

# Package ‘pgca’

November 28, 2024

**Type** Package

**Title** PGCA: An Algorithm to Link Protein Groups Created from MS/MS Data

**Version** 1.31.0

**biocViews** WorkflowStep, AssayDomain, Proteomics, MassSpectrometry, ImmunoOncology

**Author** Gabriela Cohen-Freue <gcohen@stat.ubc.ca>

**Maintainer** Gabriela Cohen-Freue <gcohen@stat.ubc.ca>

**Description** Protein Group Code Algorithm (PGCA) is a computationally inexpensive algorithm to merge protein summaries from multiple experimental quantitative proteomics data. The algorithm connects two or more groups with overlapping accession numbers. In some cases, pairwise groups are mutually exclusive but they may still be connected by another group (or set of groups) with overlapping accession numbers. Thus, groups created by PGCA from multiple experimental runs (i.e., global groups) are called “connected” groups. These identified global protein groups enable the analysis of quantitative data available for protein groups instead of unique protein identifiers.

**License** GPL (>= 2)

**Imports** utils, stats

**LazyData** TRUE

**VignetteBuilder** knitr

**Suggests** knitr, testthat, rmarkdown

**RoxygenNote** 6.0.1

**git\_url** <https://git.bioconductor.org/packages/pgca>

**git\_branch** devel

**git\_last\_commit** ec5b4ff

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-11-27

## Contents

applyDict . . . . .	2
BET1947_v339 . . . . .	3
pgcaDict . . . . .	4
saveDict . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

applyDict	<i>Apply a PGCA dictionary to data files</i>
-----------	--

---

### Description

Apply the dictionary to the data files and write the translated files to disk.

### Usage

```
applyDict(..., dict, out.dir = NULL, out.suffix = "", out.prefix = "",
           col.mapping, out.pg.col = "PGC")
```

### Arguments

...	input (see details).
dict	the PGCA dictionary to use.
out.dir	the directory to save the translated files in (see details). If NULL, the translated data frames will be returned directly.
out.suffix, out.prefix	suffix and prefix that will be added to the translated files.
col.mapping	the column mapping for the input files. Defaults to the same as used to build the dictionary.
out.pg.col	the name of the column to store the protein group.

### Details

The dictionary is applied to the data specified argument. If no input is provided, the dictionary is applied to the files used to create the dictionary. The inputs can be directory names, file names, or data.frames.

If the output directory `out.dir` is specified, the translated files will be saved in this directory. Otherwise the files will be written to the same directory as the input files. The function will not overwrite existing files and will fail if the files already exist. Parameters `out.suffix` and `out.prefix` can be used to ensure unique new file names. In case the input is a list of `data.frames` and no output directory is specified, the `data.frames` will be translated and returned as a list. The function will also return the translated `data.frames` as list if the `out.dir=NULL`.

### Value

Either a list of `data.frames` or nothing (see details).

**See Also**

[pgcaDict](#) to create the dictionary

**Examples**

```
# Build PGCA dictionary from all files in a directory
dict <- pgcaDict(
  system.file("extdata", package="pgca"),
  col.mapping=c(gene.symbol="Gene_Symbol")
)

# Translate all files in the directory and return as a list of data.frames
trans <- applyDict(system.file("extdata", package="pgca"), dict=dict,
  out.dir=NULL)

# Translate only some files in the directory and return as a list of
# data.frames
trans <- applyDict(
  system.file("extdata", "BET1947_v339.txt", package="pgca"),
  system.file("extdata", "BET2047_v339.txt", package="pgca"),
  dict=dict
)
str(trans)

# Translate all files in the directory and save to another directory
out.dir <- tempdir()
applyDict(system.file("extdata", package="pgca"), dict=dict,
  out.dir=out.dir)
```

---

 BET1947\_v339

*Data Files From BiT Heart Cohort*


---

**Description**

These are four iTRAQ runs used to process samples from heart transplant patients. The datasets belong to the Biomarkers in Transplantation (BiT) initiative and PROOF Centre of Excellence. In all four runs, the raw MS/MS data was processed using ProteinPilot(tm) software v3.0 with the integrated Paragon(tm) Search and ProGroup(tm) algorithms and searching against the International Protein Index (IPI HUMAN v3.39) database.

**Format**

Each `data.frame` has columns

**N** the local protein group identifier

**Accession** the accession number of the protein

**Gene\_Symbol** the gene symbol

**Protein\_Name** the protein name

**Source**

Takhar M, Sasaki M, Hollander Z, Kepplinger D, Smith D, McManus B, McMaster W, Ng R and Cohen Freue G (Under revision). "PGCA: An Algorithm to Link Protein Groups Created from MS/MS Data."

---

pgcaDict

*Link Protein Groups Created from MS/MS Data*

---

**Description**

Build a dictionary for protein groups from MS/MS data. Details of the algorithm can be found in Takhar et al. (Under revision). "PGCA: An Algorithm to Link Protein Groups Created from MS/MS Data."

**Usage**

```
pgcaDict(..., col.mapping, master.gene.identifier)
```

**Arguments**

... arbitrary number of directory names, file names, or data.frames used as input.

col.mapping column mapping (see Details).

master.gene.identifier if given, genes with this value in the column group.identifier are considered master genes.

**Details**

If the group.identifier column is logical (i.e., TRUE or FALSE) or master.gene.identifier is given, the TRUE accessions are assumed to be a "master gene" and the data set is assumed to be in the correct order. This means all FALSE values following the master gene are assumed to belong to the same group.

The col.mapping maps the column names in the data files to a specific function. It needs to be a named character vector, whereas the name of each item is the "function" of the given column name. The algorithm knows about the following columns:

"group.identifier" Column containing the group identifier.

"accession.nr" Column containing the accession nr.

"protein.name" Column containing the protein name.

"gene.symbol" Column containing the gene symbol (if any, can be missing)

The default column mapping is `c(group.identifier="N", accession.nr="Accession", protein.name="Protein_Na`

The supplied column mapping can ignore those columns that are already correct in the default map.

For instance, if the accession nr. is stored in column *AccessionNr* instead of *Accession*, but the remaining columns are the same as in the default mapping, specifying `col.mapping=c(accession.nr="AccessionNr")` is sufficient.

**Value**

An object of type pgcaDict.

**References**

Takhar M, Sasaki M, Hollander Z, McManus B, McMaster W, Ng R and Cohen Freue G (Under revision). "PGCA: An Algorithm to Link Protein Groups Created from MS/MS Data." PLOS ONE.

**See Also**

[applyDict](#) to apply the dictionary to the data files and [saveDict](#) to save the dictionary itself.

**Examples**

```
# Build the dictionary from all files in a directory
# and specifying the column "Gene_Symbol" holds the `gene.symbol`.
dict.dir <- pgcaDict(
  system.file("extdata", package="pgca"),
  col.mapping=c(gene.symbol="Gene_Symbol")
)

# Build the dictionary from a list of files
dict.files <- pgcaDict(
  system.file("extdata", "BET1947_v339.txt", package="pgca"),
  system.file("extdata", "BET2007_v339.txt", package="pgca"),
  system.file("extdata", "BET2047_v339.txt", package="pgca"),
  col.mapping=c(gene.symbol="Gene_Symbol")
)

# Build the dictionary from already read-in data.frames
dict.data <- pgcaDict(BET1947_v339, BET2047_v339,
  col.mapping=c(gene.symbol="Gene_Symbol"))
```

---

saveDict

*Write a PGCA dictionary to a text file*

---

**Description**

Write the dictionary to a text file using the [write.table](#) function. By default, a tab-separated file is written, but this can be changed by changing the arguments to [write.table](#).

**Usage**

```
saveDict(dict, file = stop("`file` must be specified"), ...)
```

**Arguments**

`dict` a PGCA dictionary.  
`file` either a character string naming a file or a [connection](#) open for writing. "" indicates output to the console.  
... further arguments passed to [write.table](#).

**Value**

This function returns NULL invisibly.

**See Also**

[pgcaDict](#) to create the dictionary, and [applyDict](#) to apply the dictionary for translating data files.

**Examples**

```
# Build accession dictionary from all files in a directory
dict <- pgcaDict(
  system.file("extdata", package="pgca"),
  col.mapping=c(gene.symbol="Gene_Symbol")
)

# Save dictionary to a temporary file
dictOutFile <- tempfile()
saveDict(dict, file=dictOutFile)

# Change the separator string to a tab
dictOutFile <- tempfile()
saveDict(dict, file=dictOutFile, sep="\t")
```

# Index

`applyDict`, [2](#), [5](#), [6](#)

`BET1947_v339`, [3](#)

`BET2007_v339 (BET1947_v339)`, [3](#)

`BET2047_v339 (BET1947_v339)`, [3](#)

`BET2067_v339 (BET1947_v339)`, [3](#)

`connection`, [6](#)

`pgcaDict`, [3](#), [4](#), [6](#)

`saveDict`, [5](#), [5](#)

`write.table`, [5](#), [6](#)