Package ‘PCAmatchR’
January 10, 2021

Title  Match Cases to Controls Based on Genotype Principal Components
Version  0.3.0
Maintainer  Derek W. Brown <derek.brown@nih.gov>
Description  Matches cases to controls based on genotype principal components (PC).
In order to produce better results, matches are based on the weighted
distance of PCs where the weights are equal to the % variance explained
by that PC. A weighted Mahalanobis distance metric (Kidd et al. (1987)
<DOI:10.1016/0031-3203(87)90066-5>) is used to determine matches.
License  MIT + file LICENSE
URL  https://github.com/machiela-lab/PCAmatchR
BugReports  https://github.com/machiela-lab/PCAmatchR/issues
Encoding  UTF-8
LazyData  true
Depends  R (>= 2.10)
Suggests  optmatch, testthat, knitr, rmarkdown
VignetteBuilder  knitr
RoxygenNote  7.1.1
NeedsCompilation  no
Author  Derek W. Brown [aut, cre],
        Mitchel J. Machiela [aut],
        Timothy A. Myers [ctb],
        NCI [cph, fnd]
Repository  CRAN
Date/Publication  2021-01-10 15:20:22 UTC

R topics documented:

  eigenvalues_1000G ................................................................. 2
  eigenvalues_all_1000G ........................................................... 2
  match_maker ................................................................. 3
  PCs_1000G ................................................................. 5
  plot_maker ................................................................. 6
### eigenvalues_1000G

**First 20 eigenvalues of 2504 individuals from the 1000 Genome Project**

**Description**

A sample dataset containing the first 20 eigenvalues calculated from 2504 individuals in the Phase 3 data release of the 1000 Genomes Project. The principal component analysis was conducted using PLINK.

**Usage**

```r
eigenvalues_1000G
```

**Format**

A data frame with 20 rows and 1 variable:

- **eigen_values** calculated eigenvalues

**Source**

Machiela Lab

**Examples**

```r
eigenvalues_1000G
geno_values <- eigenvalues_1000G
values <- c(geno_values)$eigen_values
```

---

### eigenvalues_all_1000G

**All eigenvalues of 2504 individuals from the 1000 Genome Project**

**Description**

A sample dataset containing all the eigenvalues calculated from 2504 individuals in the Phase 3 data release of the 1000 Genomes Project. The principal component analysis was conducted using PLINK.

**Usage**

```r
eigenvalues_all_1000G
```
Feature

A data frame with 2504 rows and 1 variable:

- **eigen_values** calculated eigenvalues

Source

Machiela Lab

Examples

```r
eigenvalues_all_1000G
genome_values <- eigenvalues_all_1000G
values <- c(genome_values)$eigen_values
```

Description

Weighted matching of controls to cases using PCA results.

Usage

```r
match_maker(
  PC = NULL,
  eigen_value = NULL,
  data = NULL,
  ids = NULL,
  case_control = NULL,
  num_controls = 1,
  num_PCs = NULL,
  eigen_sum = NULL,
  exact_match = NULL,
  weight_dist = TRUE,
  weights = NULL
)
```

Arguments

- **PC** Individual level principal component.
- **eigen_value** Computed eigenvalue for each PC. Used as the numerator to calculate the percent variance explained by each PC.
- **data** Dataframe containing id and case/control status. Optionally includes covariate data for exact matching.
**match_make**

- **ids**
  The unique id variable contained in both "PC" and "data."

- **case_control**
  The case control status variable.

- **num_controls**
  The number of controls to match to each case. Default is 1:1 matching.

- **num_PCs**
  The total number of PCs calculated within the PCA. Can be used as the denominator to calculate the percent variance explained by each PC. Default is 1000.

- **eigen_sum**
  The sum of all possible eigenvalues within the PCA. Can be used as the denominator to calculate the percent variance explained by each PC.

- **exact_match**
  Optional variables contained in the dataframe on which to perform exact matching (i.e. sex, race, etc.).

- **weight_dist**
  When set to true, matches are produced based on PC weighted Mahalanobis distance. Default is TRUE.

- **weights**
  Optional user defined weights used to compute the weighted Mahalanobis distance metric.

**Value**

A list of matches and weights.

**Examples**

```r
# Create PC data frame by subsetting provided example dataset
pcs <- as.data.frame(PCs_1000G[,c(1,5:24)])
# Create eigenvalues vector using example dataset
eigen_vals <- c(eigenvalues_1000G)$eigen_values
# Create full eigenvalues vector using example dataset
all_eigen_vals<- c(eigenvalues_all_1000G)$eigen_values
# Create Covariate data frame
cov_data <- PCs_1000G[,c(1:4)]
# Generate a case status variable using ESN 1000 Genome population
cov_data$case <- ifelse(cov_data$pop=="ESN", c(1), c(0))
# With 1 to 1 matching
if(!requireNamespace("optmatch", quietly = TRUE)){
  match_make(PC = pcs,
             eigen_value = eigen_vals,
             data = cov_data,
             ids = c("sample"),
             case_control = c("case"),
             num_controls = 1,
             eigen_sum = sum(all_eigen_vals),
             weight_dist=TRUE)
}
```
**Description**

A sample dataset containing information about population, gender, and the first 20 principal components calculated from 2504 individuals in the Phase 3 data release of the 1000 Genomes Project. The principal component analysis was conducted using PLINK.

**Usage**

PCs_1000G

**Format**

A data frame with 2504 rows and 24 variables:

- **sample**: sample ID number
- **pop**: three letter designation of 1000 Genomes reference population
- **super_pop**: three letter designation of 1000 Genomes reference super population
- **gender**: gender of individual
- **PC1**: principal component 1
- **PC2**: principal component 2
- **PC3**: principal component 3
- **PC4**: principal component 4
- **PC5**: principal component 5
- **PC6**: principal component 6
- **PC7**: principal component 7
- **PC8**: principal component 8
- **PC9**: principal component 9
- **PC10**: principal component 10
- **PC11**: principal component 11
- **PC12**: principal component 12
- **PC13**: principal component 13
- **PC14**: principal component 14
- **PC15**: principal component 15
- **PC16**: principal component 16
- **PC17**: principal component 17
- **PC18**: principal component 18
- **PC19**: principal component 19
- **PC20**: principal component 20
plot_maker

Function to plot matches from match_maker output

**Description**

Function to plot matches from match_maker output

**Usage**

```r
plot_maker(
  data = NULL,
  x_var = NULL,
  y_var = NULL,
  case_control = NULL,
  line = T,
  ...
)
```

**Arguments**

- `data`: match_maker output
- `x_var`: Principal component 1
- `y_var`: Principal component 2
- `case_control`: Case or control status
- `line`: draw line
- `...`: Arguments passed to plot

**Value**

None

---

**Source**

[https://www.internationalgenome.org](https://www.internationalgenome.org)

**Examples**

```r
head(PCs_1000G)
genome_PC <- PCs_1000G
# Create PCs
  PC <- as.data.frame(genome_PC[,c(1,5:24)])
  head(PC)
```
Examples

```r
# run match_maker()
# Create PC data frame by subsetting provided example dataset
pcs <- as.data.frame(PCs_1000G[,c(1,5:24)])
# Create eigenvalues vector using example dataset
eigen_vals <- c(eigenvalues_1000G)$eigen_values
# Create full eigenvalues vector using example dataset
all_eigen_vals<- c(eigenvalues_all_1000G)$eigen_values
# Create Covariate data frame
cov_data <- PCs_1000G[,c(1:4)]
# Generate a case status variable using ESN 1000 Genome population
cov_data$case <- ifelse(cov_data$pop=="ESN", c(1), c(0))
# With 1 to 1 matching
library(optmatch)
match_maker_output<- match_maker(PC = pcs,
eigen_value = eigen_vals,
data = cov_data,
ids = c("sample"),
case_control = c("case"),
um_controls = 1,
eigen_sum = sum(all_eigen_vals),
weight_dist=TRUE)

# run plot_maker()
plot_maker(data=match_maker_output,
x_var="PC1",
y_var="PC2",
case_control="case",
line=TRUE)
```
Index

* datasets
  * eigenvalues_1000G, 2
  * eigenvalues_all_1000G, 2
  * PCs_1000G, 5

  eigenvalues_1000G, 2
  eigenvalues_all_1000G, 2

  match_maker, 3

  PCs_1000G, 5
  plot_maker, 6