

Package ‘ukbnmr’

January 9, 2023

Type Package

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

Version 1.4

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. medRxiv (2021) <[doi:10.1101/2021.09.24.21264079](https://doi.org/10.1101/2021.09.24.21264079)>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports data.table, MASS

Suggests knitr, rmarkdown, roxygen2

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Scott C Ritchie <sritchie73@gmail.com>

Repository CRAN

Date/Publication 2023-01-09 10:00:02 UTC

R topics documented:

compute_extended_ratios	2
compute_extended_ratio_qc_flags	3
compute_nightingale_ratios	4
compute_nightingale_ratio_qc_flags	5
extract_biomarkers	6
extract_biomarker_qc_flags	7
extract_sample_qc_flags	8
nmr_info	9
recompute_derived_biomarkers	9
recompute_derived_biomarker_qc_flags	10
remove_technical_variation	11
sample_qc_info	13
test_data	14
ukbnmr	15

Index	17
--------------	-----------

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)
```

```
compute_nightingale_ratios
```

Compute 81 biomarker ratios on the Nightingale platform

Description

The Nightingale Health NMR metabolomics biomarker platform quantifies **249 biomarkers**, including 81 biomarker ratios. Prior to the August 2021 update, UK Biobank only provided the 168 biomarkers that are not ratios for **download**. This function will compute the 81 missing ratios.

Usage

```
compute_nightingale_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_nightingale_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_nightingale_ratios(ukb_data)
```

`compute_nightingale_ratio_qc_flags`*Aggregate QC Flags when computing Nightingale's 81 biomarker ratios*

Description

For the 81 biomarker ratios computed by `compute_nightingale_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_nightingale_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_nightingale_ratio_qc_flags(ukb_data)
```

extract_biomarkers	<i>Extract NMR metabolomic biomarkers from a data.frame of UK Biobank fields</i>
--------------------	--

Description

Given an input `data.frame` loaded from a dataset extracted by `ukbconv` extracts the **UK Biobank fields** corresponding to 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. "23474-0.0", "23474-1.0", ..., "23467-1.0".

Details

Data sets extracted by `ukbconv` have one row per UKB biobank participant whose project specific sample identifier is given in the first column named "eid". Columns following this have the format "<field_id>-<visit_index>-<repeat_index>", where here <field_id> corresponds to a biomarker of interest, e.g. 23474 for 3-Hydroxybutyrate, <visit_index> corresponds to the assessment time point, e.g. 0 for baseline assessment, 1 for first repeat visit. For the UKB NMR data, the <repeat_index> column is reserved for cases where biomarker measurements have more than one QC Flag (see `extract_biomarker_qc_flags()`).

In the returned `data.frame` there is single column for each biomarker, with an additional column for the visit index. Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index". There are currently no repeat measure data for the NMR biomarker data in UKB, so no `repeat_index` column is returned.

This function will also work with data extracted by 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 dataset extracted by `ukbconv` extracts the **UK Biobank fields** corresponding to 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. "23774-0.0", "23774-1.0", ..., "23767-1.0".

Details

`ukbconv` have one row per UKB biobank participant whose project specific sample identifier is given in the first column named "eid". Columns following this have the format "<field_id>-<visit_index>-<repeat_index>", where here <field_id> corresponds to a biomarker of interest, e.g. 23774 for the QC Flags for 3-Hydroxybutyrate, and <visit_index> corresponds to the assessment time point, e.g. 0 for baseline assessment, 1 for first repeat visit.

The <repeat_index> field is currently used by UK Biobank to index cases where there are multiple QC Flags for a biomarker for a single participant and visit_index. QC Flags are separated by ";" in each column where there are multiple QC Flags for a single measurement.

In the returned `data.frame` there is single column for each biomarker, with an additional column for the visit index. Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index".

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.

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 dataset extracted by `ukbconv` extracts the **UK Biobank fields** corresponding to the **sample quality control flags for the NMR metabolomics biomarker data** giving them short variable names.

Usage

```
extract_sample_qc_flags(x)
```

Arguments

`x` `data.frame` with column names "eid" followed by extracted fields e.g. "23649-0.0", "23649-1.0", ..., "23655-1.0".

Details

Data sets extracted by `ukbconv` have one row per UKB biobank participant whose project specific sample identifier is given in the first column named "eid". Columns following this have the format "<field_id>-<visit_index>-<repeat_index>", where here <field_id> corresponds to a sample QC flag, and <visit_index> corresponds to the assessment time point, e.g. 0 for baseline assessment, 1 for first repeat visit. For the UKB NMR data, the <repeat_index> column is reserved for cases where biomarker measurements have more than one QC Flag (see `extract_biomarker_qc_flags()`).

In the returned `data.frame` there is single column for each QC Flag, with an additional column for the visit index. Rows are uniquely identifiable by the combination of entries in columns "eid" and "visit_index". There are currently no repeat measure data for the NMR biomarker data in UKB, so no `repeat_index` column is returned.

This function will also work with data extracted by the `ukbtools` R package.

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 raw data from `ukbconv` 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  
)
```

Arguments

- `x` data.frame containing a dataset extracted by `ukbconv` with UK Biobank fields containing the **Nightingale Health NMR metabolomics biomarker** data.
- `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.

Details

A multi-step procedure is applied to the raw biomarker data to remove the effects of technical variation:

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.

Further details can be found in the preprint: Ritchie S. C. *et al.*, Quality control and removal of technical variation of NMR metabolic biomarker data in ~120,000 UK Biobank participants, **medRxiv** (2021). doi: [10.1101/2021.09.24.21264079v1](https://doi.org/10.1101/2021.09.24.21264079v1)

This function takes 10-15 minutes to run and requires at least 14 GB of RAM.

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 preprint 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 preprint for more details.

Examples

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

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 "23649-0.0", "23649-1.0", ..., "23655-1.0".

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 raw dataset output by `ukbconv`, 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 raw dataset output by `ukbconv`, 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 raw dataset output by `ukbconv`, and extract the **sample quality control tags** for the Nightingale NMR metabolomics data.

Removal of technical variation

The `remove_technical_variation()` function will take a raw dataset output by `ukbconv`, 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.* 2021:

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 the preprint: Ritchie S. C. *et al.*, Quality control and removal of technical variation of NMR metabolic biomarker data in ~120,000 UK Biobank participants, **medRxiv** (2021). doi: [10.1101/2021.09.24.21264079v1](https://doi.org/10.1101/2021.09.24.21264079v1)

Methods for computing biomarker ratios

The `compute_nightingale_ratios()` function will compute the **81 Nightingale Health biomarker ratios** that were missing prior to the August 2021 update of **UK Biobank**. A companion function, `compute_nightingale_ratio_qc_flags()` will aggregate the QC flags for the biomarkers underlying each ratio.

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. A companion function, `recompute_derived_biomarker_qc_flags()` will aggregate the QC flags for the biomarkers underlying each composite biomarker and ratio.

Index

* datasets

nmr_info, [9](#)
sample_qc_info, [13](#)
test_data, [14](#)

* package

ukbnmr, [15](#)

compute_extended_ratio_qc_flags, [3](#), [3](#),
[16](#)

compute_extended_ratios, [2](#), [3](#), [16](#)

compute_nightingale_ratio_qc_flags, [4](#),
[5](#), [16](#)

compute_nightingale_ratios, [4](#), [5](#), [16](#)

extract_biomarker_qc_flags, [6](#), [7](#), [8](#), [15](#)

extract_biomarkers, [3-5](#), [6](#), [10](#), [11](#), [15](#)

extract_sample_qc_flags, [8](#), [15](#)

nmr_info, [2-7](#), [9](#), [9](#), [10](#), [11](#), [13](#), [15](#)

recompute_derived_biomarker_qc_flags,
[10](#), [10](#), [16](#)

recompute_derived_biomarkers, [9](#), [10](#), [16](#)

remove_technical_variation, [11](#), [14](#), [15](#)

robust linear regression, [12](#)

sample_qc_info, [13](#), [13](#), [15](#)

test_data, [14](#)

ukbnmr, [15](#)