Package 'MiMIR'

April 15, 2022

Title Metabolomics-Based Models for Imputing Risk

Version 1.2

Description Provides an intuitive framework for ad-hoc statistical analysis of 1H-NMR metabolomics by Nightingale Health. It allows to easily explore new metabolomics measurements assayed by Nightingale Health, comparing the distributions with a large Consortium (BBMRI-nl); project previously published metabolic scores [<doi:10.1016/j.ebiom.2021.103764>, <doi:10.1161/CIRCGEN.119.002610>, <doi:10.1038/s4146
019-11311-
9>, <doi:10.7554/eLife.63033>, <doi:10.1161/CIRCULATIONAHA.114.013116>, <doi:10.1007/s00125-019-05001-w>]; and calibrate the metabolic surrogate values to a desired dataset.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Depends R (>= 4.1.0)

Imports caret, DT, foreach, ggplot2, glmnet, heatmaply, matrixStats, pkgload, plotly, plyr, pROC, purrr, shiny, shinycssloaders, shinyFiles, shinydashboard, shinyjs, shinyWidgets, stats, survival, survminer, dplyr

LazyData true

Suggests ggfortify, knitr, rmarkdown

NeedsCompilation no

Author Daniele Bizzarri [aut, cre] (<https://orcid.org/0000-0002-6881-273X>), Marcel Reinders [aut, ths] (<https://orcid.org/0000-0002-1148-1562>), Marian Beekman [aut] (<https://orcid.org/0000-0003-0585-6206>), Pieternella Eline Slagboom [aut, ths] (<https://orcid.org/0000-0002-2875-4723>), Erik van den Akker [aut, ths] (<https://orcid.org/0000-0002-7693-0728>)

Maintainer Daniele Bizzarri <d.bizzarri@lumc.nl>

Repository CRAN

Date/Publication 2022-04-15 10:32:32 UTC

R topics documented:

acc_LOBOV *Accuracies of the LOBOV*

Description

Accuracies of the Leave One Biobank Validation calculated in Bizzarri et al.

Usage

```
data("acc_LOBOV")
```

Format

A data frame with 0 observations on the following 2 variables.

`x` a numeric vector

`y` a numeric vector

Examples

```
data(acc_LOBOV)
## maybe str(acc_LOBOV) ; plot(acc_LOBOV) ...
```

Ahola_Olli_betas *Parameters to calculate the Ahola Olli type 2 diabetes score*

Description

Abbreviatinos of the metabolites, metabolites and coefficients of the score

Usage

```
data("Ahola_Olli_betas")
```

Format

A data frame with 7 observations on the following 3 variables.

`Abbreviation` a character vector

`Metabolite` a character vector

`Beta_value` a numeric vector

Source

https://pubmed.ncbi.nlm.nih.gov/31584131/

Examples

```
data(Ahola_Olli_betas)
## maybe str(Ahola_Olli_betas) ; plot(Ahola_Olli_betas) ...
```

BBMRI_hist *Histograms of the metabolites in BBMRI-nl*

Description

This list collects the histograms of the 57 metabolites selected collected in BBMRI-nl with which the user will be able to compare his/her own dataset.

Usage

```
data("BBMRI_hist")
```

Format

The format is: List of 57 $ ala :List of 6 ..$ breaks : num [1:100] 0.0691 0.0764 0.0836 0.0908 0.0981 ... ..$ counts : int [1:99] 1 0 1 2 1 4 11 18 46 49 ... ..$ density : num [1:99] 0.00443 0 0.00443 0.00885 0.00443 ... ..$ mids : num [1:99] 0.0727 0.08 0.0872 0.0945 0.1017 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ gln :List of 6 ..$ breaks : num [1:100] 0.0406 0.0495 0.0585 0.0674 0.0764 ... ..$ counts : int [1:99] 2 0 1 0 0 1 2 0 1 3 ... ..$ density : num [1:99] 0.00834 0 0.00417 0 0 ... ..$ mids : num [1:99] 0.045 0.054 0.0629 0.0719 0.0808 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ his :List of 6 ..$ breaks : num [1:100] 1.96e-06 6.39e-03 1.28e-02 1.92e-02 2.55e-02 ... ..$ counts : int [1:99] 7 16 84 299 855 1907 3387 4752 5676 5354 ... ..$ density : num [1:99] 0.0352 0.0804 0.4219 1.5018 4.2945 ... ..$ mids : num [1:99] 0.0032 0.00958 0.01597 0.02236 0.02874 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ phe :List of 6 ..$ breaks : num [1:100] 0.016 0.0184 0.0208 0.0231 0.0255 ... ..$ counts : int [1:99] 1 3 5 45 153 455 940 1814 2714 3292 ... ..$ density : num [1:99] 0.0136 0.0408 0.0681 0.6126 2.0827 ... ..$ mids : num [1:99] 0.0172 0.0196 0.0219 0.0243 0.0267 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ tyr :List of 6 ..$ breaks : num [1:100] 0.0167 0.0181 0.0196 0.0211 0.0226 ... ..$ counts : int [1:99] 1 3 7 8 15 28 40 46 80 113 ... ..$ density : num [1:99] 0.0218 0.0655 0.1527 0.1745 0.3273 ... ..$ mids : num [1:99] 0.0174 0.0189 0.0204 0.0218 0.0233 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ile :List of 6 ..$ breaks : num [1:100] 0.000928 0.00299 0.005051 0.007112 0.009174 ... ..$ counts : int [1:99] 1 0 0 3 9 19 35 58 105 168 ... ..$ density : num [1:99] 0.0156 0 0 0.0467 0.14 ... ..$ mids : num [1:99] 0.00196 0.00402 0.00608 0.00814 0.0102 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ leu :List of 6 ..$ breaks : num [1:100] 0 0.00215 0.0043 0.00645 0.00861 ... ..$ counts : int [1:99] 1 0 0 0 1 0 1 1 4 6 ... ..$ density : num [1:99] 0.0149 0 0 0 0.0149 ... ..$ mids : num [1:99] 0.00108 0.00323 0.00538 0.00753 0.00968 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ val :List of 6 ..$ breaks : num [1:100] 0.00795 0.01257 0.01719 0.0218 0.02642 ... ..$ counts : int [1:99] 2 2 3 2 3 1 5 5 6 9 ... ..$ density : num [1:99] 0.0139 0.0139 0.0209 0.0139 0.0209 ... ..$ mids : num [1:99] 0.0103 0.0149 0.0195 0.0241 0.0287 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ glc :List of 6 ..$

breaks : num [1:100] 0 0.223 0.445 0.668 0.89 ... ..$ counts : int [1:99] 88 50 56 56 69 69 92 98 108 143 ... ..$ density : num [1:99] 0.01271 0.00722 0.00809 0.00809 0.00996 ... ..$ mids : num [1:99] 0.111 0.334 0.556 0.779 1.001 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ lac :List of 6 ..$ breaks : num [1:100] 0.153 0.301 0.448 0.595 0.743 ... ..$ counts : int [1:99] 85 361 1837 4156 5210 4875 3994 2944 1972 1342 ... ..$ density : num [1:99] 0.0185 0.0785 0.3995 0.9038 1.133 ... ..$ mids : num [1:99] 0.227 0.374 0.522 0.669 0.816 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ cit :List of 6 ..$ breaks : num [1:100] 1.32e-05 1.90e-01 3.81e-01 5.71e-01 7.61e-01 ... ..$ counts : int [1:99] 30968 222 2 0 0 0 1 1 0 0 ... ..$ density : num [1:99] 5.216041 0.037392 0.000337 0 0 ... ..$ mids : num [1:99] 0.0952 0.2855 0.4758 0.6661 0.8564 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ace :List of 6 ..$ breaks : num [1:100] 0.00397 0.02377 0.04357 0.06336 0.08316 ... ..$ counts : int [1:99] 1542 13598 9185 4641 1479 360 107 38 20 12 ... ..$ density : num [1:99] 2.5 22.06 14.9 7.53 2.4 ... ..$ mids : num [1:99] 0.0139 0.0337 0.0535 0.0733 0.0931 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ acace :List of 6 ..$ breaks : num [1:100] 0 0.0111 0.0223 0.0334 0.0446 ... ..$ counts : int [1:99] 1213 5277 8407 6427 3659 2176 1201 710 502 337 ... ..$ density : num [1:99] 3.51 15.27 24.33 18.6 10.59 ... ..$ mids : num [1:99] 0.00557 0.01671 0.02785 0.03899 0.05014 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ crea :List of 6 ..$ breaks : num [1:100] 0.0161 0.0252 0.0342 0.0432 0.0523 ... ..$ counts : int [1:99] 3 24 460 3132 7060 7698 5700 3338 1650 831 ... ..$ density : num [1:99] 0.0107 0.0856 1.641 11.1731 25.1858 ... ..$ mids : num [1:99] 0.0207 0.0297 0.0387 0.0478 0.0568 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ alb :List of 6 ..$ breaks : num [1:100] 0.0108 0.0121 0.0134 0.0147 0.0161 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.0245 0 0 0 0 ... ..$ mids : num [1:99] 0.0115 0.0128 0.0141 0.0154 0.0167 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ gp :List of 6 ..$ breaks : num [1:100] 0.627 0.659 0.691 0.724 0.756 ... ..$ counts : int [1:99] 2 2 1 2 3 11 27 37 58 159 ... ..$ density : num [1:99] 0.001985 0.001985 0.000993 0.001985 0.002978 ... ..$ mids : num [1:99] 0.643 0.675 0.707 0.74 0.772 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_vldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0623 0.1246 0.1869 0.2492 ... ..$ counts : int [1:99] 34 127 655 1602 2494 3010 3000 2894 2627 2267 ... ..$ density : num [1:99] 0.0175 0.0653 0.3368 0.8237 1.2824 ... ..$ mids : num [1:99] 0.0312 0.0935 0.1558 0.2181 0.2804 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_vldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0369 0.0738 0.1108 0.1477 ... ..$ counts : int [1:99] 19 0 1 6 28 92 217 346 585 790 ... ..$ density : num [1:99] 0.016486 0 0.000868 0.005206 0.024295 ... ..$ mids : num [1:99] 0.0185 0.0554 0.0923 0.1292 0.1661 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ xs_vldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0176 0.0351 0.0527 0.0702 ... ..$ counts : int [1:99] 10 0 0 0 0 0 6 9 19 ... ..$ density : num [1:99] 0.0182 0 0 0 0 ... ..$ mids : num [1:99] 0.00878 0.02633 0.04389 0.06144 0.079 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ idl_l :List of 6 ..$ breaks : num [1:100] 0.0518 0.0914 0.1311 0.1707 0.2104 ... ..$ counts : int [1:99] 1 0 4 4 16 34 33 62 58 99 ... ..$ density : num [1:99] 0.000808 0 0.003231 0.003231 0.012924 ... ..$ mids : num [1:99] 0.0716 0.1112 0.1509 0.1906 0.2302 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ l_ldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0502 0.1004 0.1506 0.2008 ... ..$ counts : int [1:99] 32 0 1 2 18 30 58 119 140 261 ... ..$ density : num [1:99] 0.020418 0 0.000638 0.001276 0.011485 ... ..$ mids : num [1:99] 0.0251 0.0753 0.1255 0.1757 0.226 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_ldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0294 0.0588 0.0882 0.1175 ... ..$ counts : int [1:99] 134 0 0 0 3 46 101 151 260 373 ... ..$ density : num [1:99]

0.14609 0 0 0 0.00327 ... ..$ mids : num [1:99] 0.0147 0.0441 0.0735 0.1028 0.1322 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_ldl_l :List of 6 ..$ breaks : num [1:100] 0 0.0179 0.0358 0.0537 0.0716 ... ..$ counts : int [1:99] 134 0 0 0 3 8 42 69 135 201 ... ..$ density : num [1:99] 0.23982 0 0 0 0.00537 ... ..$ mids : num [1:99] 0.00895 0.02685 0.04475 0.06265 0.08055 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_hdl_l :List of 6 ..$ breaks : num [1:100] 0 0.0184 0.0368 0.0552 0.0737 ... ..$ counts : int [1:99] 345 0 0 0 0 0 1 0 0 0 ... ..$ density : num [1:99] 0.6 0 0 0 0 ... ..$ mids : num [1:99] 0.00921 0.02762 0.04604 0.06445 0.08286 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_hdl_l :List of 6 ..$ breaks : num [1:100] 0 0.0179 0.0358 0.0536 0.0715 ... ..$ counts : int [1:99] 39 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.0699 0 0 0 0 ... ..$ mids : num [1:99] 0.00894 0.02682 0.0447 0.06258 0.08045 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ idl_c :List of 6 ..$ breaks : num [1:100] 0 0.0272 0.0545 0.0817 0.1089 ... ..$ counts : int [1:99] 33 3 8 7 25 33 51 58 81 127 ... ..$ density : num [1:99] 0.03883 0.00353 0.00941 0.00824 0.02942 ... ..$ mids : num [1:99] 0.0136 0.0408 0.0681 0.0953 0.1225 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ serum_c :List of 6 ..$ breaks : num [1:100] 0.371 0.515 0.66 0.804 0.948 ... ..$ counts : int [1:99] 1 0 2 8 19 20 39 38 50 66 ... ..$ density : num [1:99] 0.000222 0 0.000445 0.001778 0.004223 ... ..$ mids : num [1:99] 0.443 0.587 0.732 0.876 1.02 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ vldl_c :List of 6 ..$ breaks : num [1:100] 1.96e-09 4.60e-02 9.19e-02 1.38e-01 1.84e-01 ... ..$ counts : int [1:99] 1 4 18 47 111 169 291 399 558 772 ... ..$ density : num [1:99] 0.000697 0.002787 0.012544 0.032753 0.077352 ... ..$ mids : num [1:99] 0.023 0.069 0.115 0.161 0.207 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ldl_c :List of 6 ..$ breaks : num [1:100] 6.34e-13 7.03e-02 1.41e-01 2.11e-01 2.81e-01 ... ..$ counts : int [1:99] 8 23 29 56 104 141 237 301 413 598 ... ..$ density : num [1:99] 0.00365 0.01048 0.01322 0.02552 0.0474 ... ..$ mids : num [1:99] 0.0351 0.1054 0.1757 0.246 0.3163 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl_c :List of 6 ..$ breaks : num [1:100] 0.0589 0.0903 0.1218 0.1532 0.1847 ... ..$ counts : int [1:99] 1 0 1 1 1 6 3 5 10 16 ... ..$ density : num [1:99] 0.00102 0 0.00102 0.00102 0.00102 ... ..$ mids : num [1:99] 0.0746 0.1061 0.1375 0.169 0.2004 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl2_c :List of 6 ..$ breaks : num [1:100] 0 0.0264 0.0528 0.0792 0.1057 ... ..$ counts : int [1:99] 124 32 47 53 46 80 82 83 101 128 ... ..$ density : num [1:99] 0.1504 0.0388 0.057 0.0643 0.0558 ... ..$ mids : num [1:99] 0.0132 0.0396 0.066 0.0924 0.1189 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl3_c :List of 6 ..$ breaks : num [1:100] 0.0589 0.0687 0.0786 0.0884 0.0983 ... ..$ counts : int [1:99] 1 0 0 0 0 0 1 0 0 0 ... ..$ density : num [1:99] 0.00325 0 0 0 0 ... ..$ mids : num [1:99] 0.0638 0.0737 0.0835 0.0934 0.1032 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ vldl_d :List of 6 ..$ breaks : num [1:100] 32.6 32.7 32.8 33 33.1 ... ..$ counts : int [1:99] 1 0 1 1 3 1 5 13 13 14 ... ..$ density : num [1:99] 0.000277 0 0.000277 0.000277 0.000831 ... ..$ mids : num [1:99] 32.7 32.8 32.9 33 33.1 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ldl_d :List of 6 ..$ breaks : num [1:100] 21.9 21.9 22 22 22.1 ... ..$ counts : int [1:99] 2 1 1 0 0 0 0 2 0 0 ... ..$ density : num [1:99] 0.001767 0.000883 0.000883 0 0 ... ..$ mids : num [1:99] 21.9 22 22 22 22.1 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl_d :List of 6 ..$ breaks : num [1:100] 8.98 9 9.02 9.04 9.06 ... ..$ counts : int [1:99] 1 0 0 0 0 1 3 2 1 3 ... ..$ density : num [1:99] 0.0016 0 0 0 0 ... ..$ mids : num [1:99] 8.99 9.01 9.03 9.05 9.07 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ serum_tg :List of 6 ..$ breaks : num [1:100] 0.117 0.235 0.353 0.472 0.59 ... ..$ counts : int [1:99] 6 61 308 944 1926 2894 3040 3214 2892 2666 ... ..$ density : num [1:99]

0.00163 0.01653 0.08347 0.25582 0.52194 ... ..$ mids : num [1:99] 0.176 0.294 0.413 0.531 0.649 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totpg :List of 6 ..$ breaks : num [1:100] 0.194 0.238 0.281 0.325 0.368 ... ..$ counts : int [1:99] 2 0 2 3 5 2 8 30 25 35 ... ..$ density : num [1:99] 0.00148 0 0.00148 0.00222 0.00369 ... ..$ mids : num [1:99] 0.216 0.259 0.303 0.346 0.39 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pc :List of 6 ..$ breaks : num [1:100] 0 0.0498 0.0995 0.1493 0.199 ... ..$ counts : int [1:99] 14 0 0 0 1 0 4 6 4 25 ... ..$ density : num [1:99] 0.009154 0 0 0 0.000654 ... ..$ mids : num [1:99] 0.0249 0.0746 0.1244 0.1742 0.2239 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sm :List of 6 ..$ breaks : num [1:100] 0.128 0.138 0.149 0.159 0.17 ... ..$ counts : int [1:99] 1 1 0 5 10 8 13 25 30 42 ... ..$ density : num [1:99] 0.00305 0.00305 0 0.01525 0.03051 ... ..$ mids : num [1:99] 0.133 0.143 0.154 0.164 0.175 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totcho :List of 6 ..$ breaks : num [1:100] 0.0621 0.1132 0.1644 0.2155 0.2667 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 2 2 ... ..$ density : num [1:99] 0.000628 0 0 0 0 ... ..$ mids : num [1:99] 0.0877 0.1388 0.19 0.2411 0.2922 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ apoa1 :List of 6 ..$ breaks : num [1:100] 0.545 0.566 0.586 0.607 0.628 ... ..$ counts : int [1:99] 1 0 0 0 3 5 4 3 11 12 ... ..$ density : num [1:99] 0.00153 0 0 0 0.0046 ... ..$ mids : num [1:99] 0.555 0.576 0.597 0.618 0.639 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ apob :List of 6 ..$ breaks : num [1:100] 0.094 0.126 0.159 0.192 0.224 ... ..$ counts : int [1:99] 1 0 0 0 0 0 1 8 17 45 ... ..$ density : num [1:99] 0.000985 0 0 0 0 ... ..$ mids : num [1:99] 0.11 0.143 0.175 0.208 0.24 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totfa :List of 6 ..$ breaks : num [1:100] 3.69 4.09 4.5 4.91 5.32 ... ..$ counts : int [1:99] 7 21 60 97 153 253 392 489 650 779 ... ..$ density : num [1:99] 0.000554 0.001662 0.004749 0.007677 0.012109 ... ..$ mids : num [1:99] 3.89 4.3 4.71 5.11 5.52 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ dha :List of 6 ..$ breaks : num [1:100] 0 0.0106 0.0213 0.0319 0.0425 ... ..$ counts : int [1:99] 308 93 151 388 786 1171 1481 1922 2247 2452 ... ..$ density : num [1:99] 0.935 0.282 0.459 1.178 2.387 ... ..$ mids : num [1:99] 0.00532 0.01595 0.02659 0.03723 0.04786 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ la :List of 6 ..$ breaks : num [1:100] 0.0408 0.1564 0.2721 0.3877 0.5034 ... ..$ counts : int [1:99] 1 0 5 12 41 62 71 82 106 124 ... ..$ density : num [1:99] 0.000279 0 0.001397 0.003352 0.011454 ... ..$ mids : num [1:99] 0.0986 0.2143 0.3299 0.4456 0.5612 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw3 :List of 6 ..$ breaks : num [1:100] 2.38e-10 2.62e-02 5.24e-02 7.87e-02 1.05e-01 ... ..$ counts : int [1:99] 95 88 132 296 578 737 761 840 1039 1298 ... ..$ density : num [1:99] 0.117 0.108 0.163 0.365 0.712 ... ..$ mids : num [1:99] 0.0131 0.0393 0.0656 0.0918 0.118 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw6 :List of 6 ..$ breaks : num [1:100] 0.85 0.969 1.088 1.207 1.326 ... ..$ counts : int [1:99] 3 19 36 66 90 117 168 191 261 346 ... ..$ density : num [1:99] 0.000815 0.005159 0.009775 0.017922 0.024439 ... ..$ mids : num [1:99] 0.909 1.028 1.147 1.266 1.385 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pufa :List of 6 ..$ breaks : num [1:100] 0.864 0.993 1.123 1.252 1.382 ... ..$ counts : int [1:99] 5 23 34 62 88 114 141 197 224 319 ... ..$ density : num [1:99] 0.00125 0.00574 0.00849 0.01549 0.02198 ... ..$ mids : num [1:99] 0.929 1.058 1.187 1.317 1.446 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ mufa :List of 6 ..$ breaks : num [1:100] 0.719 0.886 1.053 1.221 1.388 ... ..$ counts : int [1:99] 9 38 152 388 715 1168 1523 1915 2313 2320 ... ..$ density : num [1:99] 0.00174 0.00735 0.0294 0.07505 0.1383 ... ..$ mids : num [1:99] 0.803 0.97 1.137 1.304 1.471 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sfa :List of 6 ..$ breaks : num [1:100] 1.51 1.66 1.81 1.96 2.11 ... ..$ counts : int [1:99] 5 6 18 39 98 147 253 443

645 870 ... ..$ density : num [1:99] 0.00108 0.0013 0.0039 0.00844 0.02121 ... ..$ mids : num [1:99] 1.59 1.74 1.89 2.04 2.18 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw3_fa :List of 6 ..$ breaks : num [1:100] 2.88e-09 1.77e-01 3.54e-01 5.30e-01 7.07e-01 ... ..$ counts : int [1:99] 54 26 26 41 49 51 59 127 216 457 ... ..$ density : num [1:99] 0.00987 0.00475 0.00475 0.00749 0.00895 ... ..$ mids : num [1:99] 0.0884 0.2652 0.4419 0.6187 0.7955 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw6_fa :List of 6 ..$ breaks : num [1:100] 10.1 10.4 10.8 11.2 11.5 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 8.86e-05 0.00 0.00 0.00 0.00 ... ..$ mids : num [1:99] 10.3 10.6 11 11.3 11.7 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pufa_fa :List of 6 ..$ breaks : num [1:100] 15 15.5 16 16.5 17 ... ..$ counts : int [1:99] 2 1 1 1 2 1 2 6 7 10 ... ..$ density : num [1:99] 1.28e-04 6.42e-05 6.42e-05 6.42e-05 1.28e-04 ... ..$ mids : num [1:99] 15.3 15.8 16.3 16.8 17.3 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ mufa_fa :List of 6 ..$ breaks : num [1:100] 8.38 8.77 9.17 9.57 9.97 ... ..$ counts : int [1:99] 1 0 0 1 0 0 0 0 0 0 ... ..$ density : num [1:99] 8.15e-05 0.00 0.00 8.15e-05 0.00 ... ..$ mids : num [1:99] 8.58 8.97 9.37 9.77 10.16 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sfa_fa :List of 6 ..$ breaks : num [1:100] 25.8 26 26.3 26.6 26.9 ... ..$ counts : int [1:99] 1 0 1 0 0 0 0 0 0 1 ... ..$ density : num [1:99] 0.000117 0 0.000117 0 0 ... ..$ mids : num [1:99] 25.9 26.2 26.5 26.7 27 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ unsatdeg :List of 6 ..$ breaks : num [1:100] 0.645 0.658 0.672 0.685 0.698 ... ..$ counts : int [1:99] 1 0 1 0 0 1 1 2 1 4 ... ..$ density : num [1:99] 0.0024 0 0.0024 0 0 ... ..$ mids : num [1:99] 0.651 0.665 0.678 0.692 0.705 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ xxl_vldl_l:List of 6 ..$ breaks : num [1:100] 0 0.00795 0.0159 0.02385 0.0318 ... ..$ counts : int [1:99] 4020 4941 5370 4292 3172 2330 1687 1205 992 702 ... ..$ density : num [1:99] 16.3 20 21.7 17.4 12.8 ... ..$ mids : num [1:99] 0.00398 0.01193 0.01988 0.02783 0.03578 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram"

Examples

```
data(BBMRI_hist)
## maybe str(BBMRI_hist) ; plot(BBMRI_hist) ...
```

`BBMRI_hist_plot` *multi_hist*

Description

Function to plot the ~60 metabolites used for the metabolomics-based scores and compare them to to their distributions in BBMRI-nl

Usage

```
BBMRI_hist_plot(
  dat,
  x_name,
  color = MiMIR::c21,
  scaled = FALSE,
```

```
  datatype = "metabolite",
  main = "Comparison with the metabolites measures in BBMRI"
)
```

Arguments

dat	data.frame or matrix with the metabolites
x_name	string with the name of the selected variable
color	colors selected for all the variables
scaled	logical to z-scale the variables
datatype	a character vector indicating what data type is being plotted
main	title of the plot

Details

This function plots the distribution of a metabolic feature in the uploaded dataset, compared to their distributions in BBMRI-nl. The selection of features available is done following the metabolic scores features.

Value

plotly image with the histogram of the selected variable compared to the distributions in BBMRI-nl

References

The selection of metabolic features available is the one selected by the papers: Deelen,J. et al. (2019) A metabolic profile of all-cause mortality risk identified in an observational study of 44,168 individuals. Nature Communications, 10, 1–8, doi: 10.1038/s41467-019-11311-9. Ahola-Olli,A.V. et al. (2019) Circulating metabolites and the risk of type 2 diabetes: a prospective study of 11,896 young adults from four Finnish cohorts. Diabetologia, 62, 2298–2309, doi: 10.1007/s00125-019-05001-w Würtz,P. et al. (2015) Metabolite profiling and cardiovascular event risk: a prospective study of 3 population-based cohorts. Circulation, 131, 774–785, doi: 10.1161/CIRCULATION-AHA.114.013116 Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764 van den Akker Erik B. et al. (2020) Metabolic Age Based on the BBMRI-NL 1H-NMR Metabolomics Repository as Biomarker of Age-related Disease. Circulation: Genomic and Precision Medicine, 13, 541–547, doi:10.1161/CIRCGEN.119.002610

Examples

```
library(plotly)
library(MiMIR)

#load the metabolites dataset
metabolic_measures <- synthetic_metabolic_dataset

BBMRI_hist_plot(metabolic_measures, x_name="alb", scaled=TRUE)
```

`BBMRI_hist_scaled` *Histograms of the metabolites scaled in BBMRI-nl*

Description

This list collects the histograms of the 57 metabolites scaled collected in BBMRI-nl with which the user will be able to compare his/her own dataset.

Usage

```
data("BBMRI_hist_scaled")
```

Format

The format is: List of 57 $ ala :List of 6 ..$ breaks : num [1:100] -2.98 -2.89 -2.8 -2.71 -2.62 ... ..$ counts : int [1:99] 1 0 1 2 1 4 11 18 46 49 ... ..$ density : num [1:99] 0.000356 0 0.000356 0.000712 0.000356 ... ..$ mids : num [1:99] -2.93 -2.84 -2.75 -2.66 -2.57 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ gln :List of 6 ..$ breaks : num [1:100] -5.32 -5.21 -5.09 -4.98 -4.86 ... ..$ counts : int [1:99] 2 0 1 0 0 1 2 0 1 3 ... ..$ density : num [1:99] 0.000647 0 0.000323 0 0 ... ..$ mids : num [1:99] -5.27 -5.15 -5.04 -4.92 -4.8 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ his :List of 6 ..$ breaks : num [1:100] -3.72 -3.3 -2.87 -2.45 -2.03 ... ..$ counts : int [1:99] 7 16 84 299 855 1907 3387 4752 5676 5354 ... ..$ density : num [1:99] 0.000533 0.001217 0.006391 0.022748 0.065048 ... ..$ mids : num [1:99] -3.51 -3.08 -2.66 -2.24 -1.82 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ phe :List of 6 ..$ breaks : num [1:100] -2.74 -2.53 -2.32 -2.1 -1.89 ... ..$ counts : int [1:99] 1 3 5 45 153 455 940 1814 2714 3292 ... ..$ density : num [1:99] 0.000151 0.000454 0.000756 0.006804 0.023134 ... ..$ mids : num [1:99] -2.64 -2.42 -2.21 -2 -1.79 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ tyr :List of 6 ..$ breaks : num [1:100] -2.84 -2.74 -2.64 -2.54 -2.45 ... ..$ counts : int [1:99] 1 3 7 8 15 28 40 46 80 113 ... ..$ density : num [1:99] 0.000326 0.000978 0.002282 0.002608 0.004891 ... ..$ mids : num [1:99] -2.79 -2.69 -2.59 -2.5 -2.4 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ile :List of 6 ..$ breaks : num [1:100] -2.64 -2.53 -2.43 -2.32 -2.21 ... ..$ counts : int [1:99] 1 0 0 3 9 19 35 58 105 168 ... ..$ density : num [1:99] 0.000299 0 0 0.000898 0.002695 ... ..$ mids : num [1:99] -2.59 -2.48 -2.37 -2.27 -2.16 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ leu :List of 6 ..$ breaks : num [1:100] -3.82 -3.69 -3.56 -3.43 -3.29 ... ..$ counts : int [1:99] 1 0 0 0 1 0 1 1 4 6 ... ..$ density : num [1:99] 0.000241 0 0 0 0.000241 ... ..$ mids : num [1:99] -3.76 -3.63 -3.49 -3.36 -3.23 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ val :List of 6 ..$ breaks : num [1:100] -3.94 -3.82 -3.7 -3.57 -3.45 ... ..$ counts : int [1:99] 2 2 3 2 3 1 5 5 6 9 ... ..$ density : num [1:99] 0.000514 0.000514 0.000771 0.000514 0.000771 ... ..$ mids : num [1:99] -3.88 -3.76 -3.63 -3.51 -3.38 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ glc :List of 6 ..$ breaks : num [1:100] -2.9 -2.76 -2.61 -2.47 -2.33 ... ..$ counts : int [1:99] 88 50 56 56 69 69 92 98 108 143 ... ..$ density : num [1:99] 0.02 0.0113 0.0127 0.0127 0.0157 ... ..$ mids : num [1:99] -2.83 -2.68 -2.54 -2.4 -2.26 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ lac :List of 6 ..$ breaks : num [1:100] -1.05 -0.912 -0.774 -0.636 -0.498 ... ..$ counts : int [1:99] 85 361 1837 4156 5210 4875 3994 2944 1972 1342 ... ..$ density : num [1:99] 0.0197 0.0838 0.4265 0.965 1.2097 ... ..$ mids : num [1:99]

-0.981 -0.843 -0.705 -0.567 -0.429 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ cit :List of 6 ..$ breaks : num [1:100] -0.566 0.51 1.586 2.662 3.738 ... ..$ counts : int [1:99] 30968 222 2 0 0 0 1 1 0 0 ... ..$ density : num [1:99] 9.22e-01 6.61e-03 5.96e-05 0.00 0.00 ... ..$ mids : num [1:99] -0.0282 1.048 2.1242 3.2004 4.2766 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ace :List of 6 ..$ breaks : num [1:100] -1.197 -0.693 -0.188 0.317 0.822 ... ..$ counts : int [1:99] 1542 13598 9185 4641 1479 360 107 38 20 12 ... ..$ density : num [1:99] 0.0981 0.8652 0.5844 0.2953 0.0941 ... ..$ mids : num [1:99] -0.9451 -0.4403 0.0646 0.5694 1.0742 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ acace :List of 6 ..$ breaks : num [1:100] -1.236 -0.91 -0.584 -0.257 0.069 ... ..$ counts : int [1:99] 1213 5277 8407 6427 3659 2176 1201 710 502 337 ... ..$ density : num [1:99] 0.12 0.522 0.831 0.635 0.362 ... ..$ mids : num [1:99] -1.0728 -0.7466 -0.4204 -0.0942 0.2321 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ crea :List of 6 ..$ breaks : num [1:100] -2.397 -1.996 -1.595 -1.195 -0.794 ... ..$ counts : int [1:99] 3 24 460 3132 7060 7698 5700 3338 1650 831 ... ..$ density : num [1:99] 0.000241 0.001929 0.036976 0.251758 0.567501 ... ..$ mids : num [1:99] -2.196 -1.796 -1.395 -0.994 -0.594 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ alb :List of 6 ..$ breaks : num [1:100] -10.62 -10.43 -10.25 -10.06 -9.88 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.000174 0 0 0 0 ... ..$ mids : num [1:99] -10.52 -10.34 -10.16 -9.97 -9.79 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ gp :List of 6 ..$ breaks : num [1:100] -3.28 -3.13 -2.98 -2.83 -2.68 ... ..$ counts : int [1:99] 2 2 1 2 3 11 27 37 58 159 ... ..$ density : num [1:99] 0.000424 0.000424 0.000212 0.000424 0.000636 ... ..$ mids : num [1:99] -3.21 -3.06 -2.91 -2.75 -2.6 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_vldl_l :List of 6 ..$ breaks : num [1:100] -1.625 -1.465 -1.306 -1.146 -0.987 ... ..$ counts : int [1:99] 34 127 655 1602 2494 3010 3000 2894 2627 2267 ... ..$ density : num [1:99] 0.00683 0.0255 0.13149 0.3216 0.50067 ... ..$ mids : num [1:99] -1.545 -1.386 -1.226 -1.066 -0.907 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_vldl_l :List of 6 ..$ breaks : num [1:100] -2.78 -2.62 -2.47 -2.32 -2.17 ... ..$ counts : int [1:99] 19 0 1 6 28 92 217 346 585 790 ... ..$ density : num [1:99] 0.00398 0 0.00021 0.00126 0.00587 ... ..$ mids : num [1:99] -2.7 -2.55 -2.39 -2.24 -2.09 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ xs_vldl_l :List of 6 ..$ breaks : num [1:100] -4.05 -3.92 -3.79 -3.66 -3.53 ... ..$ counts : int [1:99] 10 0 0 0 0 0 0 6 9 19 ... ..$ density : num [1:99] 0.00245 0 0 0 0 ... ..$ mids : num [1:99] -3.99 -3.85 -3.72 -3.59 -3.46 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ idl_l :List of 6 ..$ breaks : num [1:100] -3.42 -3.28 -3.15 -3.01 -2.87 ... ..$ counts : int [1:99] 1 0 4 4 16 34 33 62 58 99 ... ..$ density : num [1:99] 0.000236 0 0.000943 0.000943 0.003774 ... ..$ mids : num [1:99] -3.35 -3.21 -3.08 -2.94 -2.81 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ l_ldl_l :List of 6 ..$ breaks : num [1:100] -3.12 -2.99 -2.85 -2.72 -2.58 ... ..$ counts : int [1:99] 32 0 1 2 18 30 58 119 140 261 ... ..$ density : num [1:99] 0.00755 0 0.000236 0.000472 0.004247 ... ..$ mids : num [1:99] -3.06 -2.92 -2.79 -2.65 -2.51 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_ldl_l :List of 6 ..$ breaks : num [1:100] -2.89 -2.76 -2.63 -2.5 -2.37 ... ..$ counts : int [1:99] 134 0 0 0 3 46 101 151 260 373 ... ..$ density : num [1:99] 0.033164 0 0 0 0.000742 ... ..$ mids : num [1:99] -2.82 -2.69 -2.56 -2.43 -2.3 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_ldl_l :List of 6 ..$ breaks : num [1:100] -3.1 -2.97 -2.84 -2.71 -2.58 ... ..$ counts : int [1:99] 134 0 0 0 3 8 42 69 135 201 ... ..$ density : num [1:99] 0.032903 0 0 0 0.000737 ... ..$ mids : num [1:99] -3.04 -2.91 -2.78 -2.65 -2.52 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ m_hdl_l :List of 6 ..$ breaks : num [1:100] -3.43 -3.35 -3.27 -3.19 -3.1 ... ..$ counts : int [1:99] 345 0 0 0 0 0 1 0 0 0

... ..$ density : num [1:99] 0.134 0 0 0 0 ... ..$ mids : num [1:99] -3.39 -3.31 -3.23 -3.14 -3.06 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ s_hdl_l :List of 6 ..$ breaks : num [1:100] -6.64 -6.52 -6.4 -6.28 -6.16 ... ..$ counts : int [1:99] 39 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.0105 0 0 0 0 ... ..$ mids : num [1:99] -6.58 -6.46 -6.34 -6.22 -6.1 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ idl_c :List of 6 ..$ breaks : num [1:100] -3.33 -3.19 -3.06 -2.92 -2.79 ... ..$ counts : int [1:99] 33 3 8 7 25 33 51 58 81 127 ... ..$ density : num [1:99] 0.00781 0.00071 0.00189 0.00166 0.00592 ... ..$ mids : num [1:99] -3.26 -3.13 -2.99 -2.85 -2.72 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ serum_c :List of 6 ..$ breaks : num [1:100] -3.65 -3.52 -3.38 -3.25 -3.11 ... ..$ counts : int [1:99] 1 0 2 8 19 20 39 38 50 66 ... ..$ density : num [1:99] 0.000238 0 0.000476 0.001905 0.004524 ... ..$ mids : num [1:99] -3.59 -3.45 -3.32 -3.18 -3.05 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ vldl_c :List of 6 ..$ breaks : num [1:100] -2.77 -2.61 -2.46 -2.3 -2.15 ... ..$ counts : int [1:99] 1 4 18 47 111 169 291 399 558 772 ... ..$ density : num [1:99] 0.000208 0.000832 0.003743 0.009773 0.023081 ... ..$ mids : num [1:99] -2.69 -2.54 -2.38 -2.23 -2.07 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ldl_c :List of 6 ..$ breaks : num [1:100] -2.71 -2.58 -2.45 -2.32 -2.19 ... ..$ counts : int [1:99] 8 23 29 56 104 141 237 301 413 598 ... ..$ density : num [1:99] 0.00198 0.0057 0.00719 0.01389 0.02579 ... ..$ mids : num [1:99] -2.64 -2.51 -2.38 -2.25 -2.12 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl_c :List of 6 ..$ breaks : num [1:100] -3.53 -3.44 -3.36 -3.27 -3.18 ... ..$ counts : int [1:99] 1 0 1 1 1 6 3 5 10 16 ... ..$ density : num [1:99] 0.000363 0 0.000363 0.000363 0.000363 ... ..$ mids : num [1:99] -3.49 -3.4 -3.31 -3.22 -3.13 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl2_c :List of 6 ..$ breaks : num [1:100] -2.59 -2.51 -2.43 -2.35 -2.27 ... ..$ counts : int [1:99] 124 32 47 53 46 80 82 83 101 128 ... ..$ density : num [1:99] 0.0494 0.0127 0.0187 0.0211 0.0183 ... ..$ mids : num [1:99] -2.55 -2.47 -2.39 -2.31 -2.23 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl3_c :List of 6 ..$ breaks : num [1:100] -6.42 -6.27 -6.11 -5.96 -5.8 ... ..$ counts : int [1:99] 1 0 0 0 0 0 1 0 0 0 ... ..$ density : num [1:99] 0.000207 0 0 0 0 ... ..$ mids : num [1:99] -6.34 -6.19 -6.03 -5.88 -5.73 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ vldl_d :List of 6 ..$ breaks : num [1:100] -3.03 -2.95 -2.86 -2.78 -2.69 ... ..$ counts : int [1:99] 1 0 1 1 3 1 5 13 13 14 ... ..$ density : num [1:99] 0.00038 0 0.00038 0.00038 0.00114 ... ..$ mids : num [1:99] -2.99 -2.9 -2.82 -2.74 -2.65 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ ldl_d :List of 6 ..$ breaks : num [1:100] -11.4 -11.2 -11 -10.7 -10.5 ... ..$ counts : int [1:99] 2 1 1 0 0 0 0 2 0 0 ... ..$ density : num [1:99] 0.000267 0.000133 0.000133 0 0 ... ..$ mids : num [1:99] -11.3 -11.1 -10.8 -10.6 -10.4 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ hdl_d :List of 6 ..$ breaks : num [1:100] -3.72 -3.64 -3.56 -3.49 -3.41 ... ..$ counts : int [1:99] 1 0 0 0 0 1 3 2 1 3 ... ..$ density : num [1:99] 0.000413 0 0 0 0 ... ..$ mids : num [1:99] -3.68 -3.6 -3.53 -3.45 -3.37 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ serum_tg :List of 6 ..$ breaks : num [1:100] -1.77 -1.6 -1.43 -1.26 -1.09 ... ..$ counts : int [1:99] 6 61 308 944 1926 2894 3040 3214 2892 2666 ... ..$ density : num [1:99] 0.00114 0.0116 0.05858 0.17956 0.36634 ... ..$ mids : num [1:99] -1.68 -1.52 -1.35 -1.18 -1.01 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totpg :List of 6 ..$ breaks : num [1:100] -3.75 -3.65 -3.55 -3.45 -3.35 ... ..$ counts : int [1:99] 2 0 2 3 5 2 8 30 25 35 ... ..$ density : num [1:99] 0.000641 0 0.000641 0.000962 0.001603 ... ..$ mids : num [1:99] -3.7 -3.6 -3.5 -3.4 -3.3 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pc :List of 6 ..$ breaks : num [1:100] -4.53 -4.41 -4.29 -4.17 -4.06 ... ..$ counts : int [1:99] 14 0 0 0 1 0 4 6 4 25 ... ..$ density : num [1:99] 0.003872 0 0 0 0.000277 ... ..$ mids : num [1:99] -4.47 -4.35 -4.23

-4.12 -4 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sm :List of 6 ..$ breaks : num [1:100] -3.5 -3.39 -3.28 -3.17 -3.06 ... ..$ counts : int [1:99] 1 1 0 5 10 8 13 25 30 42 ... ..$ density : num [1:99] 0.000293 0.000293 0 0.001465 0.002931 ... ..$ mids : num [1:99] -3.45 -3.34 -3.23 -3.12 -3.01 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totcho :List of 6 ..$ breaks : num [1:100] -4.89 -4.78 -4.66 -4.55 -4.43 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 0 2 2 ... ..$ density : num [1:99] 0.000279 0 0 0 0 ... ..$ mids : num [1:99] -4.84 -4.72 -4.61 -4.49 -4.38 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ apoa1 :List of 6 ..$ breaks : num [1:100] -4.24 -4.15 -4.06 -3.97 -3.88 ... ..$ counts : int [1:99] 1 0 0 0 3 5 4 3 11 12 ... ..$ density : num [1:99] 0.000358 0 0 0 0.001073 ... ..$ mids : num [1:99] -4.19 -4.1 -4.01 -3.93 -3.84 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ apob :List of 6 ..$ breaks : num [1:100] -3.71 -3.56 -3.42 -3.28 -3.14 ... ..$ counts : int [1:99] 1 0 0 0 0 0 1 8 17 45 ... ..$ density : num [1:99] 0.000225 0 0 0 0 ... ..$ mids : num [1:99] -3.64 -3.49 -3.35 -3.21 -3.07 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ totfa :List of 6 ..$ breaks : num [1:100] -2.69 -2.54 -2.39 -2.25 -2.1 ... ..$ counts : int [1:99] 7 21 60 97 153 253 392 489 650 779 ... ..$ density : num [1:99] 0.00154 0.00461 0.01316 0.02127 0.03355 ... ..$ mids : num [1:99] -2.62 -2.47 -2.32 -2.17 -2.03 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ dha :List of 6 ..$ breaks : num [1:100] -2.21 -2.03 -1.85 -1.67 -1.49 ... ..$ counts : int [1:99] 308 93 151 388 786 1171 1481 1922 2247 2452 ... ..$ density : num [1:99] 0.0557 0.0168 0.0273 0.0701 0.1421 ... ..$ mids : num [1:99] -2.12 -1.94 -1.76 -1.58 -1.4 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ la :List of 6 ..$ breaks : num [1:100] -3.58 -3.43 -3.29 -3.14 -3 ... ..$ counts : int [1:99] 1 0 5 12 41 62 71 82 106 124 ... ..$ density : num [1:99] 0.000223 0 0.001114 0.002674 0.009135 ... ..$ mids : num [1:99] -3.51 -3.36 -3.22 -3.07 -2.93 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw3 :List of 6 ..$ breaks : num [1:100] -2.45 -2.28 -2.12 -1.95 -1.78 ... ..$ counts : int [1:99] 95 88 132 296 578 737 761 840 1039 1298 ... ..$ density : num [1:99] 0.0184 0.017 0.0255 0.0573 0.1119 ... ..$ mids : num [1:99] -2.37 -2.2 -2.03 -1.87 -1.7 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw6 :List of 6 ..$ breaks : num [1:100] -3.09 -2.96 -2.82 -2.69 -2.56 ... ..$ counts : int [1:99] 3 19 36 66 90 117 168 191 261 346 ... ..$ density : num [1:99] 0.000736 0.004658 0.008826 0.016181 0.022065 ... ..$ mids : num [1:99] -3.02 -2.89 -2.76 -2.63 -2.49 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pufa :List of 6 ..$ breaks : num [1:100] -3.12 -2.99 -2.86 -2.74 -2.61 ... ..$ counts : int [1:99] 5 23 34 62 88 114 141 197 224 319 ... ..$ density : num [1:99] 0.00126 0.00582 0.0086 0.01568 0.02226 ... ..$ mids : num [1:99] -3.05 -2.93 -2.8 -2.67 -2.54 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ mufa :List of 6 ..$ breaks : num [1:100] -2.2 -2.03 -1.86 -1.68 -1.51 ... ..$ counts : int [1:99] 9 38 152 388 715 1168 1523 1915 2313 2320 ... ..$ density : num [1:99] 0.0017 0.00717 0.02869 0.07324 0.13496 ... ..$ mids : num [1:99] -2.11 -1.94 -1.77 -1.6 -1.43 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sfa :List of 6 ..$ breaks : num [1:100] -2.69 -2.54 -2.39 -2.25 -2.1 ... ..$ counts : int [1:99] 5 6 18 39 98 147 253 443 645 870 ... ..$ density : num [1:99] 0.0011 0.00132 0.00397 0.0086 0.02161 ... ..$ mids : num [1:99] -2.61 -2.47 -2.32 -2.17 -2.03 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw3_fa :List of 6 ..$ breaks : num [1:100] -3.17 -3.01 -2.84 -2.68 -2.51 ... ..$ counts : int [1:99] 54 26 26 41 49 51 59 127 216 457 ... ..$ density : num [1:99] 0.0106 0.0051 0.0051 0.00805 0.00962 ... ..$ mids : num [1:99] -3.09 -2.92 -2.76 -2.6 -2.43 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ faw6_fa :List of 6 ..$ breaks : num [1:100] -5.95 -5.85 -5.75 -5.66 -5.56 ... ..$ counts : int [1:99] 1 0 0 0 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.000339 0 0 0 0 ... ..$ mids : num [1:99] -5.9 -5.8 -5.71 -5.61 -5.52 ...

..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ pufa_fa :List of 6 ..$ breaks : num [1:100] -5.1 -4.98 -4.86 -4.74 -4.61 ... ..$ counts : int [1:99] 2 1 1 1 2 1 2 6 7 10 ... ..$ density : num [1:99] 0.000533 0.000266 0.000266 0.000266 0.000533 ... ..$ mids : num [1:99] -5.04 -4.92 -4.8 -4.68 -4.55 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ mufa_fa :List of 6 ..$ breaks : num [1:100] -4.59 -4.49 -4.38 -4.27 -4.16 ... ..$ counts : int [1:99] 1 0 0 1 0 0 0 0 0 0 ... ..$ density : num [1:99] 0.000302 0 0 0.000302 0 ... ..$ mids : num [1:99] -4.54 -4.43 -4.32 -4.22 -4.11 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ sfa_fa :List of 6 ..$ breaks : num [1:100] -4.68 -4.58 -4.48 -4.37 -4.27 ... ..$ counts : int [1:99] 1 0 1 0 0 0 0 0 0 1 ... ..$ density : num [1:99] 0.000314 0 0.000314 0 0 ... ..$ mids : num [1:99] -4.63 -4.53 -4.43 -4.32 -4.22 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ unsatdeg :List of 6 ..$ breaks : num [1:100] -5.13 -5 -4.87 -4.74 -4.61 ... ..$ counts : int [1:99] 1 0 1 0 0 1 1 2 1 4 ... ..$ density : num [1:99] 0.000252 0 0.000252 0 0 ... ..$ mids : num [1:99] -5.06 -4.94 -4.81 -4.68 -4.55 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram" $ xxl_vldl_l:List of 6 ..$ breaks : num [1:100] -1.0243 -0.7878 -0.5513 -0.3147 -0.0782 ... ..$ counts : int [1:99] 4020 4941 5370 4292 3172 2330 1687 1205 992 702 ... ..$ density : num [1:99] 0.547 0.673 0.731 0.584 0.432 ... ..$ mids : num [1:99] -0.906 -0.67 -0.433 -0.196 0.04 ... ..$ xname : chr "na.omit(x)" ..$ equidist: logi TRUE ..- attr(*, "class")= chr "histogram"

Examples

```
data(BBMRI_hist_scaled)
## maybe str(BBMRI_hist_scaled) ; plot(BBMRI_hist_scaled) ...
```

`binarize_all_pheno` *binarize_all_pheno*

Description

Helper function created to binarize the phenotypes used to calculate the metabolomics based surrogate made by Bizzarri et al.

Usage

```
binarize_all_pheno(data)
```

Arguments

`data`	phenotypes data.frame containing some of the following variables (with the same namenclature): "sex","diabetes", "lipidmed", "blood_pressure_lowering_med", "current_smoking", "metabolic_syndrome", "alcohol_consumption", "age","BMI", "ln_hscrp","waist_circumference", "weight","height", "triglycerides", "ldl_chol", "hdlchol", "totchol", "eGFR","wbc","hgb"

Details

Bizzarri et al. built multivariate models,using 56 metabolic features quantified by Nightingale, to predict the 19 binary characteristics of an individual. The binary variables are: sex, diabetes status, metabolic syndrome status, lipid medication usage, blood pressure lowering medication, current smoking, alcohol consumption, high age, middle age, low age, high hsCRP, high triglycerides, high ldl cholesterol, high total cholesterol, low hdl cholesterol, low eGFR, low white blood cells, low hemoglobin levels.

Value

The phenotypic variables binarized following the thresholds in in the metabolomics surrogates made by by Bizzarri et al.

References

This function was made to binarize the variables following the same rules indicated in the article: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

See Also

pheno_barplots

Examples

```
library(MiMIR)

#load the phenotypes dataset
phenotypes <- synthetic_phenotypic_dataset
#Calculate BMI, LDL cholesterol and eGFR
binarized_phenotypes<-binarize_all_pheno(phenotypes)
```

BMI_LDL_eGFR *BMI_LDL_eGFR*

Description

#' Function created to calculate: 1) BMI using height and weight; 2) LDL cholesterol using HDL cholesterol, triglycerides, totchol; 3) eGFR creatinine levels, sex and age.

Usage

```
BMI_LDL_eGFR(phenotypes, metabo_measures)
```

Arguments

phenotypes	data.frame containing height and weight, HDL cholesterol, triglycerides, totchol, sex and age
metabo_measures	numeric data-frame with Nightingale metabolomics quantifications containing creatinine levels (crea)

Value

phenotypes data.frame with the addition of BMI, LDL cholesterol and eGFR

References

This function is constructed to calculate BMI, LDL cholesterol and eGFR as in the following papers:

BMI: Flint AJ, Rexrode KM, Hu FB, Glynn RJ, Caspard H, Manson JE et al. Body mass index, waist circumference, and risk of coronary heart disease: a prospective study among men and women. Obes Res Clin Pract 2010; 4: e171–e181, doi: 10.1016/j.orcp.2010.01.001

LDL-cholesterol: Friedewald WT, Levy RI, Fredrickson DS. Estimation of the Concentration of Low-Density Lipoprotein Cholesterol in Plasma, Without Use of the Preparative Ultracentrifuge. Clin Chem 1972; 18: 499–502, doi: https://doi.org/10.1093/clinchem/18.6.499

eGFR: Carrero Juan Jesus, Andersson Franko Mikael, Obergfell Achim, Gabrielsen Anders, Jernberg Tomas. hsCRP Level and the Risk of Death or Recurrent Cardiovascular Events in Patients With Myocardial Infarction: a Healthcare-Based Study. J Am Heart Assoc 2019; 8: e012638, doi: 10.1161/JAHA.119.012638

Examples

```
library(MiMIR)

#load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset
#Calculate BMI, LDL cholesterol and eGFR
phenotypes<-BMI_LDL_eGFR(phenotypes, metabolic_measures)
```

---

c21 *Colors related to each score*

Description

A data.frame containing colors related to the scores for the plots.

Usage

```
data("c21")
```

Format

A data frame with 0 observations on the following 2 variables.

`x` a numeric vector

`y` a numeric vector

Examples

```
data(c21)
## maybe str(c21) ; plot(c21) ...
```

---

calculate_surrogate_scores

calculate_surrogate_scores

Description

Function to compute the surrogate scores by Bizzarri et al. from the Nightingale metabolomics matrix

Usage

```
calculate_surrogate_scores(
  met,
  pheno,
  PARAM_surrogates,
  bin_names = c("sex", "diabetes"),
  Nmax_miss = 1,
  Nmax_zero = 1,
  post = TRUE,
  roc = FALSE,
  quiet = FALSE
)
```

Arguments

`met`	numeric data-frame with Nightingale-metabolomics
`pheno`	phenotypic data.frame including this clinical variables (with the same nomenclature): "sex","diabetes", "lipidmed", "blood_pressure_lowering_med", "current_smoking", "metabolic_syndrome", "alcohol_consumption", "age","BMI", "ln_hscrp","waist_circumference", "weight","height", "triglycerides", "ldl_chol", "hdlchol", "totchol", "eGFR","wbc","hgb"
`PARAM_surrogates`	list containing the parameters to compute the metabolomics-based surrogates
`bin_names`	vector of strings containing the names of the binary variables

Nmax_miss	numeric value indicating the maximum number of missing values allowed per sample (Number suggested=1)
Nmax_zero	numeric value indicating the maximum number of zeros allowed per sample (Number suggested=1)
post	logical to indicate if the function should calculate the posterior probabilities
roc	logical to plot ROC curves for the metabolomics surrogate (available only for the phenotypes included)
quiet	logical to suppress the messages in the console

Details

Bizzarri et al. built multivariate models,using 56 metabolic features quantified by Nightingale, to predict the 19 binary characteristics of an individual. The binary variables are: sex, diabetes status, metabolic syndrome status, lipid medication usage, blood pressure lowering medication, current smoking, alcohol consumption, high age, middle age, low age, high hsCRP, high triglycerides, high ldl cholesterol, high total cholesterol, low hdl cholesterol, low eGFR, low white blood cells, low hemoglobin levels.

Value

if pheno is not available: list with the surrogates and the Nightingale metabolomics matrix after QC. if pheno is available: list with the surrogates, ROC curves, phenotypes, binarized phenotypes and the Nightingale metabolomics matrix after QC,

References

This function was made to vidualize the binarized variables calculated following the rules indicated in the article: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

See Also

QCprep_surrogates

Examples

```
require(MiMIR)
require(foreach)
require(pROC)
require(foreach)

#load dataset
m <- synthetic_metabolic_dataset
p <- synthetic_phenotypic_dataset
#Apply the surrogates
sur<-calculate_surrogate_scores(met=m,pheno=p,MiMIR::PARAM_surrogates,bin_names=c("sex","diabetes"))
```

comp.CVD_score *comp.CVD_score*

Description

Function to compute CVD-score made by Peter Wurtz et al. made by Deelen et al. on Nightingale metabolomics data-set.

Usage

```
comp.CVD_score(met, phen, betas, quiet = FALSE)
```

Arguments

met	numeric data-frame with Nightingale-metabolomics
phen	data-frame containing phenotypic information of the samples (specifically: sex, systolic_blood_pressure, current_smoking, diabetes, blood_pressure_lowering_med, lipidmed, totchol, and hdlchol)
betas	The betas of the linear regression composing the CVD-score
quiet	logical to suppress the messages in the console

Details

This multivariate model predicts all-cause mortality at 5 or 10 years better than clinical variables normally associated with mortality. It is constituted of 14 metabolic features quantified by Nightingale Health. It was originally trained using a stepwise Cox regression analysis in a meta-analysis on 12 cohorts composed by 44,168 individuals. A multi biomarker score indicating cardiovascular risk and build by Würtz et al. It is calculated using information on sex, systolic blood pressure, current smoking, prevalent diabetes, antihypertensive medication, lipid lowering medication, total cholesterol, hdl cholesterol, together with phe, mufa_fa,faw6, and dha quantified by Nightingale. It was trained using a Cox proportional hazard regression model.

Value

data-frame containing the value of the CVD-score on the uploaded data-set

References

This function is constructed to be able to apply the CVD-score as described in: Würtz,P. et al. (2015) Metabolite profiling and cardiovascular event risk: a prospective study of 3 population-based cohorts. Circulation, 131, 774–785, doi: 10.1161/CIRCULATIONAHA.114.013116

See Also

prep_met_for_scores, CVD_score_betas, comp.T2D_Ahola_Olli, comp.mort_score

Examples

```
library(MiMIR)

#load the dataset
met <- synthetic_metabolic_dataset
phen<-synthetic_phenotypic_dataset
#Prepare the metabolic features fo the mortality score
CVDscore<-comp.CVD_score(met= met, phen=phen, betas=MiMIR::CVD_score_betas, quiet=TRUE)
```

`comp.mort_score` *comp.mort_score*

Description

Function to compute the mortality score made by Deelen et al. on Nightingale metabolomics data-set.

Usage

```
comp.mort_score(dat, betas = mort_betas, quiet = FALSE)
```

Arguments

`dat`	numeric data-frame with Nightingale-metabolomics
`betas`	data.frame containing the coefficients used for the regression of the mortality score
`quiet`	logical to suppress the messages in the console

Details

This multivariate model predicts all-cause mortality at 5 or 10 years better than clinical variables normally associated with mortality. It is constituted of 14 metabolic features quantified by Nightingale Health. It was originally trained using a stepwise Cox regression analysis in a meta-analysis on 12 cohorts composed by 44,168 individuals.

Value

data-frame containing the value of the mortality score on the uploaded data-set

References

This function is constructed to be able to apply the mortality score as described in: Deelen,J. et al. (2019) A metabolic profile of all-cause mortality risk identified in an observational study of 44,168 individuals. Nature Communications, 10, 1–8, doi: 10.1038/s41467-019-11311-9.

See Also

prep_met_for_scores, mort_betas, comp.T2D_Ahola_Olli, comp.CVD_score

Examples

```
library(MiMIR)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset
#Prepare the metabolic features fo the mortality score
mortScore<-comp.mort_score(metabolic_measures,quiet=TRUE)
```

---

comp.T2D_Ahola_Olli *comp.T2D_Ahola_Olli*

Description

Function to compute the T2D score made by Ahola Olli et al. on Nightingale metabolomics data-set.

Usage

```
comp.T2D_Ahola_Olli(met, phen, betas, quiet = FALSE)
```

Arguments

met	numeric data-frame with Nightingale-metabolomics
phen	data-frame containing phenotypic information of the samples (in particular: sex, age, BMI and the clinically measured glucose)
betas	The betas of the linear regression composing the T2D-score
quiet	logical to suppress the messages in the console

Details

This metabolomics-based score is associated with incident Type 2 Diabetes, made by Ahola-Olli et al. It is constructed using phe, l_vldl_ce_percentage and l_hdl_fc quantified by Nightingale Health, and some phenotypic information: sex, age, BMI, fasting glucose. It was trained using a stepwise logistic regression on 3 cohorts.

Value

data-frame containing the value of the T2D-score on the uploaded data-set

References

This function is constructed to be able to apply the T2D-score as described in: Ahola-Olli,A.V. et al. (2019) Circulating metabolites and the risk of type 2 diabetes: a prospective study of 11,896 young adults from four Finnish cohorts. Diabetologia, 62, 2298–2309, doi: 10.1007/s00125-019-05001-w

See Also

prep_met_for_scores, Ahola_Olli_betas, comp.mort_score, comp.CVD_score

Examples

```
library(MiMIR)

#load the dataset
met <- synthetic_metabolic_dataset
phen<-synthetic_phenotypic_dataset
#Prepare the metabolic features fo the mortality score
T2Dscore<-comp.T2D_Ahola_Olli(met= met, phen=phen,betas=MiMIR::Ahola_Olli_betas, quiet=TRUE)
```

---

comp_covid_score *comp_covid_score*

Description

Function to compute the COVID severity score made by Nightingale Health UK Biobank Initiative et al. on Nightingale metabolomics data-set.

Usage

```
comp_covid_score(dat, betas = MiMIR::covid_betas, quiet = FALSE)
```

Arguments

dat	numeric data-frame with Nightingale-metabolomics
betas	data.frame containing the coefficients used for the regression of the COVID-score
quiet	logical to suppress the messages in the console

Details

Multivariate model predicting the risk of severe COVID-19 infection. It is based on 37 metabolic features and trained using LASSO regression on 52,573 samples from the UK-biobanks.

Value

data-frame containing the value of the COVID-score on the uploaded data-set

References

This function is constructed to be able to apply the COVID-score as described in: Nightingale Health UK Biobank Initiative et al. (2021) Metabolic biomarker profiling for identification of susceptibility to severe pneumonia and COVID-19 in the general population. eLife, 10, e63033, doi: 10.7554/eLife.63033

See Also

prep_data_COVID_score, covid_betas, comp.mort_score

Examples

```
library(MiMIR)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset

#Compute the mortality score
mortScore<-comp_covid_score(dat=metabolic_measures, quiet=TRUE)
```

---

cor_assoc *cor_assoc*

Description

Function to calulate the correlation between 2 matrices

Usage

```
cor_assoc(dat1, dat2, feat1, feat2, method = "pearson", quiet = FALSE)
```

Arguments

dat1	matrix 1
dat2	matrix 2
feat1	vector of strings with the names of the selected variables in dat
feat2	vector if strings with the names of the selected variables in dat2
method	indicates which methods of the correlation to use
quiet	logical to suppress the messages in the console

Value

correlations of the selected variables in the 2 martrices

See Also

plot_corply

Examples

```
library(stats)

#load the dataset
m <- as.matrix(synthetic_metabolic_dataset)

#Compute the pearson correlation of all the variables in the data.frame metabolic_measures
cors<-cor_assoc(m, m, MiMIR::metabolites_subsets$MET63,MiMIR::metabolites_subsets$MET63)
```

`covid_betas` *Parameters to calculate the susceptibility to CODIV-19 score*

Description

Abbreviatinos of the metabolites, metabolites and coefficients of the score

Usage

```
data("covid_betas")
```

Format

A data frame with 25 observations on the following 3 variables.

`Abbreviation` a character vector

`Metabolite` a character vector

`Beta_value` a numeric vector

Source

https://elifesciences.org/articles/63033

Examples

```
data(covid_betas)
## maybe str(covid_betas) ; plot(covid_betas) ...
```

CVD_score_betas *Parameters to calculate the Cardiovascular disease score*

Description

Abbreviatinos of the metabolites, metabolites and coefficients of the score

Usage

```
data("CVD_score_betas")
```

Format

A data frame with 12 observations on the following 3 variables.

`Abbreviation` a character vector

`Metabolite` a character vector

`Beta_value` a numeric vector

Source

https://pubmed.ncbi.nlm.nih.gov/25573147/

Examples

```
data(CVD_score_betas)
## maybe str(CVD_score_betas) ; plot(CVD_score_betas) ...
```

find_BBMRI_names *find_BBMRI_names*

Description

Function to translate Nightingale metabolomics alternative metabolite names to the ones used in BBMRI-nl

Usage

```
find_BBMRI_names(names)
```

Arguments

`names`	vector of strings with the metabolic features names to be translated

Value

data.frame with the uploaded metabolites names on the first column and the BBMRI names on the second column.

References

This is a function originally created for the package ggforestplot and modified ad hoc for our package (https://nightingalehealth.github.io/ggforestplot/articles/index.html).

Examples

```
library(MiMIR)
library(purrr)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset
#Find the metabolites names used in BBMRI-nl
nam<-find_BBMRI_names(colnames(metabolic_measures))
```

`hist_plots` *hist_plots*

Description

#' Function to plot the histograms for all the variables in dat

Usage

```
hist_plots(
  dat,
  x_name,
  color = MiMIR::c21,
  scaled = FALSE,
  datatype = "metabolic score",
  main = "Predictors Distributions"
)
```

Arguments

`dat`	data.frame or matrix with the variables to plot
`x_name`	string with the names of the selected variables in dat
`color`	colors selected for all the variables
`scaled`	logical to z-scale the variables
`datatype`	a character vector indicating what data type is beeing plotted
`main`	title of the plot

Value

plotly image with the histograms of the selected variables

Examples

```
require(MiMIR)
require(plotly)
require(matrixStats)
#load the metabolites dataset
m <- synthetic_metabolic_dataset

#Apply a surrogate models and plot the ROC curve
surrogates<-calculate_surrogate_scores(m, PARAM_surrogates=MiMIR::PARAM_surrogates, roc=FALSE)
#Plot the histogram of the surrogate sex values scaled
hist_plots(surrogates$surrogates, x_name="s_sex", scaled=TRUE)
```

---

hist_plots_mortality *hist_plots_mortality*

Description

#' Function to plot the histogram of the mortality score separated for different age ranges as a plotly image

Usage

```
hist_plots_mortality(mort_score, phenotypes)
```

Arguments

mort_score	data.frame containing the mortality score
phenotypes	data.frame containing age

Value

plotly image with the histogram of the mortality score separated in 3 age ranges

Examples

```
library(MiMIR)
library(plotly)
#' #load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset

#Compute the mortality score
mortScore<-comp.mort_score(metabolic_measures,quiet=TRUE)
#Plot the mortality score histogram at different ages
```

```
hist_plots_mortality(mortScore, phenotypes)
```

---

kapmeier_scores *kapmeier_scores*

Description

#' Function that creates a Kaplan Meier comparing first and last tertile of a metabolic score

Usage

```
kapmeier_scores(predictors, pheno, score, Eventname = "Event")
```

Arguments

predictors	The data.frame containing the predictors
pheno	The data.frame containing the phenotypes
score	a character string indicating which predictor to use
Eventname	a character string with the name of the event to print on the plot

Value

plotly with a Kaplan Meier comparing first and last tertile of a metabolic score

Examples

```
require(MiMIR)
require(plotly)
require(survminer)
require(ggfortify)
require(ggplot2)

#load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset

#Compute the mortality score
mortScore<-comp.mort_score(metabolic_measures,quiet=TRUE)

#Plot a Kaplan Meier
kapmeier_scores(predictors=mortScore, pheno=phenotypes, score="mortScore")
```

LOBOV_accuracies *LOBOV_accuracies*

Description

Function created to visualize the accuracies in the current dataset compared to the accuracies in the Leave One Biobank Out Validation in Bizzarri et al.

Usage

```
LOBOV_accuracies(surrogates, bin_phenotypes, bin_pheno_available, acc_LOBOV)
```

Arguments

surrogates	numeric data.frame containing the surrogate values by Bizzarri et al.
bin_phenotypes	numeric data.frame with the binarized phenotypes output of binarize_all_pheno
bin_pheno_available	vector of strings with the available phenotypes
acc_LOBOV	accuracy of LOBOV calculated in Bizzarri et al.

Details

Comparison of the AUCs of the surrogates in the updated dataset and the results of the Leave One Biobank Out Validation made in BBMRI-nl.

Value

Boxplot with the accuracies of the LOBOV

References

This function was made to vidualize the binarized variables calculated following the rules indicated in the article: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

Examples

```
require(pROC)
require(plotly)
require(MiMIR)
require(foreach)
require(ggplot2)

#load the dataset
m <- synthetic_metabolic_dataset
p<- synthetic_phenotypic_dataset

#Calculating the binarized surrogates
```

```
b_p<-binarize_all_pheno(p)
#Apply a surrogate models and plot the ROC curve
sur<-calculate_surrogate_scores(m, p, MiMIR::PARAM_surrogates, bin_names=colnames(b_p))
p_avail<-colnames(b_p)[c(1:5)]
LOBOV_accuracies(sur$surrogates, b_p, p_avail, MiMIR::acc_LOBOV)
```

`metabolites_subsets` *The subsets of the Nightingale metabolomics platform used in the application MiMIR*

Description

Subsets of the metabolites for the app

Usage

```
data("metabolites_subsets")
```

Format

The format is: List of 8 $: chr [1:63] "ala" "gln" "his" "phe" ... $: chr [1:62] "ala" "gln" "his" "phe" ... $: chr [1:57] "ala" "gln" "his" "phe" ... $: chr [1:56] "ala" "gln" "his" "phe" ... $: chr [1:14] "pufa_fa" "gp" "glc" "s_hdl_l" ... $: chr [1:25] "gp" "dha" "crea" "mufa" ... $: chr [1:3] "phe" "l_vldl_ce_percentage" "l_hdl_fc" $: chr [1:4] "phe" "mufa_fa" "faw6" "dha"

Examples

```
data(metabolites_subsets)
## maybe str(metabolites_subsets) ; plot(metabolites_subsets) ...
```

`MetaboWAS` *MetaboWAS*

Description

Function to calculate a Metabolome Wide Association study

Usage

```
MetaboWAS(met, pheno, test_variable, covariates, img = TRUE, adj_method = "BH")
```

Arguments

met	numeric data.frame with the metabolomics features
pheno	data.frame containing the phenotype of interest
test_variable	string vector with the name of the phenotype of interest
covariates	string vector with the name of the variables to be added as a covariate
img	logical indicating if the function should plot a Manhattan plot
adj_method	multiple testing correction method

Details

This is a function to compute linear associations individually for each variable in the first data.frame with the test variable and corrected for the selected covariates. This function to computes linear regression modelindividually for each variable in the first data.frame with the test variable and adjusted for potential confounders. False Discovery Rate (FDR) is applied to account for multiple testing correction. The user has the faculty to select the test variable and the potential covariates within the pool of variables in the phenotypic file input. The results of the associations are reported in a Manhattan plot

The p-value of the association is then corrected using Benjamini Hochberg. Finally we use plotly to plot a Manhattan Plot, which reports on the x-axis the list of metabolites reported in the Nightingale Health, divided in groups, and on the y-axis the -log (adjusted p-value).

Value

res= the results of the MetaboWAS, manhplot= the Manhattan plot made with plotly, N_hits= the number of significant hits

References

This method is also described and used in: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

Examples

```
require(MiMIR)
require(plotly)
require(ggplot2)

#' #load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset

#Computing a MetaboWAS for age corrected by sex
MetaboWAS(met=metabolic_measures, pheno=phenotypes, test_variable="age", covariates= "sex")
```

metabo_names_translator *Metabolites names translators*

Description

This data.frame is used to translate alternative names of Nightingale dataset to the metabolites names used in BBMRI-nl

Usage

```
data("metabo_names_translator")
```

Format

A data frame with 228 observations on the following 9 variables.

`abbreviation` a character vector

`machine_readable_name` a character vector

`name` a character vector

`description` a character vector

`alternative_names` a list vector

`group` a character vector

`subgroup` a character vector

`unit` a character vector

`BBMRI_names` a character vector

Details

This file contains: abbreviations of the metabolites names, machine_readable_name are the same abbreviations without spaces or other non readable characters name is the full name of the metabolites description what the metabolic feature represents alternative_names other names used by Nightingale Health to indicate the same metabolites group what group of variables it is part of subgroup of the variables unit measure of the metabolites BBMMRI_names which are the names used in BBMRI for those variables

Source

This file was first created in the package ggforestplot and then modified to include the names used in BBMRI-nl.

References

https://nightingalehealth.github.io/ggforestplot/articles/ggforestplot.html

Examples

```
data(metabo_names_translator)
```

`mort_betas` *Parameters to calculate the mortality score by Deelen et al.*

Description

Abbreviatinos of the metabolites, metabolites and coefficients of the score

Usage

```
data("mort_betas")
```

Format

A data frame with 14 observations on the following 3 variables.

`Abbreviation` a character vector

`Metabolite` a character vector

`Beta_value` a numeric vector

Source

https://www.nature.com/articles/s41467-019-11311-9

Examples

```
data(mort_betas)
## maybe str(mort_betas) ; plot(mort_betas) ...
```

`multi_hist` *multi_hist*

Description

#' Function to plot the histograms for all the variables in dat

Usage

```
multi_hist(dat, color = MiMIR::c21, scaled = FALSE)
```

Arguments

`dat`	data.frame or matrix with the variables to plot
`color`	colors selected for all the variables
`scaled`	logical to z-scale the variables

Value

plotly image with the histograms for all the variables in dat

Examples

```
library(plotly)
library(MiMIR)

#load the dataset
metabolic_measures <- synthetic_metabolic_dataset

multi_hist(metabolic_measures[,MiMIR::metabolites_subsets$MET14], scaled=T)
```

PARAM_metaboAge *Parameters to calculate MetaboAge*

Description

This list contains all the informations to calculate the MetaboAge by van den Akker et al.

Usage

```
data("PARAM_metaboAge")
```

Format

The format is: List of 8 $ MET : chr [1:56] "acace" "ace" "ala" "alb" ... $ Nmax_miss: num 1 $ Nmax_zero: num 1 $ logMEAN : Named num [1:56] -3.37 -3.173 -1.233 -2.437 0.457 ... ..- attr(*, "names")= chr [1:56] "acace" "ace" "ala" "alb" ... $ logSD : Named num [1:56] 0.747 0.391 0.259 0.084 0.126 ... ..- attr(*, "names")= chr [1:56] "acace" "ace" "ala" "alb" ... $ MEAN : Named num [1:56] 0.0432 0.0445 0.3006 0.0877 1.5945 ... ..- attr(*, "names")= chr [1:56] "acace" "ace" "ala" "alb" ... $ SD : Named num [1:56] 0.03465 0.01919 0.0767 0.00686 0.20025 ... ..- attr(*, "names")= chr [1:56] "acace" "ace" "ala" "alb" ... $ FIT_COEF : Named num [1:57] 58.619 1.056 0.689 -0.377 -2.843 ... ..- attr(*, "names")= chr [1:57] "(Intercept)" "acace" "ace" "ala" ...

Details

MET=Metabolites that are necessary for the logistic regression. Nmax_miss and Nmax_zero= the max number of missing and zeros raccommended for each sample. mean and sd= mean and standard deviation of each metabolites in BBMRI-NL. model_betas= a matrix with coefficients for the pre-trained models

Source

https://www.ahajournals.org/doi/full/10.1161/CIRCGEN.119.002610

Examples

```
data(PARAM_metaboAge)
## maybe str(PARAM_metaboAge) ; plot(PARAM_metaboAge) ...
```

PARAM_surrogates *Parameters to calculate the surrogate variables*

Description

This list contains all the informations to calculate the surrogate clinical variables by Bizzarri et al.

Usage

```
data("PARAM_surrogates")
```

Format

The format is: List of 6 $ MET : chr [1:56] "ala" "gln" "his" "phe" ... $ Nmax_miss : num 1 $ Nmax_zero : num 1 $ mean :'data.frame': 1 obs. of 56 variables: ..$ ala : num 0.315 ..$ gln : num 0.454 ..$ his : num 0.0581 ..$ phe : num 0.0472 ..$ tyr : num 0.0604 ..$ ile : num 0.0519 ..$ leu : num 0.0622 ..$ val : num 0.156 ..$ glc : num 4.68 ..$ lac : num 1.29 ..$ cit : num 0.103 ..$ ace : num 0.0455 ..$ acace : num 0.0423 ..$ crea : num 0.0709 ..$ alb : num 0.0872 ..$ gp : num 1.34 ..$ m_vldl_l : num 0.647 ..$ s_vldl_l : num 0.68 ..$ xs_vldl_l: num 0.554 ..$ idl_l : num 1.09 ..$ l_ldl_l : num 1.21 ..$ m_ldl_l : num 0.691 ..$ s_ldl_l : num 0.446 ..$ m_hdl_l : num 0.825 ..$ s_hdl_l : num 1.03 ..$ idl_c : num 0.695 ..$ serum_c : num 4.47 ..$ vldl_c : num 0.838 ..$ ldl_c : num 1.55 ..$ hdl_c : num 1.38 ..$ hdl2_c : num 0.907 ..$ hdl3_c : num 0.476 ..$ vldl_d : num 36.8 ..$ ldl_d : num 23.6 ..$ hdl_d : num 9.96 ..$ serum_tg : num 1.38 ..$ totpg : num 1.92 ..$ pc : num 1.99 ..$ sm : num 0.459 ..$ totcho : num 2.33 ..$ apoa1 : num 1.58 ..$ apob : num 0.963 ..$ totfa : num 11.6 ..$ dha : num 0.141 ..$ la : num 3.05 ..$ faw3 : num 0.416 ..$ faw6 : num 3.83 ..$ pufa : num 4.25 ..$ mufa : num 2.97 ..$ sfa : num 4.36 ..$ faw3_fa : num 3.61 ..$ faw6_fa : num 33.4 ..$ pufa_fa : num 37 ..$ mufa_fa : num 25.4 ..$ sfa_fa : num 37.7 ..$ unsatdeg : num 1.21 $ sd :'data.frame': 1 obs. of 56 variables: ..$ ala : num 0.0794 ..$ gln : num 0.0761 ..$ his : num 0.014 ..$ phe : num 0.011 ..$ tyr : num 0.015 ..$ ile : num 0.0193 ..$ leu : num 0.0163 ..$ val : num 0.037 ..$ glc : num 1.56 ..$ lac : num 1.06 ..$ cit : num 0.0324 ..$ ace : num 0.0203 ..$ acace : num 0.0324 ..$ crea : num 0.0181 ..$ alb : num 0.00688 ..$ gp : num 0.205 ..$ m_vldl_l : num 0.37 ..$ s_vldl_l : num 0.236 ..$ xs_vldl_l: num 0.131 ..$ idl_l : num 0.282 ..$ l_ldl_l : num 0.353 ..$ m_ldl_l : num 0.216 ..$ s_ldl_l : num 0.131 ..$ m_hdl_l : num 0.172 ..$ s_hdl_l : num 0.123 ..$ idl_c : num 0.195 ..$ serum_c : num 0.986 ..$ vldl_c : num 0.289 ..$ ldl_c : num 0.524 ..$ hdl_c : num 0.333 ..$ hdl2_c : num 0.31 ..$ hdl3_c : num 0.0625 ..$ vldl_d : num 1.35 ..$ ldl_d : num 0.134 ..$ hdl_d : num 0.259 ..$ serum_tg : num 0.661 ..$ totpg : num 0.36 ..$ pc : num 0.366 ..$ sm : num 0.087 ..$ totcho : num 0.381 ..$ apoa1 : num 0.202 ..$ apob : num 0.22 ..$ totfa : num 2.44 ..$ dha : num 0.0548 ..$ la : num 0.684 ..$ faw3 : num 0.136 ..$ faw6 : num 0.756 ..$ pufa : num 0.831 ..$ mufa : num 0.899 ..$ sfa : num 0.951 ..$ faw3_fa : num 0.948 ..$ faw6_fa : num 3.53 ..$ pufa_fa : num 3.63 ..$ mufa_fa : num 3.67 ..$ sfa_fa : num 1.89 ..$ unsatdeg : num 0.0722 $ models_betas: num [1:19, 1:57] -0.103 -2.397 -1.33 -0.165 -0.546 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:19] "s_sex" "s_diabetes" "s_lipidmed" "s_blood_pressure_lowering_med" ... .$: chr [1:57] "(Intercept)" "ala" "gln" "his" ...

Details

MET=Metabolites that are necessary for the logistic regression. Nmax_miss and Nmax_zero= the max number of missing and zeros raccommended for each sample. mean and sd= mean and standard deviation of each metabolites in BBMRI-NL. model_betas= a matrix with coefficients for the pre-trained models

Source

https://www.medrxiv.org/content/10.1101/2021.07.19.21258470v1.article-info

Examples

```
data(PARAM_surrogates)
## maybe str(PARAM_surrogates) ; plot(PARAM_surrogates) ...
```

---

phenotypes_names *Nomenclatures for the phenotypes used in the app MiMIR*

Description

Phenotypes, Binary phenotypes,...

Usage

```
data("phenotypes_names")
```

Format

The format is: List of 5 $: chr [1:19] "sex" "diabetes" "lipidmed" "blood_pressure_lowering_med" ... $: chr [1:19] "s_sex" "s_diabetes" "s_lipidmed" "s_blood_pressure_lowering_med" ... $: chr [1:19] "sex" "diabetes" "lipidmed" "blood_pressure_lowering_med" ... $: chr [1:19] "s_sex" "s_diabetes" "s_lipidmed" "s_blood_pressure_lowering_med" ... $: chr [1:24] "sex" "diabetes" "lipidmed" "blood_pressure_lowering_med" ...

Examples

```
data(phenotypes_names)
## maybe str(phenotypes_names) ; plot(phenotypes_names) ...
```

pheno_barplots *pheno_barplots*

Description

#' Function created to binarize the phenotypes used to calculate the metabolomics based surrogate made by Bizzarri et al.

Usage

```
pheno_barplots(bin_phenotypes)
```

Arguments

bin_phenotypes	phenotypes data.frame containing some of the following variables (with the same namenclature): "sex","diabetes", "lipidmed", "blood_pressure_lowering_med", "current_smoking", "metabolic_syndrome", "alcohol_consumption", "age","BMI", "ln_hscrp","waist_circumference", "weight","height", "triglycerides", "ldl_chol", "hdlchol", "totchol", "eGFR","wbc","hgb"

Details

Bizzarri et al. built multivariate models,using 56 metabolic features quantified by Nightingale, to predict the 19 binary characteristics of an individual. The binary variables are: sex, diabetes status, metabolic syndrome status, lipid medication usage, blood pressure lowering medication, current smoking, alcohol consumption, high age, middle age, low age, high hsCRP, high triglycerides, high ldl cholesterol, high total cholesterol, low hdl cholesterol, low eGFR, low white blood cells, low hemoglobin levels.

Value

The phenotypic variables binarized following the thresholds in in the metabolomics surrogates made by by Bizzarri et al.

References

This function was made to vidualize the binarized variables calculated following the rules indicated in the article: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

See Also

binarize_all_pheno

Examples

```
require(MiMIR)
require(foreach)

#load the phenotypes dataset
phenotypes <- synthetic_phenotypic_dataset

#Calculate BMI, LDL cholesterol and eGFR
binarized_phenotypes<-binarize_all_pheno(phenotypes)
#Plot the variables
pheno_barplots(binarized_phenotypes)
```

`plattCalibration` *plattCalibration*

Description

Function that calculates the Platt Calibrations

Usage

```
plattCalibration(r.calib, p.calib, nbins = 10, pl = FALSE)
```

Arguments

`r.calib`	observed binary phenotype
`p.calib`	predicted probabilities
`nbins`	number of bins to create the plots
`pl`	logical indicating if the function should plot the Reliability diagram and histogram of the calibrations

Details

Many popular machine learning algorithms produce inaccurate predicted probabilities, especially when applied on a dataset different than the training set. Platt (1999) proposed an adjustment, in which the original probabilities are used as a predictor in a single-variable logistic regression to produce more accurate adjusted predicted probabilities. The function will also help the evaluation of the calibration, by plotting: reliability diagrams and distributions of the calibrated and non-calibrated probabilities. The reliability diagrams plots the mean predicted value within a certain range of posterior probabilities, against the fraction of accurately predicted values. Finally, we also report accuracy measures for the calibrations: the ECE, MCE and the Log-Loss of the probabilities before and after calibration.

Value

list with samples, responses, calibrations, ECE, MCE and calibration plots if save==T

References

This is a function originally created for the package in eRic, under the name prCalibrate and modified ad hoc for our purposes (Github)

J. C. Platt, 'Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods', in Advances in Large Margin Classifiers, 1999, pp. 61–74.

Examples

```
library(stats)
library(plotly)

#load the dataset
met <- synthetic_metabolic_dataset
phen <- synthetic_phenotypic_dataset

#Calculating the binarized surrogates
b_phen<-binarize_all_pheno(phen)
#Apply a surrogate models and plot the ROC curve
surr<-calculate_surrogate_scores(met, phen,MiMIR::PARAM_surrogates, bin_names=colnames(b_phen))
#Calibration of the surrogate sex
real_data<-as.numeric(b_phen$sex)
pred_data<-surr$surrogates[,"s_sex"]
plattCalibration(r.calib=real_data, p.calib=pred_data, nbins = 10, pl=TRUE)
```

---

plot_corply *plot_corply*

Description

Function creating plottig the correlation between 2 datasets, dat1 x dat2 on basis of (partial) correlations

Usage

```
plot_corply(
  res,
  main = NULL,
  zlim = NULL,
  reorder.x = FALSE,
  reorder.y = reorder.x,
  resort_on_p = FALSE,
  abs = FALSE,
  cor.abs = FALSE,
  reorder_dend = FALSE
)
```

Arguments

res	associations obtained with cor.assoc
main	title of the plot
zlim	max association to plot
reorder.x	logical indicating if the function should reorder the x axis based on clustering
reorder.y	logical indicating if the function should reorder the y axis based on clustering
resort_on_p	logical indicating if the function should reorder x and y axis based on the pvalues of the associations
abs	logical indicating if the function should reorder based the absolute values
cor.abs	logical indicating if the function should reorder the plot base on the absolute values
reorder_dend	Tlogical indicating if the function should reorder the plot based on dendrogram

Value

heatmap with the results of cor.assoc

See Also

cor_assoc

Examples

```
library(stats)

#load the dataset
m <- as.matrix(synthetic_metabolic_dataset)

#Compute the pearson correlation of all the variables in the data.frame metabolic_measures
cors<-cor_assoc(m, m, MiMIR::metabolites_subsets$MET63,MiMIR::metabolites_subsets$MET63)
#Plot the correlations
plot_corply(cors, main="Correlations metabolites")
```

---

plot_na_heatmap *plot_na_heatmap*

Description

Function plotting information about missing & zero values on the indicated matrix.

Usage

```
plot_na_heatmap(dat)
```

Arguments

dat	The matrix or data.frame

Details

This heatmap indicates the available values in grey and missing or zeros in white. On the sides two bar plots on the sides, one showing the missingn or zero values per row and another to show the missing or zeroes per column.

Value

Plot with a central heatmap and two histogram on the sides

Examples

```
library(graphics)
library(MiMIR)

#load the metabolites dataset
metabolic_measures <- synthetic_metabolic_dataset
#Plot the missing values in the metabolomics matrix
plot_na_heatmap(metabolic_measures)
```

---

prep_data_COVID_score *prep_data_COVID_score*

Description

Helper function to pre-process the Nightingale Health metabolomics data-set before applying the COVID score.

Usage

```
prep_data_COVID_score(
  dat,
  featID = c("gp", "dha", "crea", "mufa", "apob_apoa1", "tyr", "ile", "sfa_fa", "glc",
   "lac", "faw6_faw3", "phe", "serum_c", "faw6_fa", "ala", "pufa", "glycine", "his",
    "pufa_fa", "val", "leu", "alb", "faw3", "ldl_c", "serum_tg"),
  quiet = FALSE
)
```

Arguments

dat	numeric data-frame with Nightingale-metabolomics
featID	vector of strings with the names of metabolic features included in the COVID-score
quiet	logical to suppress the messages in the console

Value

The Nightingale-metabolomics data-frame after pre-processing (checked for zeros, z-scaled and log-transformed) according to what has been done by the authors of the original papers.

References

This function is constructed to be able to follow the pre-processing steps described in: Nightingale Health UK Biobank Initiative et al. (2021) Metabolic biomarker profiling for identification of susceptibility to severe pneumonia and COVID-19 in the general population. eLife, 10, e63033, doi: 10.7554/eLife.63033

See Also

prep_met_for_scores, covid_betas, comp_covid_score

Examples

```
require(MiMIR)
require(matrixStats)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset
#Prepare the metabolic features fo the mortality score
prepped_met <- prep_data_COVID_score(dat=metabolic_measures)
```

`prep_met_for_scores` *prep_met_for_scores*

Description

Helper function to pre-process the Nightingale Health metabolomics data-set before applying the mortality, Type-2-diabetes and CVD scores.

Usage

```
prep_met_for_scores(dat, featID, plusone = FALSE, quiet = FALSE)
```

Arguments

`dat`	numeric data-frame with Nightingale-metabolomics
`featID`	vector of strings with the names of metabolic features included in the score selected
`plusone`	logical to determine if a value of 1.0 should be added to all metabolic features (TRUE) or only to the ones featuring zeros before log-transforming (FALSE)
`quiet`	logical to suppress the messages in the console

Value

The Nightingale-metabolomics data-frame after pre-processing (checked for zeros, zscale and log-transformed) according to what has been done by the authors of the original papers.

References

This function is constructed to be able to follow the pre-processing steps described in: Deelen,J. et al. (2019) A metabolic profile of all-cause mortality risk identified in an observational study of 44,168 individuals. Nature Communications, 10, 1–8, doi: 10.1038/s41467-019-11311-9.

Ahola-Olli,A.V. et al. (2019) Circulating metabolites and the risk of type 2 diabetes: a prospective study of 11,896 young adults from four Finnish cohorts. Diabetologia, 62, 2298–2309, doi: 10.1007/s00125-019-05001-w

Würtz,P. et al. (2015) Metabolite profiling and cardiovascular event risk: a prospective study of 3 population-based cohorts. Circulation, 131, 774–785, doi: 10.1161/CIRCULATIONAHA.114.013116

See Also

comp.mort_score, mort_betas, comp.T2D_Ahola_Olli, comp.CVD_score

Examples

```
library(MiMIR)

#load the Nightingale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset
#Prepare the metabolic features fo the mortality score
prepped_met <- prep_met_for_scores(metabolic_measures,featID=MiMIR::mort_betas$Abbreviation)
```

`QCprep` *QCprep*

Description

Helper function to pre-process the Nightingale Health metabolomics data-set before applying the MetaboAge score by van den Akker et al.

Usage

```
QCprep(mat, PARAM_metaboAge, quiet = TRUE, Nmax_zero = 1, Nmax_miss = 1)
```

Arguments

mat	numeric data-frame NH-metabolomics matrix.
PARAM_metaboAge	list containing all the parameters to compute the metaboAge (metabolic features list,BBMRI-nl means and SDs of the metabolic features, and coefficients)
quiet	logical to suppress the messages in the console
Nmax_zero	numberic value indicating the maximum number of zeros allowed per sample (Number suggested=1)
Nmax_miss	numberic value indicating the maximum number of missing values allowed per sample (Number suggested=1)

Value

Nightingale-metabolomics data-frame after pre-processing (checked for zeros, missing values, samples>5SD from the BBMRI-mean, imputing the missing values and z-scaled)

References

This function is constructed to be able to follow the pre-processing steps described in: van den Akker Erik B. et al. (2020) Metabolic Age Based on the BBMRI-NL 1H-NMR Metabolomics Repository as Biomarker of Age-related Disease. Circulation: Genomic and Precision Medicine, 13, 541–547, doi:10.1161/CIRCGEN.119.002610

See Also

apply.fit

Examples

```
library(MiMIR)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset

#Pre-process the metabolic features
prepped_met<-QCprep(as.matrix(metabolic_measures[,metabolites_subsets$MET63]), PARAM_metaboAge)
```

QCprep_surrogates *QCprep_surrogates*

Description

Helper function to pre-process the Nightingale Health metabolomics data-set before applying metabolomics-based surrogates by Bizzarri et al.

Usage

```
QCprep_surrogates(
  mat,
  PARAM_surrogates,
  Nmax_miss = 1,
  Nmax_zero = 1,
  quiet = FALSE
)
```

Arguments

mat	numeric data-frame Nightingale metabolomics matrix.
PARAM_surrogates	is a list holding the parameters to compute the surrogates
Nmax_miss	numeric value indicating the maximum number of missing values allowed per sample (Number suggested=1)
Nmax_zero	numeric value indicating the maximum number of zeros allowed per sample (Number suggested=1)
quiet	logical to suppress the messages in the console

Details

Bizzarri et al. built multivariate models,using 56 metabolic features quantified by Nightingale, to predict the 19 binary characteristics of an individual. The binary variables are: sex, diabetes status, metabolic syndrome status, lipid medication usage, blood pressure lowering medication, current smoking, alcohol consumption, high age, middle age, low age, high hsCRP, high triglycerides, high ldl cholesterol, high total cholesterol, low hdl cholesterol, low eGFR, low white blood cells, low hemoglobin levels.

Value

Nightingale-metabolomics data-frame after pre-processing (checked for zeros, missing values, samples>5SD from the BBMRI-mean, imputing the missing values and z-scaled)

References

This function was made to vidualize the binarized variables calculated following the rules indicated in the article: Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764

See Also

binarize_all_pheno

Examples

```
library(MiMIR)

#load the Nightignale metabolomics dataset
metabolic_measures <- synthetic_metabolic_dataset
#Pre-process the metabolic features
prepped_met<-QCprep_surrogates(as.matrix(metabolic_measures), MiMIR::PARAM_surrogates)
```

---

roc_surro *roc_surro*

Description

Function that creates a ROC curve of the selected metabolic surrogates as a plotly image

Usage

```
roc_surro(surrogates, bin_phenotypes, x_name)
```

Arguments

surrogates	numeric data.frame of metabolomics-based surrogate values by Bizzarri et al.
bin_phenotypes	logic data.frame of binarized phenotypes
x_name	vector of strings with the names of the selected binary phenotypes for the roc

Value

plotly image with the ROC curves for one or more selected variables

Examples

```
require(pROC)
require(plotly)
require(foreach)
require(MiMIR)

#load the dataset
met <- synthetic_metabolic_dataset
phen<- synthetic_phenotypic_dataset

#Calculating the binarized surrogates
b_phen<-binarize_all_pheno(phen)
#Apply a surrogate models and plot the ROC curve
surr<-calculate_surrogate_scores(met, phen, MiMIR::PARAM_surrogates, colnames(b_phen))
#Plot the ROC curves
roc_surro(surr$surrogates, b_phen, "sex")
```

roc_surro_subplots *roc_surro_subplots*

Description

Function that plots the ROCs of the surrogates of all the available surrogate models as plotly subplots

Usage

```
roc_surro_subplots(surrogates, bin_phenotypes)
```

Arguments

surrogates	numeric data.frame containing the surrogate values by Bizzarri et al.
bin_phenotypes	numeric data.frame with the binarized phenotypes output of binarize_all_pheno

Value

plotly image with all the ROCs for all the available clinical variables

Examples

```
library(pROC)
library(plotly)
library(MiMIR)

#load the dataset
met <- synthetic_metabolic_dataset
phen<- synthetic_phenotypic_dataset

#Calculating the binarized surrogates
b_phen<-binarize_all_pheno(phen)
#Apply a surrogate models and plot the ROC curve
surr<-calculate_surrogate_scores(met, phen, MiMIR::PARAM_surrogates, colnames(b_phen))

roc_surro_subplots(surr$surrogates, b_phen)
```

scatterplot_predictions *scatterplot_predictions*

Description

Function to visualize a scatter-plot comparing two variables

Usage

```
scatterplot_predictions(x, p, title, xname = "x", yname = "predicted x")
```

Arguments

x	numeric vector
p	second numeric vector
title	string vector with the title
xname	string vector with the name of the variable on the x axis
yname	string vector with the name of the variable on the y axis

Value

plotly image with the scatterplot

Examples

```
library(plotly)
#load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset

#Pre-process the metabolic features
prepped_met<-QCprep(as.matrix(metabolic_measures), MiMIR::PARAM_metaboAge)
#Apply the metaboAge
metaboAge<-apply.fit(prepped_met, FIT=PARAM_metaboAge$FIT_COEF)

age<-data.frame(phenotypes$age)
rownames(age)<-rownames(phenotypes)
scatterplot_predictions(age, metaboAge, title="Chronological Age vs MetaboAge")
```

`startApp` *startMiMIR*

Description

Start the application MiMIR.

Usage

```
startApp(launch.browser = TRUE)
```

Arguments

`launch.browser`	TRUE/FALSE

Details

This function starts the R-Shiny tool called MiMIR (Metabolomics-based Models for Imputing Risk), a graphical user interface that provides an intuitive framework for ad-hoc statistical analysis of Nightingale Health's 1H-NMR metabolomics data and allows for the projection and calibration of 24 pre-trained metabolomics-based models, without any pre-required programming knowledge.

Value

Opens application. If `launch.browser=TRUE` in default web browser

References

Deelen,J. et al. (2019) A metabolic profile of all-cause mortality risk identified in an observational study of 44,168 individuals. Nature Communications, 10, 1–8, doi: 10.1038/s41467-019-11311-9. Ahola-Olli,A.V. et al. (2019) Circulating metabolites and the risk of type 2 diabetes: a prospective study of 11,896 young adults from four Finnish cohorts. Diabetologia, 62, 2298–2309, doi: 10.1007/s00125-019-05001-w Würtz,P. et al. (2015) Metabolite profiling and cardiovascular event risk: a prospective study of 3 population-based cohorts. Circulation, 131, 774–785, doi: 10.1161/CIRCULATIONAHA.114.013116 Bizzarri,D. et al. (2022) 1H-NMR metabolomics-based surrogates to impute common clinical risk factors and endpoints. EBioMedicine, 75, 103764, doi: 10.1016/j.ebiom.2021.103764 van den Akker Erik B. et al. (2020) Metabolic Age Based on the BBMRI-NL 1H-NMR Metabolomics Repository as Biomarker of Age-related Disease. Circulation: Genomic and Precision Medicine, 13, 541–547, doi:10.1161/CIRCGEN.119.002610

---

synthetic_metabolic_dataset

Synthetic example Nightingal metabolomics dataset made for the package MetaboRiSc

---

Description

Synthetic example dataset made with synthpop

Usage

```
data("synthetic_metabolic_dataset")
```

Format

The format is: num [1:500, 1:228] 1.14e-12 2.11e-10 3.18e-11 1.05e-10 4.38e-10 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:5000] "Dummy1" "Dummy2" "Dummy3" "Dummy4" ... ..$: chr [1:228] "xxl_vldl_p" "xxl_vldl_l" "xxl_vldl_pl" "xxl_vldl_c" ...

Examples

```
data(synthetic_metabolic_dataset)
## maybe str(synthetic_metabolic_dataset) ; plot(synthetic_metabolic_dataset) ...
```

---

synthetic_phenotypic_dataset

Synthetic example phenotypic dataset made for the package MetaboRiSc

---

Description

Synthetic example dataset made with synthpop

Usage

```
data("synthetic_phenotypic_dataset")
```

Format

A data frame with 500 observations on the following 20 variables.

Examples

```
data(synthetic_phenotypic_dataset)
## maybe str(synthetic_phenotypic_dataset) ; plot(synthetic_phenotypic_dataset) ...
```

ttest_scores *ttest_scores*

Description

#' Function that creates a boxplot with a continuous variable split using the binary variable

Usage

```
ttest_scores(dat, pred, pheno)
```

Arguments

dat	The data.frame containing the 2 variables
pred	character indicating the y variable
pheno	character indicating the binary variable

Value

plotly boxplot with the continuous variable split using the binary variable

Examples

```
library(MiMIR)
library(plotly)

#load the dataset
metabolic_measures <- synthetic_metabolic_dataset
phenotypes <- synthetic_phenotypic_dataset

#Compute the mortality score
mortScore<-comp.mort_score(metabolic_measures,quiet=TRUE)
dat<-data.frame(predictor=mortScore, pheno=phenotypes$sex)
colnames(dat)<-c("predictor","pheno")
ttest_scores(dat = dat, pred= "mortScore", pheno="sex")
```

ttest_surrogates *ttest_surrogates*

Description

Function that calculates a t-test and a plotly image of the selected surrogates

Usage

```
ttest_surrogates(surrogates, bin_phenotypes)
```

Arguments

surrogates	numeric data.frame containing the surrogate values by Bizzarri et al.
bin_phenotypes	numeric data.frame with the binarized phenotypes output of binarize_all_pheno

Details

Barplot and T-test indicating if the surrogate variables could split accordingly the real value of the binary clinical variables.

Value

plotly image with all the ROCs for all the available clinical variables

Examples

```
require(pROC)
require(plotly)
require(MiMIR)
require(foreach)

#load the dataset
m <- synthetic_metabolic_dataset
p <- synthetic_phenotypic_dataset

#Calculating the binarized surrogates
b_p<-binarize_all_pheno(p)
#Apply a surrogate models and plot the ROC curve
surr<-calculate_surrogate_scores(met=m, pheno=p, MiMIR::PARAM_surrogates, bin_names=colnames(b_p))
ttest_surrogates(surr$surrogates, b_p)
```

Index