Package 'ukbnmr'

November 30, 2025

Type Package

Title Removal of Unwanted Technical Variation from UK Biobank NMR Metabolomics Biomarker Data

Version 3.3.1

BugReports https://github.com/sritchie73/ukbnmr/issues

Description A suite of utilities for working with the UK Biobank <https://www.ukbiobank.ac.uk/> Nuclear Magnetic Resonance spectroscopy (NMR) metabolomics data <https://biobank.ndph.ox.ac.uk/showcase/label.cgi?id=220>. Includes functions for extracting biomarkers from decoded UK Biobank field data, removing unwanted technical variation from biomarker concentrations, computing an extended set of lipid, fatty acid, and cholesterol fractions, and for re-deriving composite biomarkers and ratios after adjusting data for unwanted biological variation. For further details on methods see Ritchie SC et al. Sci Data (2023) <doi:10.1038/s41597-023-01949-y>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports data.table, bit64, lubridate, MASS

Suggests knitr, rmarkdown

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/Needs/development roxygen2

NeedsCompilation no

Author Scott C Ritchie [aut, cre] (0000-0002-8454-9548)

Maintainer Scott C Ritchie <sritchie73@gmail.com>

Repository CRAN

Date/Publication 2025-11-30 16:40:02 UTC

Contents

`compute_extended_ratios`

Compute extended set of biomarker ratios

Description

Computes 76 additional ratios not provided by the Nightingale platform. These include lipid fractions in HDL, LDL, VLDL, and total serum, as well as cholesterol fractions, and omega to polyunsaturated fatty acid ratios. See `nmr_info` for details.

Usage

```
compute_extended_ratios(x)
```

Arguments

`x`	`data.frame` containing NMR metabolomics data from UK Biobank. May either be raw field data output by ukbconv or data with column names corresponding to biomarkers listed in `nmr_info`.

Details

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

All biomarkers in the input data are also returned alongside the ratios computed by this function.

Value

a `data.frame` with the additional computed biomarker ratios.

See Also

nmr_info for list of computed biomarker ratios, compute_extended_ratio_qc_flags() for obtaining an aggregate of the biomarker QC flags from the biomarkers underlying each computed ratio, and extract_biomarkers() for details on how raw data from ukbconv is processed.

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
nmr <- compute_extended_ratios(ukb_data)
```

---

compute_extended_ratio_qc_flags

Aggregate QC Flags when computing the extended set of biomarker ratios

---

Description

For the 76 biomarker ratios computed by compute_extended_ratios(), aggregates the biomarker QC flags from the biomarkers composing each ratio (see nmr_info), which can be useful for determining the reason underlying missing values in the biomarker ratios.

Usage

```
compute_extended_ratio_qc_flags(x)
```

Arguments

x	data.frame containing NMR metabolomics data from UK Biobank. May either be raw field data output by ukbconv or data with column names corresponding to biomarkers listed in nmr_info.

Details

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

Biomarker QC Flags in the input data are also returned alongside those aggregated by this function for the computed biomarker ratios.

Value

a data.frame with QC flags aggregated for all computed biomarker ratios.

See Also

nmr_info for list of computed biomarker ratios and extract_biomarkers() for details on how raw data from ukbconv is processed.

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
biomarker_qc_flags <- compute_extended_ratio_qc_flags(ukb_data)
```

`extract_biomarkers` *Extract NMR metabolomic biomarkers from a data.frame of UK Biobank fields*

Description

Given an input `data.frame` loaded from a dasaset of NMR metabolomics fields extracted by the Table Exporter tool on the UK Biobank Research Analysis Platform, this function extracts the NMR metabolomics biomarkers giving them short variable names as listed in the `nmr_info` information data sheet available in this package.

Usage

```
extract_biomarkers(x)
```

Arguments

x	`data.frame` with column names "eid" followed by extracted fields e.g. "p23474_i0", "p23474_i1", . . . , "p23467_i1".

Details

Data sets extracted on the UK Biobank Research Analysis Platform have one row per UK Biobank participant, whose project specific sample identifier is given in the first column named "eid". Columns following this follow a naming scheme based on the unique identifier of each field, assessment visit, and (optionally if relevant) repeated measurement of "p<field_id>_i<visit_index>_a<repeat_index>". For example, the measurement of 3-Hydroxybutyrate at baseline assessment has the column name "p23474_i0". For the UKB NMR data, measurements are available at baseline assessment and the first repeat assessment (e.g. "p23474_i1"). For the UKB NMR data, the <repeat_index> is reserved for cases where biomarker measurements have more than one QC Flag (see `extract_biomarker_qc_flags()`).

The `data.frame` returned by this function gives each field a unique recognizable name, with measurements from baseline and repeat assessment given in separate rows. The "visit_index" column immediately after the "eid" column indicates whether the biomarker measurement was quantified from the blood samples taken at baseline assessment (visit_index == 0) or first repeat assessment (visit_index == 1). Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index".

This function will also work with data predating the Research Analysis Platform, including data sets extracted by the ukbconv tool and/or the `ukbtools` R package.

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

A `data.table` will be returned instead of a `data.frame` if the the user has loaded the package into their R session.

Value

a data.frame or data.table with column names "eid", and "visit_index", followed by columns for each biomarker e.g. "bOHbutyrate", . . . , "Valine".

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
nmr <- extract_biomarkers(ukb_data)
```

---

extract_biomarker_qc_flags

Extract NMR biomarker QC flags from a data.frame of UK Biobank fields

---

Description

Given an input data.frame loaded from a dasaset of NMR metabolomics QC indicator fields extracted by the Table Exporter tool on the UK Biobank Research Analysis Platform, this function extracts the quality control (QC) flags for the NMR metaolomics biomarkers giving them short variable names as listed in the nmr_info information data sheet available in this package. QC Flags are separated by "; " in each column where there are multiple QC Flags for a single measurement.

Usage

```
extract_biomarker_qc_flags(x)
```

Arguments

x	data.frame with column names "eid" followed by extracted fields e.g. "p23774_i0_a0", . . . "p23774_i1_a0", . . . , "p23767_i1".

Details

#' Data sets extracted on the UK Biobank Research Analysis Platform have one row per UK Biobank participant, whose project specific sample identifier is given in the first column named "eid". Columns following this follow a naming scheme based on the unique identifier of each field, assessment visit, and (optionally if relevant) repeated measurement of "p<field_id>_i<visit_index>_a<repeat_index>". For example, the QC flags for the measurement of 3-Hydroxybutyrate at baseline assessment has the column names "p23774_i0_a0", "p23774_i0_a1", and "p23774_i0_a2"; indicating that the 3-Hydroxybutyrate measurement can have up to three QC flags per sample at baseline assessment. Measurements for blood samples collected at the first repeat assessment have 1 in the visit index, e.g. for 3-Hydroxybutyrate at the first repeat assessment there are three columns "p23774_i1_a0", "p23774_i1_a1", "p23774_i1_a2".

The data.frame returned by this function gives each field a unique recognizable name, with measurements from baseline and repeat assessment given in separate rows. The "visit_index" column immediately after the "eid" column indicates whether the biomarker measurement was quantified

from the blood samples taken at baseline assessment (visit_index == 0) or first repeat assessment (visit_index == 1). Where multiple QC flags were present at the same measurement, these are collated into a single entry with the multiple QC flags separated by a "; ". Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index".

This function will also work with data predating the Research Analysis Platform, including data sets extracted by the ukbconv tool and/or the `ukbtools` R package.

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

A `data.table` will be returned instead of a `data.frame` if the the user has loaded the package into their R session.

Value

a `data.frame` or `data.table` with column names "eid", and "visit_index" followed by columns for each biomarker e.g. "bOHbutyrate", . . . , "Valine".

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
biomarker_qc_flags <- extract_biomarker_qc_flags(ukb_data)
```

`extract_sample_qc_flags`

Extract NMR sample QC flags from a data.frame of UK Biobank fields

---

Description

Given an input `data.frame` loaded from a dasaset of NMR metabolomics processing fields extracted by the Table Exporter tool on the UK Biobank Research Analysis Platform, this function extracts the sample quality control flags for the NMR metabolomics biomarker data giving them short variable names as listed in the `sample_qc_info` information data sheet available in this package.

Usage

```
extract_sample_qc_flags(x)
```

Arguments

`x`	`data.frame` with column names "eid" followed by extracted fields e.g. "p23649_i0", "p23649_i1", . . . , "p23655_i1".

Details

Data sets extracted on the UK Biobank Research Analysis Platform have one row per UK Biobank participant, whose project specific sample identifier is given in the first column named "eid". Columns following this follow a naming scheme based on the unique identifier of each field, assessment visit, and (optionally if relevant) repeated measurement of "p<field_id>_i<visit_index>_a<repeat_index>". For example, the Shipment Plate for each sample collected at baseline assessment has the column name "p23649_i0". For the UKB NMR data, measurements are available at baseline assessment and the first repeat assessment (e.g. "p23649_i1"). For the UKB NMR data, the <repeat_index> is reserved for cases where biomarker measurements have more than one QC Flag (see `extract_biomarker_qc_flags()`).

The `data.frame` returned by this function gives each field a unique recognizable name, with measurements from baseline and repeat assessment given in separate rows. The "visit_index" column immediately after the "eid" column indicates whether the biomarker measurement was quantified from the blood samples taken at baseline assessment (visit_index == 0) or first repeat assessment (visit_index == 1). Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index".

This function will also work with data predating the Research Analysis Platform, including data sets extracted by the ukbconv tool and/or the `ukbtools` R package.

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

A `data.table` will be returned instead of a `data.frame` if the the user has loaded the package into their R session.

Value

a `data.frame` or `data.table` with column names "eid" and "visit_index", followed by columns for each sample QC tag, e.g. "Shipment.Plate", . . . , "Low.Protein".

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
sample_qc_flags <- extract_sample_qc_flags(ukb_data)
```

---

nmr_info *Nightingale biomarker information*

Description

Contains details on the Nightingale biomarkers available in UK Biobank and computed by this package.

Usage

```
nmr_info
```

Format

A data table with 325 rows and 8 columns:

Biomarker Short column name assigned to the biomarker

Description Biomarker description, matching the description field provided by UK Biobank and Nightingale Health

Units Units of measurement for the biomarker ("mmol/L", "g/L", "nm", "degree", "ratio", or "%")

Type Biomarker type ("Non-derived", "Composite", "Ratio", or "Percentage")

Group Biomarker group as provided by Nightingale Health

Sub.Group Biomarker sub-group as provided by Nightingale Health

Nightingale Logical. Indicates biomarker is quantified by the Nightingale Health platform

UKB.Field.ID Field ID in UK Biobank, see https://biobank.ndph.ox.ac.uk/showcase/label.cgi?id=220

QC.Flag.Field.ID Field ID in UK Biobank for the biomarker QC Flags, see https://biobank.ndph.ox.ac.uk/showcase/label.cgi?id=220

Full.Formula For composite biomarkers and ratios, details formula through which the biomarker can be derived from the 107 non-derived biomarkers

Simplified.Formula Simplified form of the full formula most clearly expressing how each composite biomarker and ratio can be rederived from other biomarkers

---

recompute_derived_biomarkers

Recompute composite biomarkers and ratios from the 107 non-derived biomarkers

---

Description

When adjusting biomarkers for unwanted biological covariates, it is desirable to recompute composite biomarkers and ratios to ensure consistency in the adjusted dataset. This function will compute all composite biomarkers and ratios from their parts (see `nmr_info` for biomarker details).

Usage

```
recompute_derived_biomarkers(x)
```

Arguments

`x`	`data.frame` containing NMR metabolomics data from UK Biobank. May either be raw field data output by ukbconv or data with column names corresponding to biomarkers listed in `nmr_info`.

Details

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

All biomarkers in the input data are also returned alongside those computed by this function.

Value

a data.frame with all composite biomarkers and ratios (re)computed from the 107 non-derived biomarkers (see nmr_info for details).

See Also

nmr_info for list of computed biomarker ratios, recompute_derived_biomarker_qc_flags() for obtaining an aggregate of the biomarker QC flags from the biomarkers underlying each computed biomarker, and extract_biomarkers() for details on how the raw field data extracted by the Table Exporter tool is processed.

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
bio_qc <- recompute_derived_biomarkers(ukb_data)
```

---

recompute_derived_biomarker_qc_flags

Aggregate QC Flags when recomputing all composite and derived biomarkers

---

Description

For the 61 composite biomarkers, 81 Nightingale biomarker ratios, and 76 extended biomarker ratios computed by recompute_derived_biomarkers(), aggregates the biomarker QC flags from the underlying biomarkers (see nmr_info).

Usage

```
recompute_derived_biomarker_qc_flags(x)
```

Arguments

x	data.frame containing NMR metabolomics data from UK Biobank. May either be raw field data output by ukbconv or data with column names corresponding to biomarkers listed in nmr_info.

Details

If your UK Biobank project only has access to a subset of biomarkers, then this function will only return the subset of ratios that can be computed from the biomarker data provided.

Biomarker QC Flags in the input data are also returned alongside those aggregated by this function for the computed biomarker ratios.

Value

a data.frame with QC flags aggregated for all computed biomarkers and ratios.

See Also

nmr_info for list of computed biomarker ratios and extract_biomarkers() for details on how raw data from ukbconv is processed.

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
biomarker_qc_flags <- recompute_derived_biomarker_qc_flags(ukb_data)
```

`remove_technical_variation` *Remove technical variation from NMR biomarker data in UK Biobank.*

Description

Remove technical variation from NMR biomarker data in UK Biobank.

Usage

```
remove_technical_variation(
  x,
  remove.outlier.plates = TRUE,
  skip.biomarker.qc.flags = FALSE,
  version = 3L
)
```

Arguments

`x`	`data.frame` containing a tabular phenotype dataset extracted by the Table Exporter tool on the UK Biobank Research Analysis Platform containing the project specific sample id (eid) and all fields (and instances) relating to the NMR metabolomics data (i.e. fields listed in showcase categories 220, 221, and 222.
`remove.outlier.plates`	logical, when set to `FALSE` biomarker concentrations on outlier shipment plates (see details) are not set to missing but simply flagged in the `biomarker_qc_flags` `data.frame` in the returned `list`.
`skip.biomarker.qc.flags`	logical, when set to `TRUE` biomarker QC flags are not processed or returned.
`version`	version of the QC algorithm to apply. Defaults to 3, the latest version of the algorithm (see details).

Details

Three versions of the QC algorithm have been developed. Version 1 was designed based on the first phase of data released to the public covering ~120,000 UK Biobank participants. Version 2 made several improvements to the algorithm based on the subsequent second public release of data covering an additional ~150,000 participants. Version 3, the default, makes some further minor tweaks primarily so that the algorithm is compatible with the full public data release covering all ~500,000 participants.

Details on the impact of these algorithms on technical variation in the latest UK Biobank data are provided in the package vignette and on the github readme.

Version 1 Setting version to 1 applies the algorithm as exactly described in Ritchie S. C. *et al.* _Sci Data_ 2023. In brief, this multi-step procedure applies the following steps in sequence:

1. First biomarker data is filtered to the 107 biomarkers that cannot be derived from any combination of other biomarkers. 2. Absolute concentrations are log transformed, with a small offset applied to biomarkers with concentrations of 0. 3. Each biomarker is adjusted for the time between sample preparation and sample measurement (hours). 4. Each biomarker is adjusted for systematic differences between rows (A-H) on the 96-well shipment plates. 5. Each biomarker is adjusted for remaining systematic differences between columns (1-12) on the 96-well shipment plates. 6. Each biomarker is adjusted for drift over time within each of the six spectrometers. To do so, samples are grouped into 10 bins, within each spectrometer, by the date the majority of samples on their respective 96-well plates were measured. 7. Regression residuals after the sequential adjustments are transformed back to absolute concentrations. 8. Samples belonging to shipment plates that are outliers of non-biological origin are identified and set to missing. 9. The 61 composite biomarkers and 81 biomarker ratios are recomputed from their adjusted parts. 10. An additional 76 biomarker ratios of potential biological significance are computed.

At each step, adjustment for technical covariates is performed using robust linear regression. Plate row, plate column, and sample measurement date bin are treated as factors, using the group with the largest sample size as reference in the regression.

Version 2 Version 2 of the algorithm modifies the above procedure to (1) adjust for well and column within each processing batch separately in steps 4 and 5, and (2) in step 6 instead of splitting samples into 10 bins per spectrometer uses a fixed bin size of approximately 2,000 samples per bin, ensuring samples measured on the same plate and plates measured on the same day are grouped into the same bin.

The first modification was made as applying version 1 of the algorithm to the combined data from the first and second tranche of measurements revealed introduced stratification by well position when examining the corrected concentrations in each data release separately.

The second modification was made to ensure consistent bin sizes across data releases when correcting for drift over time. Otherwise, spectrometers used in multiple data releases would have different bin sizes when adjusting different releases. A bin split is also hard coded on spectrometer 5 between plates 0490000006726 and 0490000006714 which correspond to a large change in concentrations akin to a spectrometer recalibration event most strongly observed for alanine concentrations.

Version 3 Version 3 of the algorithm makes two further minor changes:

1. Imputation of missing sample measurement times has been improved. Previously, any samples missing time of measurement (N=3 in the phase 2 public release) had their time of measurement set to 00:00. In version 3, the time of measurement is set to the median time of measurement for that spectrometer on that day, which is between 12:00-13:00, instead of 00:00. 2. Samples

missing sample preparation dates and times (N=182 in the V20 UK Biobank data release) use their sample centrifugation date and time as a proxy to allow adjustment for time between sample prep and sample measurement. Sample centrifugation always takes place a short time after sample preparation. As sample centrifugation date and time is not made available via UK Biobank, this data and time is hard coded; all samples missing sample preparation date and time had a sample centrifugation date and time of 2022-12-20 06:39:03 in the extended advance access data, so we use this date and time for any samples missing sample preparation date and time. 3. Underlying code for adjusting drift over time has been modified to accommodate the phase 3 public release, which includes one spectrometer with ~2,500 samples. Version 2 of the algorithm would split this into two bins, whereas version 3 keeps this as a single bin to better match the bin sizes of the rest of the spectrometers.

Value

a `list` containing three `data.frames`:

biomarkers A `data.frame` with column names "eid", and "visit_index", containing project-specific sample identifier and UK Biobank visit index (0 for baseline assessment, 1 for first repeat assessment), followed by columns for each biomarker containing their absolute concentrations (or ratios thereof) adjusted for technical variation. See `nmr_info` for information on each biomarker.

biomarker_qc_flags A `data.frame` with the same format as `biomarkers` with entries corresponding to the quality control indicators for each sample. "High plate outlier" and "Low plate outlier" indicate the value was set to missing due to systematic abnormalities in the biomarker's concentration on the sample's shipment plate compared to all other shipment plates (see Details). For composite and derived biomarkers, quality control flags are aggregates of any quality control flags for the underlying biomarkers from which the composite biomarker or ratio is derived.

sample_processing A `data.frame` containing the processing information and quality control indicators for each sample, including those derived for removal of unwanted technical variation by this function. See `sample_qc_info` for details.

log_offset A `data.frame` containing diagnostic information on the offset applied so that biomarkers with concentrations of 0 could be log transformed, and any right shift applied to prevent negative concentrations after rescaling adjusted residuals back to absolute concentrations. Should contain only biomarkers with minimum concentrations of 0 (in the "Minimum" column). "Minimum.Non.Zero" gives the smallest non-zero concentration for the biomarker. "Log.Offset" the small offset added to all samples prior to log transformation: half the minimum non-zero concentration. "Right.Shift" gives the small offset added to prevent negative concentrations that arise after rescaling residuals to log concentrations: this should be at least one order of magnitude smaller than the smallest non-zero value (i.e. the offset added should amount to noise in numeric precision for all samples). See publication for more details.

outlier_plate_detection A `data.frame` containing diagnostic information and details of outlier plate detection. For each of the 107 non-derived biomarkers, the median concentration on each of the 1,352 plates was calculated, then plates were flagged as outliers if their median value deviated more than expected from the mean of plate medians. "Mean.Plate.Medians" gives the mean of the plate medians for each biomarker. "Lower.Limit" and "Upper.Limit" give the values below and above which plates are flagged as outliers based on their plate median. See publication for more details.

algorithm_version Version of the algorithm run, either 1, 2, or 3 (default).

Examples

```
ukb_data <- ukbnmr::test_data # Toy example dataset for testing package
processed <- remove_technical_variation(ukb_data)
```

`sample_qc_info` *Nightingale biomarker sample processing information*

Description

Contains details on the sample processing and quality control information for the NMR biomarker data in UK Biobank.

Usage

```
sample_qc_info
```

Format

A data table with 18 rows and 3 columns:

Name Column name assigned to the sample information field

Description Brief description of the field contents. Further details on the Nightingale sample QC columns can be found in the accompanying resource on the UK Biobank showcase.

UKB.Field.ID Field ID in UK Biobank, see `https://biobank.ndph.ox.ac.uk/showcase/label.cgi?id=222`. Rows missing `UKB.Field.ID` entries correspond to additional sample processing information derived from these fields and returned by `remove_technical_variation`.

`test_data` *Data for testing package functions*

Description

Dataset mimicking structure of decoded UK Biobank dataset of NMR metabolomics biomarker concentrations and associated processing variables for testing package functions.

Usage

```
test_data
```

Format

A data table with 50 rows and 735 columns with column names "eid" followed by extracted UK Biobank field data of the format "p23649_i0", "p23649_i1", . . . , "p23655_i1".

Source

Data in each column has been randomly drawn from the distribution present in the UK Biobank dataset. Importantly, random sampling was performed for each column separately, thus no rows represent real observations or participants in UK Biobank.

---

ukbnmr *Tools for processing the UK Biobank NMR metabolomics biomarker data*

Description

@description This package provides utilities for working with the UK Biobank MR metabolomics data. Details are provided below, and in the package vignette (type `vignette("ukbnmr")` to view).

Details

There are three groups of functions in this package: (1) data extraction, (2) removal of technical variation, and (3) recomputing derived biomarkers and biomarker ratios.

All functions can be applied directly to raw data extracted from UK Biobank.

This package also provides a `data.frame` of biomarker information, loaded as `nmr_info`, and `data.frame` of sample processing information, loaded as `sample_qc_info`.

Data Extraction Functions

The `extract_biomarkers()` function will take a phenotype dataset extracted on the UK Biobank Research Analysis Platform by the Table Exporter tool, extract the NMR biomarker fields and give them short comprehensible column names as described in `nmr_info`. Measurements are also split into multiple rows where a participant has measurements at both baseline and first repeat assessment.

The `extract_biomarker_qc_flags()` function will take a phenotype dataset extracted on the UK Biobank Research Analysis Platform by the Table Exporter tool, extract the Nightingale quality control flags for each biomarker measurement, returning a single column per biomarker (corresponding to respective columns output by `extract_biomarkers()`).

The `extract_sample_qc_flags()` function will take a phenotype dataset extracted on the UK Biobank Research Analysis Platform by the Table Exporter tool and extract the sample quality control tags for the Nightingale NMR metabolomics data.

These functions will also work with older datasets predating the UK Biobank Research Analysis Platform, e.g. those extracted by ukbconv, and/or by the ukbtools R package.

Removal of technical variation

The `remove_technical_variation()` function will take a phenotype dataset extracted on the UK Biobank Research Analysis Platform by the Table Exporter tool, extract all the biomarkers and QC flags, remove the effects of technical variation on biomarker concentrations, and return a list containing the adjusted NMR biomarker data, biomarker QC flags, and sample quality control and processing information.

This applies a multistep process as described in Ritchie *et al.* 2023:

1. First biomarker data is filtered to the 107 biomarkers that cannot be derived from any combination of other biomarkers.
2. Absolute concentrations are log transformed, with a small offset applied to biomarkers with concentrations of 0.
3. Each biomarker is adjusted for the time between sample preparation and sample measurement (hours) on a log scale.
4. Each biomarker is adjusted for systematic differences between rows (A-H) on the 96-well shipment plates.
5. Each biomarker is adjusted for remaining systematic differences between columns (1-12) on the 96-well shipment plates.
6. Each biomarker is adjusted for drift over time within each of the six spectrometers. To do so, samples are grouped into 10 bins, within each spectrometer, by the date the majority of samples on their respective 96-well plates were measured.
7. Regression residuals after the sequential adjustments are transformed back to absolute concentrations.
8. Samples belonging to shipment plates that are outliers of non-biological origin are identified and set to missing.
9. The 61 composite biomarkers and 81 biomarker ratios are recomputed from their adjusted parts.
10. An additional 76 biomarker ratios of potential biological significance are computed.

Further details can be found in Ritchie S. C. *et al.* Quality control and removal of technical variation of NMR metabolic biomarker data in ~120,000 UK Biobank participants, *Sci Data* **10**, 64 (2023). doi: 10.1038/s41597-023-01949-y

Methods for computing biomarker ratios

The `compute_extended_ratios()` function will compute an extended set of biomarker ratios expanding on the biomarkers available directly from the Nightingale platform. A companion function, `compute_extended_ratio_qc_flags()`, will aggregate the QC flags for the biomarkers underlying each ratio.

The `recompute_derived_biomarkers()` function will recompute all composite biomarkers and ratios from 107 non-derived biomarkers, which is useful for ensuring data consistency when adjusting for unwanted biological variation. This includes the extended biomarker rations computed by the `compute_extended_ratios()` function. A companion function, `recompute_derived_biomarker_qc_flags()` will aggregate the QC flags for the biomarkers underlying each composite biomarker and ratio.

Author(s)

Maintainer: Scott C Ritchie <sritchie73@gmail.com> (0000-0002-8454-9548)

See Also

Useful links:

- Report bugs at https://github.com/sritchie73/ukbnmr/issues

Index