd = "Rspectra" for svds. program_svd = "bootSVD" for fastSVD. See details.
sample_project	Numeric vector indicating column numbers (ancient samples) projected onto (modern) PCA space. Default sample_project = FALSE implements no projection. See details.
pc_project	Numeric vector indicating the ranks of the PCA axes ancient samples are projected onto. Default pc_ancient = c(1, 2) for PCA axes 1 and 2. If program_svd = "RSpectra", length(pc_ancient) must be smaller than or equal to pc_axes. No effect on computation, if no ancient samples present.
sample_distance	Type of inter-sample proximity computed (distance, similarity, dissimilarity). Default is Euclidean distance. See details.
program_distance	A string value indicating R package to estimate proximities between pairs of samples. Default program_distance = "Rfast" uses function Dist; program_distance = "vegan" uses vegdist. See details.

`target_space` String. Default `target_space = "multidimensional"` applies PERMANOVA and/or PERMDISP to sample-by-sample triangular matrix computed from variable-by-sample data, `pc_axes` has no effect on computation. `target_space = "pca"` applies PERMANOVA and/or PERMDISP to sample-by-sample data in PCA space, `pc_axes` determines number of PCA axes for testing.

`pc_axes` Number of PCA axes computed always starting with PCA axis 1. Default `pc_axes = 2` computes PCA axes 1 and 2 if `target_space = "pca"`. No effect on computation if `target_space = "multidimensional"`.

`pairwise` Logical. Default `pairwise = FALSE` computes global test. `pairwise = TRUE` computes global and pairwise tests.

`pairwise_method`
String specifying type of correction for multiple testing. Default `"holm"`.

`permutation_n` Number of permutations resulting in PERMANOVA/PERMDISP test *p value*. Default `9999`.

`permutation_seed`
Number fixing random generator of permutations. Default `1`.

`dispersion_type`
String indicating quantification of group dispersion whether relative to spatial `"median"` or `"centroid"` in PERMDISP. Default `"median"`.

`samplesize_bias`
Logical. `samplesize_bias = TRUE` for dispersion weighted by number of samples per group in PERMDISP. Default `pairwise = FALSE` for no weighting.

Details

See details in other functions for conceptualization of PCA (`smart_pca`) (Hotelling 1993), SMART-PCA (Patterson, Price and Reich 2006), PERMANOVA (`smart_permanova`) (Anderson 2001) and PERMDISP (`smart_permdisp` (Anderson 2006), types of scaling, ancient projection, and correction for multiple testing.

Users can compute any combination of the three analyses by assigning `TRUE` or `FALSE` to `pca` and/or `permanova` and/or `permdisp`.

PERMANOVA and PERMDISP exclude samples (columns) specified in either `sample_remove` or `sample_project`. Projected samples are not used for testing as their PCA coordinates are derived from, and therefore depend on, the coordinates of non-projected samples.

Data read from working directory with SNPs as rows and samples as columns. Two alternative formats: (1) text file of SNPs by samples (file extension and column separators recognized automatically) read using `fread`; or (2) duet of `EIGENSTRAT` files (see `https://reich.hms.harvard.edu/software`) using `vroom_fwf`, including a genotype file of SNPs by samples (`*.geno`), and a sample file (`*.ind`) containing three vectors assigning individual samples to unique user-predefined groups (populations), sexes (or other user-defined descriptor) and alphanumeric identifiers. For `EIGENSTRAT`, vector `sample_group` assigns samples to groups retrievable from column 3 of file `*.ind`. SNPs with zero variance removed prior to SVD to optimize computation time and avoid undefined values if `scaling = "sd"` or `"drift"`.

Users can select subsets of samples or SNPs by introducing a vector including column numbers for samples (`sample_remove`) and/or row numbers for SNPs (`snp_remove`) to be removed from computations. Function stops if the final number of SNPs is 1 or 2. `EIGENSOFT` was conceived for the analysis of human genes and its SMARTPCA suite so accepts 22 (autosomal) chromosomes by default. If >22 chromosomes are provided and the internal parameter `numchrom` is not set to the target number chromosomes of interest, SMARTPCA automatically subsets chromosomes 1 to 22. In contrast, `smart_mva` accepts any number of autosomes with or without the sex chromosomes from an `EIGENSTRAT` file.

Value

Returns a list containing the following elements:

- pca.snp_loadingsDataframe of principal coefficients of SNPs. One set of coefficients per PCA axis computed.
- pca.eigenvaluesDataframe of eigenvalues, variance and cumulative variance explained. One eigenvalue per PCA axis computed.
- pca_sample_coordinatesDataframe showing PCA sample summary. Column *Group* assigns samples to groups. Column *Class* specifies if samples "Removed" from PCA or "Projected" onto PCA space. Sequence of additional columns shows principal components (coordinates) of samples in PCA space (1 column per PCA computed named PC1, PC2, ...).
- test_samplesDataframe showing test sample summary. Column *Group* assigns samples to tested groups. Column *Class* specifies if samples were used in, or removed from, testing (PERMANOVA and/or PERMDISP). Column *Sample_dispersion* shows dispersion of individual samples relative to spatial `"median"` or `"centroid"` used in PERMDISP.
- permanova.global_testList showing PERMANOVA table with degrees of freedom, sum of squares, mean sum of squares, *F* statistic, variance explained (*R2*) and *p* value.
- permanova.pairwise_testList showing PERMANOVA table with *F* statistic, variance explained (*R2*), *p* value and corrected *p* value per pair of groups.
- permdisp.global_testList showing PERMDISP table with degrees of freedoms, sum of squares, mean sum of squares, *F* statistic and *p* value.
- permdisp.pairwise_testList showing PERMDISP table with *F* statistic, *p* value and corrected *p* value per pair of groups. Obtained only if `pairwise = TRUE`.
- permdisp.biasString indicating if PERMDISP dispersion corrected for number of samples per group.
- permdisp.group_locationDataframe showing coordinates of spatial `"median"` or `"centroid"` per group in PERMDISP.
- test.pairwise_correctionString indicating type of correction for multiple testing in PERMANOVA and/or PERMDISP.
- test.permutation_numberNumber of permutations applied to obtain the distribution of *F* statistic of PERMANOVA and/or PERMDISP.
- test.permutation_seedNumber fixing random generator of permutations of PERMANOVA and/or PERMDISP for reproducibility of results.

See Also

smart_pca, smart_permanova, smart_permdisp

Examples

```
# Path to example genotype matrix "dataSNP"
pathToGenoFile = system.file("extdata", "dataSNP", package = "smartsnp")

# Assign 50 samples to each of two groups and colors
my_groups <- as.factor(c(rep("A", 50), rep("B", 50))); cols = c("red", "blue")

# Run PCA, PERMANOVA and PERMDISP
mvaR <- smart_mva(snp_data = pathToGenoFile, sample_group = my_groups)
mvaR$pca$pca.eigenvalues # extract PCA eigenvalues
mvaR$pca$pca.snp_loadings # extract principal coefficients (SNP loadings)
mvaR$pca$pca.sample_coordinates # extract PCA principal components (sample position in PCA space)

# plot PCA
plot(mvaR$pca$pca.sample_coordinates[,c("PC1","PC2")], cex = 2,
     pch = 19, col = cols[my_groups], main = "genotype smartpca")
legend("topleft", legend = levels(my_groups), cex = 1,
       pch = 19, col = cols, text.col = cols)

# Extract PERMANOVA table
mvaR$test$permanova.global_test

# Extract PERMDISP table
mvaR$test$permdisp.global_test # extract PERMDISP table

# Extract sample summary and dispersion of individual samples used in PERMDISP
mvaR$test$test_samples
```

---

smart_pca *Smart Principal Component Analysis*

Description

Compute Principal Component Analysis (PCA) for variable x sample genotype data including covariance (centered), correlation (z-score) and SMARTPCA scaling, and implements projection of ancient samples onto modern PCA space. SMARTPCA scaling controls for genetic drift when variables are bi-allelic genetic markers such as single nucleotide polymorphisms (SNP) following Patterson, Price and Reich (2006). Optimized to run fast single value decomposition for big datasets.

Arguments

snp_data	File name read from working directory. SNP = rows, samples = columns without row names or column headings. SNP values must be count data (no decimals

allowed). File extension detected automatically whether text or EIGENSTRAT. See details.

packed_data — Logical value for EIGENSTRAT, irrelevant for text data. Default packed_data = FALSE assumes uncompressed EIGENSTRAT. packed_data = TRUE for compressed or binary EIGENSTRAT (PACKENDANCESTRYMAP).

sample_group — Character or numeric vector assigning samples to groups. Coerced to factor.

sample_remove — Logical FALSE or numeric vector indicating column numbers (samples) to be removed from computations. Default sample_remove = FALSE keeps all samples.

snp_remove — Logical FALSE or numeric vector indicating row numbers (SNPs) to be removed from computations. Default snp_remove = FALSE keeps all SNPs. See details.

missing_value — Number 9 or string NA indicating missing value. Default missing_value = 9 as in EIGENSTRAT. If no missing values present, no effect on computation.

missing_impute — String handling missing values. Default missing_impute = "mean" replaces missing values of each SNP by mean of non-missing values across samples. missing_impute = "remove" removes SNPs with at least one missing value. If no missing values present, no effect on computation.

scaling — String. Default scaling = "drift" scales SNPs to control for expected allele frequency dispersion caused by genetic drift (SMARTPCA). scaling = "center" for centering (covariance-based PCA). scaling = "sd" for centered SNPs divided by standard deviation (correlation-based PCA). scaling = "none" for no scaling. See details.

program_svd — String indicating R package computing single value decomposition (SVD). Default program_svd = "Rspectra" for svds. program_svd = "bootSVD" for fastSVD. See details.

pc_axes — A numeric value. If program_svd = "Rspectra" this argument indicates number of PCA axes computed starting with PCA axis 1. Default pc_axes = 2 computes PCA axes 1 and 2. No effect on computation if program_svd = "bootSVD" since all PCA axes are computed.

sample_project — Numeric vector indicating column numbers (ancient samples) projected onto (modern) PCA space. Default sample_project = FALSE indicates no samples will be used for projection. See details.

pc_project — Numeric vector indicating the ranks of the PCA axes ancient samples are projected onto. Default pc_ancient = c(1, 2) for PCA axes 1 and 2. If program_svd = "RSpectra", length(pc_ancient) must be smaller than or equal to pc_axes. No effect on computation, if no ancient samples present.

Details

PCA is a rigid rotation of a Cartesian coordinate system (samples = points, axes = variables or SNPs) that maximizes the dispersion of points along a new system of axes (Pearson 1901; Hotelling 1933; Jolliffe 2002). In rotated space (ordination), axes are principal axes (PCA axes), eigenvalues measure variance explained, and principal coefficients measure importance of SNPs (eigenvectors), principal components are coordinates of samples (i.e., linear combinations of scaled variables weighted by eigenvectors). Principal coefficients are direction cosines between original

and PCA axes (Legendre & Legendre 2012). PCA can be computed by `eigenanalysis` or, as implemented here, single value decomposition (SVD).

SNPs can be scaled in four different ways prior to SVD: (1) no scaling; (2) covariance: SNPs `centered` such that *M(i,j)* = *C(i,j)* minus *mean(j)*) where *C(i,j)* is the number of variant alleles for SNP *j* and sample *i*, and *M(i,j)* is the `centered` value of each data point; (3) correlation (z-scores): SNPs `centered` then divided by standard deviation *sd(j)*, (4) SMARTPCA: SNPs `centered` then divided by *sqrt(p(j)(1-p(j)))*, where *p(j)* equals *mean(j)* divided by *2*, quantifies the underlying allele frequency (autosomal chromosomes) and conceptualizes that SNP frequency changes at rate proportional to *sqrt(p(j)(1-p(j)))* per generation due to genetic drift (Patterson, Price and Reich 2006). SMARTPCA standardization results in all SNPs that comply with Hardy-Weinberg equilibrium having identical variance. SMARTPCA (Patterson, Price and Reich 2006) and `EIGENSTRAT` (Price, Patterson, Plenge, Weinblatt, Shadick and Reich 2006) are the computing suites of software `EIGENSOFT` (`https://reich.hms.harvard.edu/software`).

`svds` runs single value decomposition much faster than `fastSVD`. With `svds`, `pc_axes` indicates number of eigenvalues and eigenvectors computed starting from PCA axis 1. `fastSVD` computes all eigenvalues and eigenvectors. Eigenvalues calculated from singular values divided by number of samples minus 1. If number of samples equals number of SNPS, `fastSVD` prints message alert that no computing efficiency is achieved for square matrices.

Ancient samples (with many missing values) can be projected onto modern PCA space derived from modern samples. Following Nelson Taylor and MacGregor (1996), the projected coordinates of a given ancient sample equal the slope coefficient of linear fit through the origin of (scaled) non-missing SNP values of that sample (response) versus principal coefficients of same SNPs in modern samples. Number of projected coordinates per ancient sample given by `length(pc_ancient)`. With `svds`, `pc_axes` must be larger or equal to `length(pc_ancient)`.

Data read from working directory with SNPs as rows and samples as columns. Two alternative formats: (1) text file of SNPs by samples (file extension and column separators recognized automatically) read using `fread`; or (2) duet of `EIGENSTRAT` files (see `https://reich.hms.harvard.edu/software`) using `vroom_fwf`, including a genotype file of SNPs by samples (`*.geno`), and a sample file (`*.ind`) containing three vectors assigning individual samples to unique user-predefined groups (populations), sexes (or other user-defined descriptor) and alphanumeric identifiers. For `EIGENSTRAT`, vector `sample_group` assigns samples to groups retrievable from column of file `*.ind`. SNPs with zero variance removed prior to SVD to optimize computation time and avoid undefined values if `scaling = "sd"` or `"drift"`.

Users can select subsets of samples or SNPs by introducing a vector including column numbers for samples (`sample_remove`) and/or row numbers for SNPs (`snp_remove`) to be removed from computations. Function stops if the final number of SNPs is 1 or 2. `EIGENSOFT` was conceived for the analysis of human genes and its SMARTPCA suite so accepts 22 (autosomal) chromosomes by default. If >22 chromosomes are provided and the internal parameter `numchrom` is not set to the target number chromosomes of interest, SMARTPCA automatically subsets chromosomes 1 to 22. In contrast, `smart_pca` accepts any number of autosomes with or without the sex chromosomes from an `EIGENSTRAT` file.

Value

Returns a list containing the following elements:

- `pca.snp_loadings` Dataframe of principal coefficients of SNPs. One set of coefficients per PCA axis computed.
- `pca.eigenvalues` Dataframe of eigenvalues, variance and cumulative variance explained. One eigenvalue per PCA axis computed.
- `pca_sample_coordinates` Dataframe showing PCA sample summary. Column *Group* assigns samples to groups. Column *Class* specifies if samples "Removed" from PCA or "Projected" onto PCA space. Sequence of additional columns shows principal components (coordinates) of samples in PCA space (1 column per PCA computed named PC1, PC2, ...).

References

Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417-441.

Jolliffe, I.T. (2002) Principal Component Analysis (Springer, New York, USA).

Legendre, P. & L. F. J. Legendre (2012). Numerical ecology. Developments in environmental modelling (Elsevier, Oxford, UK).

Nelson, P.R.C., P.A. Taylor, and J.F. MacGregor (1996) Missing data methods in PCA and PLS: score calculations with incomplete observations. Chemometrics and Intelligent Laboratory Systems, 35, 45-65.

Patterson, N.J., A. L. Price and D. Reich (2006) Population structure and eigenanalysis. PLoS Genetics, 2, e190.

Pearson, K. (1901) On lines and planes of closest fit to systems of points in space. Philosophical Magazine, 2, 559-572.

Price, A.L., N.J. Patterson, R.M. Plenge, M.E. Weinblatt, N.A. Shadick and David Reich (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nature Genetics, 38, 904-909.

See Also

`fastSVD` (package **bootSVD**), `foreach` (package **foreach**), `fread` (package **data.table**), `rowVars` (package **Rfast**), `svds` (package **RSpectra**), `vroom_fwf` (package **vroom**)

Examples

```
# Path to example genotype matrix "dataSNP"
pathToGenoFile = system.file("extdata", "dataSNP", package = "smartsnp")

# Example 1: modern samples
#assign 50 samples to each of two groups and colors
my_groups <- c(rep("A", 50), rep("B", 50)); cols = c("red", "blue")
#run PCA with truncated SVD (PCA 1 x PCA 2)
pcaR1 <- smart_pca(snp_data = pathToGenoFile, sample_group = my_groups)
pcaR1$pca.eigenvalues # extract eigenvalues
pcaR1$pca.snp_loadings # extract principal coefficients (SNP loadings)
pcaR1$pca.sample_coordinates # extract principal components (sample position in PCA space)
#plot PCA
plot(pcaR1$pca.sample_coordinates[,c("PC1","PC2")], cex = 2,
     pch = 19, col = cols[as.factor(my_groups)], main = "genotype smartpca")
legend("topleft", legend = levels(as.factor(my_groups)), cex =1,
     pch = 19, col = cols, text.col = cols)

# Example 2: modern and ancient samples (ancient samples projected onto modern PCA space)
#assign samples 1st to 10th per group to ancient
my_ancient <- c(1:10, 51:60)
#run PCA with truncated SVD (PCA 1 x PCA 2)
pcaR2 <- smart_pca(snp_data = pathToGenoFile, sample_group = my_groups, sample_project = my_ancient)
pcaR2$pca.eigenvalues # extract eigenvalues
pcaR2$pca.snp_loadings # extract principal coefficients (SNP loading)
pcaR2$pca.sample_coordinates # extract principal components (sample position in PCA space)
#assign samples to groups (A, ancient, B) and colors
my_groups[my_ancient] <- "ancient"; cols = c("red", "black", "blue")
#plot PCA
plot(pcaR2$pca.sample_coordinates[,c("PC1","PC2")],
     cex = 2, col = cols[as.factor(my_groups)], pch = 19, main = "genotype smartpca")
legend("topleft", legend = levels(as.factor(my_groups)),  cex = 1,
     pch = 19, col = cols, text.col = cols)
```

`smart_permanova` *Smart Permutational Multivariate Analysis of Variance*

Description

Computes Permutational Multivariate Analysis of Variance (PERMANOVA) for testing differences in group location using multivariate data. Variance partitioning computed on a sample-by-sample triangular matrix obtained from variable-by-sample data following Anderson (2001). Calculates a range of inter-sample distances, similarities and dissimilarities. Includes control for genetic drift for bi-allelic genetic markers such as single nucleotide polymorphisms (SNP) following Patterson, Price and Reich (2006) that can be combined with SMART Principal Component Analysis (PCA). Optimized to run fast matrix building and permutations for big datasets in ecological, evolutionary and genomic research.

Arguments

snp_data	File name read from working directory. SNP = rows, samples = columns without row names or column headings. SNP values must be count data (no decimals allowed). File extension detected automatically whether text or EIGENSTRAT. See details.
packed_data	Logical value for EIGENSTRAT, irrelevant for text data. Default packed_data = FALSE assumes uncompressed EIGENSTRAT. packed_data = TRUE for compressed or binary EIGENSTRAT (PACKENDANCESTRYMAP).
sample_group	Character or numeric vector assigning samples to groups. Coerced to factor.
sample_remove	Logical FALSE or numeric vector indicating column numbers (samples) to be removed from computations. Default sample_remove = FALSE keeps all samples.
snp_remove	Logical FALSE or numeric vector indicating row numbers (SNPs) to be removed from computations. Default snp_remove = FALSE keeps all SNPs. See details.
missing_value	Number 9 or string NA indicating missing value. Default missing_value = 9 as in EIGENSTRAT. If no missing values present, no effect on computation.
missing_impute	String handling missing values. Default missing_impute = "mean" replaces missing values of each SNP by mean of non-missing values across samples. missing_impute = "remove" removes SNPs with at least one missing value. If no missing values present, no effect on computation.
scaling	String. Default scaling = "drift" scales SNPs to control for expected allele frequency dispersion caused by genetic drift (SMARTPCA). scaling = "center" for centering (covariance-based PCA). scaling = "sd" for centered SNPs divided by standard deviation (correlation-based PCA). scaling = "none" for no scaling. See details.
sample_distance	Type of inter-sample proximity computed (distance, similarity, dissimilarity). Default is Euclidean distance. See details.
program_distance	A string value indicating R package to estimate proximities between pairs of samples. Default program_distance = "Rfast" uses function Dist; program_distance = "vegan" uses vegdist. See details.
target_space	String. Default target_space = "multidimensional" applies PERMANOVA to sample-by-sample triangular matrix computed from variable-by-sample data, pc_axes has no effect on computation. target_space = "pca" applies PERMANOVA to sample-by-sample data in PCA space, pc_axes determines number of PCA axes for testing.
pc_axes	Number of PCA axes computed always starting with PCA axis 1. Default pc_axes = 2 computes PCA axes 1 and 2 if target_space = "pca". No effect on computation if target_space = "multidimensional".
pairwise	Logical. Default pairwise = FALSE computes global test. pairwise = TRUE computes global and pairwise tests.
pairwise_method	String specifying type of correction for multiple testing. Default "holm". See details.

permutation_n Number of permutations resulting in PERMANOVA test *p value*. Default 9999.

permutation_seed
Number fixing random generator of permutations. Default 1.

Details

PERMANOVA is a form of linear modelling that partitions variation in a triangular matrix of inter-sample proximities obtained from variable-by-sample data. Uses permutations to estimate the probability of observed group differences in SNP composition given a null hypothesis of no differences between groups (Anderson 2001). Proximity between samples can be any type of distance, similarity or dissimilarity. Original acronym NPMANOVA (Non-Parametric MANOVA) replaced with PERMANOVA (Anderson 2004, 2017).

Univariate ANOVA captures differences in mean and variance referred to as location and dispersion in PERMANOVA's multivariate context (Anderson & Walsh 2013, Warton, Wright and Wang 2012). To attribute group differences to location (position of sample groups) and/or dispersion (spread of sample groups), PERMANOVA must be combined with PERMDISP as implemented through smart_permdisp.

Function smart_permanova uses adonis to fit formula snp_eucli ~ sample_group, where snp_eucli is the sample-by-sample triangular matrix in Principal Coordinate Analysis (Gower 1966) space. Current version restricted to one-way designs (one categorical predictor) though PERMANOVA can handle >1 crossed and/or nested factors (Anderson 2001) and continuous predictors (McArdle & Anderson 2001). If >2 sample groups tested, pairwise = TRUE allows pairwise testing and correction for multiple testing by holm (Holm) [default], hochberg (Hochberg), hommel (Hommel), bonferroni (Bonferroni), BY (Benjamini-Yekuieli), BH (Benjamini-Hochberg) or fdr (False Discovery Rate).

For big data, Dist builds sample-by-sample triangular matrix much faster than vegdist. Dist computes proximities euclidean, manhattan, canberra1, canberra2, minimum, maximum, minkowski, bhattacharyya, hellinger, kullback_leibler and jensen_shannon. vegdist computes manhattan, euclidean, canberra, clark, bray, kulczynski, jaccard, gower, altGower, morisita, horn, mountford, raup, binomial, chao, cao and mahalanobis. Euclidean distance required for SMART-PCA scaling.

sample_remove should include both samples removed from PCA and ancient samples projected onto PCA space (if any).

Data read from working directory with SNPs as rows and samples as columns. Two alternative formats: (1) text file of SNPs by samples (file extension and column separators recognized automatically) read using fread; or (2) duet of EIGENSTRAT files (see https://reich.hms.harvard.edu/software) using vroom_fwf, including a genotype file of SNPs by samples (*.geno), and a sample file (*.ind) containing three vectors assigning individual samples to unique user-predefined groups (populations), sexes (or other user-defined descriptor) and alphanumeric identifiers. For EIGENSTRAT, vector sample_group assigns samples to groups retrievable from column 3 of file *.ind. SNPs with zero variance removed prior to SVD to optimize computation time and avoid

undefined values if `scaling = "sd"` or `"drift"`.

Users can select subsets of samples or SNPs by introducing a vector including column numbers for samples (`sample_remove`) and/or row numbers for SNPs (`snp_remove`) to be removed from computations. Function stops if the final number of SNPs is 1 or 2. `EIGENSOFT` was conceived for the analysis of human genes and its SMARTPCA suite so accepts 22 (autosomal) chromosomes by default. If >22 chromosomes are provided and the internal parameter `numchrom` is not set to the target number chromosomes of interest, SMARTPCA automatically subsets chromosomes 1 to 22. In contrast, `smart_permanova` accepts any number of autosomes with or without the sex chromosomes from an `EIGENSTRAT` file.

Value

Returns a list containing the following elements:

- permanova.samplesDataframe showing sample summary. Column *Group* assigns samples to tested groups. Column *Class* specifies if samples were used in, or removed from, testing.
- permanova.global_testList showing table with degrees of freedom, sum of squares, mean sum of squares, *F* statistic, variance explained (*R2*) and *p* value.
- permanova.pairwise_testList showing table *F* statistic, variance explained (*R2*), *p* value and corrected *p* value per pair of groups. Obtained only if `pairwise = TRUE`.
- permanova.pairwise_correctionString indicating type of correction for multiple testing.
- permanova.permutation_numberNumber of permutations applied to obtain the distribution of *p value*.
- permanova.permutation_seedNumber fixing random generator of permutations for reproducibility of results.

References

Anderson, M. J. (2001) A new method for non-parametric multivariate analysis of variance. Austral Ecology, 26, 32-46.
Anderson, M. J. (2004). PERMANOVA_2factor: a FORTRAN computer program for permutational multivariate analysis of variance (for any two-factor ANOVA design) using permutation tests (Department of Statistics, University of Auckland, New Zealand).
Anderson, M. J. & D. C. I. Walsh (2013) PERMANOVA, ANOSIM, and the Mantel test in the face of heterogeneous dispersions: What null hypothesis are you testing? Ecological Monographs, 83, 557-574.
Gower, J. C. (1966) Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika, 53, 325-338.
McArdle, B. H. & M. J. Anderson (2001) Fitting multivariate models to community data: a comment on distance-based redundancy analysis. Ecology, 82, 290-297.
Patterson, N., A. L. Price and D. Reich (2006) Population structure and eigenanalysis. PLoS Genetics, 2, e190.
Warton, D. I., S. T. Wright and Y. Wang (2012) Distance-based multivariate analyses confound location and dispersion effects. Methods in Ecology and Evolution, 3, 89-101.

See Also

adonis (package **vegan**), Dist (package **Rfast**), fread (package **data.table**), vegdist (package **vegan**), vroom_fwf (package **vroom**)

Examples

```
# Path to example genotype matrix "dataSNP"
pathToGenoFile = system.file("extdata", "dataSNP", package = "smartsnp")

# Assign 50 samples to each of two groups
my_groups <- as.factor(c(rep("A", 50), rep("B", 50)))

# Run PERMANOVA
permanovaR <- smart_permanova(snp_data = pathToGenoFile, sample_group = my_groups)

# Extract summary table assigning samples to groups
permanovaR$permanova.samples

# Extract PERMANOVA table
permanovaR$permanova.global_test

# Plot means of squares per group
#run pca with truncated SVD (PCA 1 x PCA 2)
pcaR1 <- smart_pca(snp_data = pathToGenoFile, sample_group = my_groups)
#compute Euclidean inter-sample distances in PCA space (triangular matrix)
snp_eucli <- vegan::vegdist(pcaR1$pca.sample_coordinates[,c("PC1","PC2")], method = "euclidean")
#run PERMANOVA
permanova <- vegan::adonis(formula = snp_eucli ~ my_groups, permutations = 9999)
#extract meanSqs (groups versus residuals)
meanSqs <- as.matrix(t(permanova$aov.tab$MeanSqs[1:2]))
colnames(meanSqs) <- c("Groups", "Residuals")
#two horizontal plots
oldpar <- par(mfrow = c(2,1), oma = c(0,5,0.1,0.1), lwd = 2)
barplot(meanSqs, horiz = TRUE, main = "PERMANOVA mean of squares",
  cex.names = 2, cex.main = 2, col = c("grey40"))
#run ANOSIM
anosimD <- vegan::anosim(snp_eucli, my_groups, permutations = 999)
#remove outputs for clean plotting
#anosimD[2] <- ""; anosimD[5] <- ""
par(mar = c(5, 0.1, 3.5, 0.1))
plot(anosimD, xlab = "", ylab = "distance/similarity ranks",
  main = "Inter-sample proximity ranks", cex.main =2, cex.axis = 2,
  col = c("cyan", "red", "blue"))
par(oldpar)
```

---

smart_permdisp *Smart Permutational Multivariate Analysis of Dispersion*

Description

Computes Permutational Multivariate Analysis of Dispersion (PERMDISP) in group dispersion using multivariate data. Variance partitioning computed on a sample-by-sample triangular matrix obtained from variable-by-sample data following Anderson (2006). Calculates a range of inter-sample distances, similarities and dissimilarities. Includes control for genetic drift for bi-allelic genetic markers including single nucleotide polymorphisms (SNP) following Patterson, Price and Reich (2006) that can be combined with SMART Principal Component Analysis (PCA). Optimized to run fast matrix building and permutations for big datasets in ecological, evolutionary and genomic research.

Arguments

`snp_data`	File name read from working directory. SNP = rows, samples = columns without row names or column headings. SNP values must be count data (no decimals allowed). File extension detected automatically whether text or `EIGENSTRAT`. See details.
`packed_data`	Logical value for `EIGENSTRAT`, irrelevant for text data. Default `packed_data = FALSE` assumes uncompressed `EIGENSTRAT`. `packed_data = TRUE` for compressed or binary `EIGENSTRAT` (`PACKENDANCESTRYMAP`).
`sample_group`	Character or numeric vector assigning samples to groups. Coerced to factor.
`sample_remove`	Logical `FALSE` or numeric vector indicating column numbers (samples) to be removed from computations. Default `sample_remove = FALSE` keeps all samples.
`snp_remove`	Logical `FALSE` or numeric vector indicating row numbers (SNPs) to be removed from computations. Default `snp_remove = FALSE` keeps all SNPs. See details.
`missing_value`	Number `9` or string `NA` indicating missing value. Default `missing_value = 9` as in `EIGENSTRAT`. If no missing values present, no effect on computation.
`missing_impute`	String handling missing values. Default `missing_impute = "mean"` replaces missing values of each SNP by mean of non-missing values across samples. `missing_impute = "remove"` removes SNPs with at least one missing value. If no missing values present, no effect on computation.
`scaling`	String. Default `scaling = "drift"` scales SNPs to control for expected allele frequency dispersion caused by genetic drift (SMARTPCA). `scaling = "center"` for `centering` (covariance-based PCA). `scaling = "sd"` for `centered` SNPs divided by standard deviation (correlation-based PCA). `scaling = "none"` for no scaling. See details.
`sample_distance`	Type of inter-sample proximity computed (distance, similarity, dissimilarity). Default is `Euclidean distance`. See details.
`program_distance`	A string value indicating R package to estimate proximities between pairs of samples. Default `program_distance = "Rfast"` uses function `Dist`; `program_distance = "vegan"` uses `vegdist`. See details.
`target_space`	String. Default `target_space = "multidimensional"` applies PERMANOVA to sample-by-sample triangular matrix computed from variable-by-sample data,

pc_axes has no effect on computation. target_space = "pca" applies PERMANOVA to sample-by-sample data in PCA space, pc_axes determines number of PCA axes for testing.

pc_axes — Number of PCA axes computed always starting with PCA axis 1. Default pc_axes = 2 computes PCA axes 1 and 2 if target_space = "pca". No effect on computation if target_space = "multidimensional".

pairwise — Logical. Default pairwise = FALSE computes global test. pairwise = TRUE computes global and pairwise tests.

pairwise_method — String specifying type of correction for multiple testing. Default "holm". See details.

permutation_n — Number of permutations resulting in PERMDISP test *p value*. Default 9999.

permutation_seed — Number fixing random generator of permutations. Default 1.

dispersion_type — String indicating quantification of group dispersion whether relative to spatial "median" or "centroid". Default "median". See details.

samplesize_bias — Logical. samplesize_bias = TRUE for dispersion weighted by number of samples per group. Default pairwise = FALSE for no weighting. See details.

Details

PERMDISP is a form of homoscedasticity test analogous to univariate Levene's (1960) and, more closely, Brown & Forsythe's (1974) tests. Applies PERMANOVA test (Anderson 2001) for differences in Euclidean dispersion among groups (Anderson 2006). Proximity between samples can be any type of distance, similarity or dissimilarity. Group dispersion estimated relative to group centroids (central point) or to spatial (geometric) medians (point minimizing distance to group samples) in Principal Coordinate Analysis (PCoA, Gower 1966) space. Acronym PERMDISP originates from Marti Anderson's FORTRAN program (Anderson 2004).

Control for unequal number of samples among groups optionally done by weighting sample distance to group spatial *median* or *centroid* by *sqrt(n/(n-1))* (O'Neill & Mathews 2000), and the null hypothesis then being tested changes from *d_1 = d_2 = ... = d_t* (balanced design) to *((n_1-1)/n_1) x d_1 = ((n_2-1)/n_2) x d_2 = ... = ((n_t-1)/n_t) x d_t* (unbalanced design) where *d* represents dispersion of groups *1* to *t*, and *n* represents number of samples per group. To attribute group differences to location (position of sample groups) and/or dispersion (spread of sample groups), PERMDISP must be combined with PERMANOVA as implemented through smart_permanova.

smart_permdisp uses betadisper to estimate an ANOVA *F* statistic and group dispersions using formula snp_eucli ~ sample_group, where snp_eucli is the sample-by-sample triangular matrix in PCoA space. If >2 sample groups tested, pairwise = TRUE allows pairwise testing and correction for multiple testing by holm (Holm) [default], hochberg (Hochberg), hommel (Hommel), bonferroni (Bonferroni), BY (Benjamini-Yekuieli), BH (Benjamini-Hochberg) or fdr (False Discovery Rate).

For big data, Dist builds sample-by-sample triangular matrix much faster than vegdist. Dist computes proximities euclidean, manhattan, canberra1, canberra2, minimum, maximum, minkowski,

bhattacharyya, hellinger, kullback_leibler and jensen_shannon. vegdist computes manhattan, euclidean, canberra, clark, bray, kulczynski, jaccard, gower, altGower, morisita, horn, mountford, raup, binomial, chao, cao and mahalanobis. Euclidean distance required for SMART-PCA scaling.

sample_remove should include both samples removed from PCA and ancient samples projected onto PCA space (if any).

Data read from working directory with SNPs as rows and samples as columns. Two alternative formats: (1) text file of SNPs by samples (file extension and column separators recognized automatically) read using fread; or (2) duet of EIGENSTRAT files (see https://reich.hms.harvard.edu/software) using vroom_fwf, including a genotype file of SNPs by samples *.geno, and a sample file (*.ind) containing three vectors assigning individual samples to unique user-predefined groups (populations), sexes (or other user-defined descriptor) and alphanumeric identifiers. For EIGENSTRAT, vector sample_group assigns samples to groups retrievable from column 3 of file *.ind. SNPs with zero variance removed prior to SVD to optimize computation time and avoid undefined values if scaling = "sd" or "drift".

Users can select subsets of samples or SNPs by introducing a vector including column numbers for samples (sample_remove) and/or row numbers for SNPs (snp_remove) to be removed from computations. Function stops if the final number of SNPs is 1 or 2. EIGENSOFT was conceived for the analysis of human genes and its SMARTPCA suite so accepts 22 (autosomal) chromosomes by default. If >22 chromosomes are provided and the internal parameter numchrom is not set to the target number chromosomes of interest, SMARTPCA automatically subsets chromosomes 1 to 22. In contrast, smart_permdisp accepts any number of autosomes with or without the sex chromosomes from an EIGENSTRAT file.

Value

Returns a list containing the following elements:

- permdisp.samplesDataframe showing sample summary. Column *Group* assigns samples to tested groups. Column *Class* specifies if samples were used in, or removed from, testing (PERMDISP). Column *Sample_dispersion* shows dispersion of individual samples relative to spatial "median" or "centroid".
- permdisp.biasString indicating if PERMDISP dispersions corrected for number of samples per group.
- permdisp.group_locationDataframe showing coordinates of spatial "median" or "centroid" per group.
- permdisp.global_testList showing table with degrees of freedom, sum of squares, mean sum of squares, F statistic and p value.
- permdisp.pairwise_testList showing table with F statistic, p value and corrected p value per pair of groups. Obtained only if pairwise = TRUE.
- permdisp.pairwise_correctionString indicating type of correction for multiple testing.
- permdisp.permutation_numberNumber of permutations applied to obtain the distribution of F statistic.

- permdisp.permutation_seedNumber fixing random generator of permutations for reproducibility of results.

References

Anderson, M. J. (2001) A new method for non-parametric multivariate analysis of variance. Austral Ecology, 26, 32-46.
Anderson, M. J. (2004). PERMANOVA_2factor: a FORTRAN computer program for permutational multivariate analysis of variance (for any two-factor ANOVA design) using permutation tests (Department of Statistics, University of Auckland, New Zealand).
Anderson, M. J. & D. C. I. Walsh (2013) PERMANOVA, ANOSIM, and the Mantel test in the face of heterogeneous dispersions: What null hypothesis are you testing? Ecological Monographs, 83, 557-574.
Gower, J. C. (1966) Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika, 53, 325-338.
McArdle, B. H. & M. J. Anderson (2001) Fitting multivariate models to community data: a comment on distance-based redundancy analysis. Ecology, 82, 290-297.
Patterson, N., A. L. Price and D. Reich (2006) Population structure and eigenanalysis. PLoS Genetics, 2, e190.
Warton, D. I., S. T. Wright and Y. Wang (2012) Distance-based multivariate analyses confound location and dispersion effects. Methods in Ecology and Evolution, 3, 89-101.

See Also

`adonis` (package **vegan**), `Dist` (package **Rfast**), `fread` (package **data.table**), `vegdist` (package **vegan**), `vroom_fwf` (package **vroom**)

Examples

```
# Path to example genotype matrix "dataSNP"
pathToGenoFile = system.file("extdata", "dataSNP", package = "smartsnp")

# Assign 50 samples to each of two groups and colours
my_groups <- as.factor(c(rep("A", 50), rep("B", 50))); cols = c("red", "blue")

# Run PERMDISP
permdispR <- smart_permdisp(snp_data = pathToGenoFile, sample_group = my_groups)

# Extract summary table assigning samples to groups and dispersion of individual samples
permdispR$permdisp.samples

# Extract PERMDISP table
permdispR$permdisp.global_test

# Plot sample distances to group central medians
#run pca with truncated SVD (PCA 1 x PCA 2)
pcaR1 <- smart_pca(snp_data = pathToGenoFile, sample_group = my_groups)
#compute Euclidean inter-sample distances in PCA space (triangular matrix)
snp_eucli <- vegan::vegdist(pcaR1$pca.sample_coordinates[,c("PC1","PC2")], method = "euclidean")
#calculate spatial medians
disMed <- vegan::betadisper(d = snp_eucli, group = my_groups); disMed
```

```
#plot
oldpar <- par(mar = c(4, 4, 5, 0.1), lwd = 2)
boxplot(disMed, las =2, cex.axis = 2, cex.main = 1.5, horizontal = TRUE, varwidth = TRUE,
   col = cols, xlab = "", ylab = "", main = "Sample distance to group spatial medians")
par(oldpar)
```

Index