

Package ‘BCGcalc’

August 8, 2026

Type Package

Title Biological Condition Gradient, Calculator

Version 2.3.1

Maintainer Erik W. Leppo <Erik.Leppo@tetrattech.com>

Description Functions to calculate Biological Condition Gradient (BCG) using input files with one row per sample with metric values and site classes as columns. A second file with the BCG Rules (example included) to define the memberships is also needed. The three main functions convert metric scores to metric memberships following fuzzy set BCG Rules (BCG.Level.Assignment), combine metric memberships to level memberships according to BCG Rules (BCG.Level.Membership), and then assign a BCG primary and secondary level based on level memberships (BCG.Level.Assignment). Originally developed as a package for use with BCG for Puget Lowland/Willamette Valley but has been further enhanced for use with multiple communities (benthic macroinvertebrates, fish, periphyton, or coral) and different rule sets. Oregon and Washington reference -
`` Stamp, J. and J. Gerritsen. 2018. Calibration of the Biological Condition Gradient (BCG) for Macroinvertebrate Assemblages in Puget Lowland/Willamette Valley Freshwater Wadeable Streams. Prepared by Tetra Tech for the US EPA Office of Water, Office of Science and Technology and US EPA Region 10." BCG process documentation -
`` USEPA. 2016. A Practitioner’s Guide to the Biological Condition Gradient - A Framework to Describe Incremental Change in Aquatic Ecosystems. EPA 842-R-16-001. Office of Science and Technology, Washington, DC 20460."

License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports dplyr, reshape2

Suggests BioMonTools, readxl, knitr, rmarkdown, ggplot2, shiny, DT, httr, testthat (>= 3.0.0), plyr, tidyr, tibble, rioja, nhdplusTools, ComplexUpset

Encoding UTF-8

LazyData true

URL <https://github.com/leppott/BCGcalc>

BugReports <https://github.com/leppott/BCGcalc/issues>
VignetteBuilder knitr
Config/testthat/edition 3
Config/roxygen2/version 8.0.0
NeedsCompilation no
Author Erik W. Leppo [aut, cre],
Jen Stamp [ctb],
Matt Hedin [ctb],
Ben Block [ctb]
Repository CRAN
Date/Publication 2026-08-08 10:30:02 UTC

Contents

BCGcalc-package	2
BCG.Level.Assignment	3
BCG.Level.Membership	6
BCG.Metric.Membership	9
metrics_values	11
TaxaMaster_Ben_BCG_PugLowWilVal	24
Index	25

BCGcalc-package	<i>BCGcalc: Metric calculation and other tasks related to BCG</i>
-----------------	---

Description

Enables users to employ various tools and calculations related to BCG.

BCGcal functions

#' qc.checks #' ...

Author(s)

Maintainer: Erik W. Leppo <Erik.Leppo@tetrattech.com>
Authors:
• Erik W. Leppo <Erik.Leppo@tetrattech.com>

Other contributors:

- Jen Stamp <Jen.Stamp@tetrattech.com> [contributor]
- Matt Hedin <Matt.Hedin@tetrattech.com> [contributor]
- Ben Block <Ben.Block@tetrattech.com> [contributor]

See Also

Useful links:

- <https://github.com/leppott/BCGcalc>
- Report bugs at <https://github.com/leppott/BCGcalc/issues>

BCG.Level.Assignment *BCG Level Assignment*

Description

Biological Condition Gradient level assignment (1st and 2nd) given Level memberships.

Usage

```
BCG.Level.Assignment(
  df.level.membership,
  col_SampleID = "SAMPLEID",
  col_Index_Name = "INDEX_NAME",
  col_Index_Class = "INDEX_CLASS",
  col_L1 = "L1",
  col_L2 = "L2",
  col_L3 = "L3",
  col_L4 = "L4",
  col_L5 = "L5",
  col_L6 = "L6"
)
```

Arguments

<code>df.level.membership</code>	Wide data frame with level memberships (0-1).
<code>col_SampleID</code>	Column name for sample id. Default = "SAMPLEID"
<code>col_Index_Name</code>	Column name for index name. Default = "INDEX_NAME"
<code>col_Index_Class</code>	Column name for index class. Default = "INDEX_CLASS"
<code>col_L1</code>	Column name for memberships, Level 1. Default = "L1"
<code>col_L2</code>	Column name for memberships, Level 2. Default = "L2"
<code>col_L3</code>	Column name for memberships, Level 3. Default = "L3"
<code>col_L4</code>	Column name for memberships, Level 4. Default = "L4"
<code>col_L5</code>	Column name for memberships, Level 5. Default = "L5"
<code>col_L6</code>	Column name for memberships, Level 6. Default = "L6"

Details

Input is L1 to L6 with membership values of 0 to 1. Result is 1st Level (Primary_BCG_Level) and 2nd Level (Secondary_BCG_Level). Also give close (Membership_Close) and a continuous proportional Level assignment ("Continuous_BCG_Level").

Value

Returns a data frame of results in the wide format.

Examples

```
# Example 1

# construct a dummy dataset
L1 <- c(rep(0, 12))
L2 <- c(0.4, 0, 0.4, rep(0,7), 0, 0)
L3 <- c(0.6, 0, 0.6, 0, 0.42, 0, 1, 1, 0.22, 0.33, 0.5, 0)
L4 <- c(0, 0.9, 0, 0, 0.58, 0.05, 0, 0, 0.78, 0.67, 0.5, 0)
L5 <- c(0, 0.1, 0, 1, 0, 0.95, rep(0,4), 0, 1)
L6 <- c(rep(0, length(L1)))
SAMPLEID <- LETTERS[1:length(L1)]
df_lev_memb <- data.frame(SAMPLEID = SAMPLEID
                          , INDEX_NAME = "TEST_NAME"
                          , INDEX_CLASS = "TEST_CLASS"
                          , L1 = L1
                          , L2 = L2
                          , L3 = L3
                          , L4 = L4
                          , L5 = L5
                          , L6 = L6)

# Run Function
df_Levels <- BCG.Level.Assignment(df_lev_memb)

# Show Results
#View(df_Levels)

# Save Results
write.table(df_Levels
            , file.path(tempdir(), "Levels.tsv")
            , row.names = FALSE
            , col.names = TRUE
            , sep = "\t")

#~~~~~

# Example 2

# library(readxl)
# library(reshape2)
# library(BioMonTools)
```

```

# Calculate Metrics
df_samps_bugs <- readxl::read_excel(system.file(
                                "extdata/Data_BCG_PugLowWilVal.xlsx"
                                , package="BCGcalc")
                                , guess_max = 10^6)

# Run Function
myDF <- df_samps_bugs[1:250, ]
myCols <- c("Area_mi2",
            "SurfaceArea",
            "Density_m2",
            "Density_ft2")

#' # populate missing columns prior to metric calculation
col_missing <- c("INFRAORDER",
                "HABITAT",
                "ELEVATION_ATTR",
                "GRADIENT_ATTR",
                "WSAREA_ATTR",
                "HABSTRUCT",
                "UFC",
                "SAMP_AREA_M2")
myDF[, col_missing] <- NA
df_met_val_bugs <- BioMonTools::metric.values(myDF
                                              , "bugs"
                                              , fun.cols2keep = myCols)

# Import Rules
df_rules <- readxl::read_excel(system.file("extdata/Rules.xlsx"
                                           , package="BCGcalc")
                               , sheet="Rules")

# Calculate Metric Memberships
df_met_memb <- BCG.Metric.Membership(df_met_val_bugs, df_rules)

# Calculate Level Memberships
df_lev_memb <- BCG.Level.Membership(df_met_memb, df_rules)

# Run Function
df_Levels <- BCG.Level.Assignment(df_lev_memb)

# QC Checks (flags)
#
# Import Checks
df_checks <- readxl::read_excel(system.file("extdata/MetricFlags.xlsx"
                                           , package="BCGcalc")
                               , sheet="Flags")

# Run Function
df_flags <- BioMonTools::qc.checks(df_met_val_bugs, df_checks)
# Change terminology; PASS/FAIL to NA/flag
df_flags[, "FLAG"][df_flags[, "FLAG"] == "FAIL"] <- "flag"
df_flags[, "FLAG"][df_flags[, "FLAG"] == "PASS"] <- NA

```

```

# long to wide format
df_flags_wide <- reshape2::dcast(df_flags
                                , SAMPLEID ~ CHECKNAME
                                , value.var="FLAG")

# Calc number of "flag"s by row.
df_flags_wide$NumFlags <- rowSums(df_flags_wide == "flag", na.rm = TRUE)
# Rearrange columns
NumCols <- ncol(df_flags_wide)
df_flags_wide <- df_flags_wide[, c(1, NumCols, 2:(NumCols - 1))]

# Merge Levels and Flags
df_lev_flags <- merge(df_Levels
                      , df_flags_wide
                      , by.x = "SampleID"
                      , by.y = "SAMPLEID"
                      , all.x = TRUE)

# Summarize Results
table(df_flags[, "CHECKNAME"], df_flags[, "FLAG"], useNA = "ifany")
table(df_lev_flags$BCG_Status)

# Show Results
# View(df_lev_flags)

# Save Results
write.csv(df_lev_flags, file.path(tempdir(), "Level_Flags.csv"))

# # Summary Report
# strFile.RMD <- system.file(paste0("rmd/Results_Summary.Rmd")
#                             , package = "BCGcalc")
# strFile.RMD.format <- "html_document"
# strFile.out <- "_bcgcalc_RESULTS.html"
# dir.export <- tempdir()
# rmarkdown::render(strFile.RMD
#                   , output_format = strFile.RMD.format
#                   , output_file = strFile.out
#                   , output_dir = dir.export
#                   , quiet = TRUE)

```

BCG.Level.Membership *BCG Level Membership*

Description

Biological Condition Gradient Level assignment given metric memberships.

Usage

```
BCG.Level.Membership(
```

```

df.metric.membership,
df.rules,
col_SAMPLEID = "SAMPLEID",
col_INDEX_NAME = "INDEX_NAME",
col_INDEX_CLASS = "INDEX_CLASS",
col_LEVEL = "LEVEL",
col_METRIC_NAME = "METRIC_NAME",
col_RULE_TYPE = "RULE_TYPE",
col_EXC_RULE = "EXC_RULE",
col_MEMBERSHIP = "MEMBERSHIP",
...
)

```

Arguments

<code>df.metric.membership</code>	Data frame of metric memberships (long format, the same as the output of <code>BCG.Metric.Membership</code>).
<code>df.rules</code>	Data frame of BCG model rules.
<code>col_SAMPLEID</code>	column name for sample id. Default = <code>SAMPLEID</code>
<code>col_INDEX_NAME</code>	column name for index name. Default = <code>INDEX_NAME</code>
<code>col_INDEX_CLASS</code>	column name for site type. Default = <code>INDEX_CLASS</code>
<code>col_LEVEL</code>	column name for level. Default = <code>LEVEL</code>
<code>col_METRIC_NAME</code>	column name for metric name. Default = <code>METRIC_NAME</code>
<code>col_RULE_TYPE</code>	column name for rule type (e.g., <code>Rule0</code> , <code>Rule1</code> , or <code>Rule2</code>). Default = <code>RULE_TYPE</code>
<code>col_EXC_RULE</code>	column name for exception rules. Default = <code>EXC_RULE</code>
<code>col_MEMBERSHIP</code>	column name for metric membership. Default = <code>MEMBERSHIP</code>
<code>...</code>	Arguments passed to <code>BCG.MetricMembership</code> used internally

Details

Input is metric memberships and a rules tables.

Output is a data frame with the membership for each row to each Level (1:6).

Minimum of:

- 1- sum of previous levels
- Rule0 memberships
- max of Rule1 (Alternate1) rules (and min of Rule2 (Alternate2) rules)

That is, perform calculations in this order:

1. Min of Rule2 (Alternate2) metric memberships
2. Max of Rule1 (Alternate1) rules and the above result.

3. Min of: Rule0, the above results, and 1 - the sum of previous levels.

Some exceptions exist for particular models.

Index_Name	INDEX_CLASS
CT_BCG_2015	fish02
CT_BCG_2015	fish03
BCG_NMSandyRivers	bugs

These exceptions are mostly hard coded into the function but gather some information with the parameter `col_EXC_RULE` from the rules table. A future update may fully automate this process.

2021 saw the introduction of Median Exception rule. For the Pacific Northwest some metrics were grouped and the 2nd of 3 values is used and the other 2 values tossed when determining level membership. This equates to using the median of the 3 values. This is handled by including "MEDIAN" in the `Exc_Rule` column in `Rules.xlsx`. Superseded by "SMALL2".

2024 added SMALL2 and SMALL3 Exception rules. For New Mexico BCG some metrics are grouped so use the 2nd or 3rd smallest value instead of the minimum. As above, this is handled by including "SMALL2" or "SMALL3" in the `Exc_Rule` column in `Rules.xlsx`.

Some Great Plains rules use multiple groupings of SMALL2. These are coded as "SMALL2A" and "SMALL2B". If additional groupings are needed the code needs to be tweaked.

Deprecated `col_SITE_TYPE` for `col_INDEX_CLASS` in v2.0.0.9001.

Value

Returns a data frame of results in the wide format.

Examples

```
# library(readxl)
# library(BioMonTools)

# Calculate Metrics
df_samps_bugs <- readxl::read_excel(
  system.file("extdata/Data_BCG_PugLowWilVal.xlsx"
    , package="BCGcalc")
  , guess_max = 10^6)
myDF <- df_samps_bugs[1:500, ]
myCols <- c("Area_mi2",
  "SurfaceArea",
  "Density_m2",
  "Density_ft2")
# populate missing columns prior to metric calculation
col_missing <- c("INFRAORDER",
  "HABITAT",
  "ELEVATION_ATTR",
  "GRADIENT_ATTR",
  "WSAREA_ATTR",
  "HABSTRUCT",
  "UFC",
```

```

      "SAMP_AREA_M2")
myDF[, col_missing] <- NA
df_met_val_bugs <- BioMonTools::metric.values(myDF
      , "bugs"
      , fun.cols2keep = myCols)

# Import Rules
df_rules <- readxl::read_excel(system.file("extdata/Rules.xlsx"
      , package = "BCGcalc")
      , sheet="Rules")

# Calculate Metric Memberships
df_met_memb <- BCG.Metric.Membership(df_met_val_bugs, df_rules)

# Calculate Level Memberships
df_lev_memb <- BCG.Level.Membership(df_met_memb, df_rules)

# Show results
#View(df_lev_memb)

# Save Results
write.table(df_lev_memb
      , file.path(tempdir(), "Level_Membership.tsv")
      , row.names = FALSE
      , col.names = TRUE
      , sep = "\t")

```

BCG.Metric.Membership *BCG Metric Membership*

Description

Biological Condition Gradient fuzzy membership for metrics.

Usage

```

BCG.Metric.Membership(
  df.metrics,
  df.rules,
  input.shape = "wide",
  col_SAMPLEID = "SAMPLEID",
  col_INDEX_NAME = "INDEX_NAME",
  col_INDEX_CLASS = "INDEX_CLASS",
  col_LEVEL = "LEVEL",
  col_METRIC_NAME = "METRIC_NAME",
  col_RULE_TYPE = "RULE_TYPE",
  col_LOWER = "LOWER",
  col_UPPER = "UPPER",

```

```
col_METRIC_VALUE = "METRIC_VALUE",
col_INCREASE = "INCREASE",
...
)
```

Arguments

<code>df.metrics</code>	Wide data frame with metric values to be evaluated.
<code>df.rules</code>	Data frame of metric thresholds to check.
<code>input.shape</code>	Shape of <code>df.metrics</code> ; wide or long. Default is wide.
<code>col_SAMPLEID</code>	Column name for sample id. Default = "SAMPLEID"
<code>col_INDEX_NAME</code>	Column name for index name. Default = "INDEX_NAME"
<code>col_INDEX_CLASS</code>	Column name for index class Default = "INDEX_CLASS"
<code>col_LEVEL</code>	Column name for level. Default = "LEVEL"
<code>col_METRIC_NAME</code>	Column name for metric name. Default = "METRIC_NAME"
<code>col_RULE_TYPE</code>	Column name for rule type (e.g., Rule0). Default = "RULE_TYPE"
<code>col_LOWER</code>	Column name for lower limit. Default = "LOWER"
<code>col_UPPER</code>	Column name for upper limit. Default = "UPPER"
<code>col_METRIC_VALUE</code>	Column name for metric value. Default = "METRIC_VALUE"
<code>col_INCREASE</code>	Column name for if the metric value increases. Default = "INCREASE"
<code>...</code>	Arguments passed to 'BCG.MetricMembership' used internally

Details

Converts metric values into BCG membership values. Uses a rules table to define the metrics, scoring range, and direction for each named index.

Deprecated `col_SITE_TYPE` for `col_INDEX_CLASS` in v2.0.0.9001.

Value

Returns a data frame of results in the long format.

Examples

```
# library(readxl)
# library(BioMonTools)

# Calculate Metrics
df_samps_bugs <- readxl::read_excel(
  system.file("extdata/Data_BCG_PugLowWilVal.xlsx"
    , package = "BCGcalc")
  , guess_max = 10^6)
myDF <- df_samps_bugs[1:500, ]
```

```

myCols <- c("Area_mi2",
            "SurfaceArea",
            "Density_m2",
            "Density_ft2")
# populate missing columns prior to metric calculation
col_missing <- c("INFRAORDER",
                "HABITAT",
                "ELEVATION_ATTR",
                "GRADIENT_ATTR",
                "WSAREA_ATTR",
                "HABSTRUCT",
                "UFC",
                "SAMP_AREA_M2")
myDF[, col_missing] <- NA
df_met_val_bugs <- BioMonTools::metric.values(myDF
                                              , "bugs"
                                              , fun.cols2keep = myCols)

# Import Rules
df_rules <- readxl::read_excel(system.file("extdata/Rules.xlsx"
                                           , package = "BCGcalc")
                              , sheet="Rules")

# Run function
df_met_memb <- BCG.Metric.Membership(df_met_val_bugs, df_rules)

# Show Results
#View(df_met_memb)

# Save Results
write.table(df_met_memb
            , file.path(tempdir(), "Metric_Membership.tsv")
            , row.names = FALSE
            , col.names = TRUE
            , sep = "\t")

```

metrics_values

Metric Values

Description

Example metric values

Usage

```
metrics_values
```

Format

A data frame with 678 observations on the following 448 variables.:

SAMPLEID a character vector
AREA_MI2 a numeric vector
SURFACEAREA a character vector
DENSITY_M2 a logical vector
DENSITY_FT2 a logical vector
INDEX_NAME a character vector
INDEX_CLASS a character vector
ni_total a numeric vector
li_total a numeric vector
ni_Chiro a numeric vector
ni_EPT a numeric vector
ni_Trich a numeric vector
ni_Americo a numeric vector
ni_Gnorimo a numeric vector
ni_brackish a numeric vector
ni_Ramello a numeric vector
nt_total a numeric vector
nt_Amph a numeric vector
nt_Bival a numeric vector
nt_Coleo a numeric vector
nt_COET a numeric vector
nt_CruMol a numeric vector
nt_Deca a numeric vector
nt_Dipt a numeric vector
nt_ECT a numeric vector
nt_Ephem a numeric vector
nt_Ephemerellid a numeric vector
nt_EPT a numeric vector
nt_ET a numeric vector
nt_Gast a numeric vector
nt_Hemipt a numeric vector
nt_Hepta a numeric vector
nt_Insect a numeric vector
nt_Isop a numeric vector
nt_Mega a numeric vector
nt_Mol a numeric vector
nt_Nemour a numeric vector

nt_NonIns a numeric vector
nt_Odon a numeric vector
nt_OET a numeric vector
nt_Oligo a numeric vector
nt_Perlid a numeric vector
nt_Pleco a numeric vector
nt_POET a numeric vector
nt_Ptero a numeric vector
nt_Rhya a numeric vector
nt_Tipulid a numeric vector
nt_Trich a numeric vector
nt_Tromb a numeric vector
nt_Tubif a numeric vector
pi_Ampe a logical vector
pi_AmpeHaust a logical vector
pi_Amph a numeric vector
pi_AmphIsop a numeric vector
pi_Baet a numeric vector
pi_Bival a numeric vector
pi_Caen a numeric vector
pi_Coleo a numeric vector
pi_COET a numeric vector
pi_Corb a numeric vector
pi_CraCaeGam a numeric vector
pi_Cru a numeric vector
pi_CruMol a numeric vector
pi_Deca a numeric vector
pi_Dipt a numeric vector
pi_DiptNonIns a numeric vector
pi_ECT a numeric vector
pi_Ephem a numeric vector
pi_EphemNoCae a numeric vector
pi_EphemNoCaeBae a numeric vector
pi_EPT a numeric vector
pi_EPTNoBaeHydro a numeric vector
pi_EPTNoCheu a numeric vector
pi_EPTNoHydro a numeric vector

pi_ET a numeric vector
pi_Gast a numeric vector
pi_Hemipt a numeric vector
pi_Hydro a numeric vector
pi_Hydro2EPT a numeric vector
pi_Hydro2Trich a numeric vector
pi_Insect a numeric vector
pi_Isop a numeric vector
pi_IsopGastHiru a numeric vector
pi_Juga a numeric vector
pi_JugaFlumi a numeric vector
pi_Mega a numeric vector
pi_Mol a numeric vector
pi_Nemata a numeric vector
pi_NonIns a numeric vector
pi_Odon a numeric vector
pi_OET a numeric vector
pi_Oligo a numeric vector
pi_Pleco a numeric vector
pi_POET a numeric vector
pi_Sphaer a numeric vector
pi_SphaerCorb a numeric vector
pi_Trich a numeric vector
pi_TrichNoHydro a numeric vector
pi_Tromb a numeric vector
pi_Tubif a numeric vector
pt_Amph a numeric vector
pt_Bival a numeric vector
pt_Coleo a numeric vector
pt_COET a numeric vector
pt_Deca a numeric vector
pt_Dipt a numeric vector
pt_ECT a numeric vector
pt_Ephem a numeric vector
pt_EPT a numeric vector
pt_ET a numeric vector
pt_Gast a numeric vector

pt_Hemipt a numeric vector
pt_Insect a numeric vector
pt_Isop a numeric vector
pt_Mega a numeric vector
pt_NonIns a numeric vector
pt_Odon a numeric vector
pt_OET a numeric vector
pt_Oligo a numeric vector
pt_Pleco a numeric vector
pt_POET a numeric vector
pt_Trich a numeric vector
pt_Tromb a numeric vector
nt_Chiro a numeric vector
pi_Chiro a numeric vector
pt_Chiro a numeric vector
pi_Ortho a numeric vector
pi_Tanyt a numeric vector
pi_Tanyp a numeric vector
pi_COC2Chi a numeric vector
pi_ChCr2Chi a numeric vector
pi_Orth2Chi a numeric vector
pi_Tanyp2Chi a numeric vector
pi_ChiroAnne a numeric vector
pi_dom02_BCG_att456_NoJugaRiss a numeric vector
nt_NonIns_BCG_att456 a numeric vector
pi_NonIns_BCG_att456 a numeric vector
pt_NonIns_BCG_att456 a numeric vector
nt_NonInsJugaRiss_BCG_att456 a numeric vector
pi_NonInsJugaRiss_BCG_att456 a numeric vector
pt_NonInsJugaRiss_BCG_att456 a numeric vector
pi_SimBtri a numeric vector
pi_Colesens a numeric vector
nt_longlived a numeric vector
nt_noteworthy a numeric vector
nt_ffg2_pred a numeric vector
ni_Noto a numeric vector
nt_ti_stenocold a numeric vector

nt_ti_cold a numeric vector
nt_ti_cool a numeric vector
nt_ti_warm a numeric vector
nt_ti_stenowarm a numeric vector
nt_ti_eury a numeric vector
nt_ti_cowa a numeric vector
nt_ti_na a numeric vector
nt_ti_stenocold_cold a numeric vector
nt_ti_stenocold_cold_cool a numeric vector
nt_ti_cowa_warm_stenowarm a numeric vector
nt_ti_warm_stenowarm a numeric vector
pi_ti_stenocold a numeric vector
pi_ti_cold a numeric vector
pi_ti_cool a numeric vector
pi_ti_warm a numeric vector
pi_ti_stenowarm a numeric vector
pi_ti_eury a numeric vector
pi_ti_cowa a numeric vector
pi_ti_na a numeric vector
pi_ti_stenocold_cold a numeric vector
pi_ti_stenocold_cold_cool a numeric vector
pi_ti_cowa_warm_stenowarm a numeric vector
pi_ti_warm_stenowarm a numeric vector
pt_ti_stenocold a numeric vector
pt_ti_cold a numeric vector
pt_ti_cool a numeric vector
pt_ti_warm a numeric vector
pt_ti_stenowarm a numeric vector
pt_ti_eury a numeric vector
pt_ti_cowa a numeric vector
pt_ti_na a numeric vector
pt_ti_stenocold_cold a numeric vector
pt_ti_stenocold_cold_cool a numeric vector
pt_ti_cowa_warm_stenowarm a numeric vector
pt_ti_warm_stenowarm a numeric vector
ri_ti_sccc_wsw a numeric vector
nt_tv_intol a numeric vector

nt_tv_intol4 a numeric vector
nt_tv_tolcr a numeric vector
pi_tv_intol a numeric vector
pi_tv_intol4 a numeric vector
pi_tv_tolcr a numeric vector
pi_tv_tolcr6 a numeric vector
pt_tv_intol a numeric vector
pt_tv_intol4 a numeric vector
pt_tv_tolcr a numeric vector
nt_tv_intol4_EPT a numeric vector
nt_tv_ntol a numeric vector
nt_tv_stol a numeric vector
pi_tv_ntol a numeric vector
pi_tv_stol a numeric vector
pt_tv_ntol a numeric vector
pt_tv_stol a numeric vector
pi_tv2_intol a numeric vector
pi_tv2_tolcr_ISA_SalHi_xFL a logical vector
pi_tv2_intol_ISA_SalHi_xFL a logical vector
pt_tv2_intol_ISA_SalHi_xFL a logical vector
nt_ffg_col a numeric vector
nt_ffg_filt a numeric vector
nt_ffg_pred a numeric vector
nt_ffg_scrap a numeric vector
nt_ffg_shred a numeric vector
nt_ffg_mah a numeric vector
nt_ffg_omn a numeric vector
nt_ffg_par a numeric vector
nt_ffg_pih a numeric vector
nt_ffg_xyl a numeric vector
nt_ffg_pred_scrap_shred a numeric vector
pi_ffg_col a numeric vector
pi_ffg_filt a numeric vector
pi_ffg_pred a numeric vector
pi_ffg_scrap a numeric vector
pi_ffg_shred a numeric vector
pi_ffg_mah a numeric vector

pi_ffg_omn a numeric vector
pi_ffg_par a numeric vector
pi_ffg_pih a numeric vector
pi_ffg_xyl a numeric vector
pi_ffg_col_filt a numeric vector
pt_ffg_col a numeric vector
pt_ffg_filt a numeric vector
pt_ffg_pred a numeric vector
pt_ffg_scrap a numeric vector
pt_ffg_shred a numeric vector
pt_ffg_mah a numeric vector
pt_ffg_omn a numeric vector
pt_ffg_par a numeric vector
pt_ffg_pih a numeric vector
pt_ffg_xyl a numeric vector
nt_ffg2_intface a logical vector
nt_ffg2_subsurf a logical vector
pi_ffg2_scavburr a logical vector
nt_habit_burrow a numeric vector
nt_habit_climb a numeric vector
nt_habit_cling a numeric vector
nt_habit_sprawl a numeric vector
nt_habit_swim a numeric vector
pi_habit_burrow a numeric vector
pi_habit_climb a numeric vector
pi_habit_cling a numeric vector
pi_habit_sprawl a numeric vector
pi_habit_swim a numeric vector
pt_habit_burrow a numeric vector
pt_habit_climb a numeric vector
pt_habit_cling a numeric vector
pt_habit_sprawl a numeric vector
pt_habit_swim a numeric vector
pi_habit_cling_PlecoNoCling a numeric vector
nt_volt_multi a numeric vector
nt_volt_semi a numeric vector
nt_volt_uni a numeric vector

pi_volt_multi a numeric vector
pi_volt_semi a numeric vector
pi_volt_uni a numeric vector
pt_volt_multi a numeric vector
pt_volt_semi a numeric vector
pt_volt_uni a numeric vector
pi_dom01 a numeric vector
pi_dom02 a numeric vector
pi_dom03 a numeric vector
pi_dom04 a numeric vector
pi_dom05 a numeric vector
pi_dom06 a numeric vector
pi_dom07 a numeric vector
pi_dom08 a numeric vector
pi_dom09 a numeric vector
pi_dom10 a numeric vector
x_Becks a numeric vector
x_Becks3 a numeric vector
x_HBI a numeric vector
x_HBI2 a numeric vector
x_NCBI a numeric vector
x_Shan_e a numeric vector
x_Shan_2 a numeric vector
x_Shan_10 a numeric vector
x_D a numeric vector
x_D_G a numeric vector
x_D_Mg a numeric vector
x_Evenness a numeric vector
nt_habitat_brac a numeric vector
nt_habitat_depo a numeric vector
nt_habitat_gene a numeric vector
nt_habitat_head a numeric vector
nt_habitat_rheo a numeric vector
nt_habitat_rive a numeric vector
nt_habitat_spec a numeric vector
nt_habitat_unkn a numeric vector
pi_habitat_brac a numeric vector

pi_habitat_depo a numeric vector
pi_habitat_gene a numeric vector
pi_habitat_head a numeric vector
pi_habitat_rheo a numeric vector
pi_habitat_rive a numeric vector
pi_habitat_spec a numeric vector
pi_habitat_unkn a numeric vector
pt_habitat_brac a numeric vector
pt_habitat_depo a numeric vector
pt_habitat_gene a numeric vector
pt_habitat_head a numeric vector
pt_habitat_rheo a numeric vector
pt_habitat_rive a numeric vector
pt_habitat_spec a numeric vector
pt_habitat_unkn a numeric vector
nt_BCG_att1 a numeric vector
nt_BCG_att1i a numeric vector
nt_BCG_att1m a numeric vector
nt_BCG_att12 a numeric vector
nt_BCG_att1i2 a numeric vector
nt_BCG_att123 a numeric vector
nt_BCG_att1i23 a numeric vector
nt_BCG_att1i236i a numeric vector
nt_BCG_att2 a numeric vector
nt_BCG_att23 a numeric vector
nt_BCG_att234 a numeric vector
nt_BCG_att3 a numeric vector
nt_BCG_att4 a numeric vector
nt_BCG_att45 a numeric vector
nt_BCG_att456 a numeric vector
nt_BCG_att5 a numeric vector
nt_BCG_att56 a numeric vector
nt_BCG_att56t a numeric vector
nt_BCG_att6 a numeric vector
nt_BCG_att6i a numeric vector
nt_BCG_att6m a numeric vector
nt_BCG_att6t a numeric vector

nt_BCG_attNA a numeric vector
nt_BCG_att4b a numeric vector
nt_BCG_att4w a numeric vector
nt_BCG_att1i234b5 a numeric vector
nt_BCG_att1i234w5 a numeric vector
nt_Ephem_BCG_att1i2 a numeric vector
nt_EPT_BCG_att123 a numeric vector
nt_EPT_BCG_att1i23 a numeric vector
nt_Pleco_BCG_att1i2 a numeric vector
nt_Trich_BCG_att1i2 a numeric vector
pi_BCG_att1 a numeric vector
pi_BCG_att1i a numeric vector
pi_BCG_att1m a numeric vector
pi_BCG_att12 a numeric vector
pi_BCG_att1i2 a numeric vector
pi_BCG_att123 a numeric vector
pi_BCG_att1i23 a numeric vector
pi_BCG_att1i236i a numeric vector
pi_BCG_att2 a numeric vector
pi_BCG_att23 a numeric vector
pi_BCG_att234 a numeric vector
pi_BCG_att3 a numeric vector
pi_BCG_att4 a numeric vector
pi_BCG_att45 a numeric vector
pi_BCG_att456 a numeric vector
pi_BCG_att5 a numeric vector
pi_BCG_att5extra a numeric vector
pi_BCG_att56 a numeric vector
pi_BCG_att56t a numeric vector
pi_BCG_att6 a numeric vector
pi_BCG_att6i a numeric vector
pi_BCG_att6m a numeric vector
pi_BCG_att6t a numeric vector
pi_BCG_attNA a numeric vector
pi_BCG_att4b a numeric vector
pi_BCG_att4w a numeric vector
pi_BCG_att1i234b5 a numeric vector

pi_BCG_att1i234w5 a numeric vector
pi_EPT_BCG_att123 a numeric vector
pi_EPT_BCG_att1i23 a numeric vector
pt_BCG_att1 a numeric vector
pt_BCG_att1i a numeric vector
pt_BCG_att1m a numeric vector
pt_BCG_att12 a numeric vector
pt_BCG_att1i2 a numeric vector
pt_BCG_att123 a numeric vector
pt_BCG_att1i23 a numeric vector
pt_BCG_att1i236i a numeric vector
pt_BCG_att2 a numeric vector
pt_BCG_att23 a numeric vector
pt_BCG_att234 a numeric vector
pt_BCG_att3 a numeric vector
pt_BCG_att4 a numeric vector
pt_BCG_att45 a numeric vector
pt_BCG_att456 a numeric vector
pt_BCG_att5 a numeric vector
pt_BCG_att56 a numeric vector
pt_BCG_att56t a numeric vector
pt_BCG_att6 a numeric vector
pt_BCG_att6i a numeric vector
pt_BCG_att6m a numeric vector
pt_BCG_att6t a numeric vector
pt_BCG_attNA a numeric vector
pt_BCG_att4b a numeric vector
pt_BCG_att4w a numeric vector
pt_BCG_att1i234b5 a numeric vector
pt_BCG_att1i234w5 a numeric vector
pt_EPT_BCG_att123 a numeric vector
pt_EPT_BCG_att1i23 a numeric vector
x_UFC a numeric vector
nt_elev_low a numeric vector
nt_elev_high a numeric vector
nt_grad_low a numeric vector
nt_grad_mod a numeric vector

nt_grad_high a numeric vector
nt_wsarea_small a numeric vector
nt_wsarea_medium a numeric vector
nt_wsarea_large a numeric vector
nt_wsarea_xlarge a numeric vector
nt_habstruct_coarsesub a numeric vector
nt_habstruct_noflow a numeric vector
nt_habstruct_rootmat a numeric vector
nt_habstruct_snag a numeric vector
nt_habstruct_NA a numeric vector
pi_habstruct_coarsesub a numeric vector
pi_habstruct_noflow a numeric vector
pi_habstruct_rootmat a numeric vector
pi_habstruct_snag a numeric vector
pi_habstruct_NA a numeric vector
pt_habstruct_coarsesub a numeric vector
pt_habstruct_noflow a numeric vector
pt_habstruct_rootmat a numeric vector
pt_habstruct_snag a numeric vector
pt_habstruct_NA a numeric vector
nval_habstruct a numeric vector
nfam_Coleo a numeric vector
nfam_Ephem a numeric vector
nfam_Odon a numeric vector
nfam_Trich a numeric vector
ngen_Coleo a numeric vector
ngen_Ephem a numeric vector
ngen_Odon a numeric vector
ngen_Trich a numeric vector
ngen_Elmid a numeric vector
nt_NonInsArachDeca_BCG_att456 a numeric vector
pi_NonInsArachDeca_BCG_att456 a numeric vector
pt_NonInsArachDeca_BCG_att456 a numeric vector
nt_NonInsArachDecaJugaRiss_BCG_att456 a numeric vector
pi_NonInsArachDecaJugaRiss_BCG_att456 a numeric vector
pt_NonInsArachDecaJugaRiss_BCG_att456 a numeric vector
nt_NonInsTrombJuga_BCG_att456 a numeric vector
pi_NonInsTrombJuga_BCG_att456 a numeric vector
pt_NonInsTrombJuga_BCG_att456 a numeric vector
nt_oneind a numeric vector
pt_oneind a numeric vector

TaxaMaster_Ben_BCG_PugLowWilVal

Benthic Master Taxa List, BCG Puget Lowlands Willamette Valley

Description

Master taxa list for BCG model for Puget Lowlands Willamette Valley; current as of 2018-09-27.

This list is included for demonstration purposes only! Contact the proper entities to get a updated list for any analyses.

Usage

TaxaMaster_Ben_BCG_PugLowWilVal

Format

A data frame with 684 rows and 20 variables:

TaxaID unique taxonomic identifier

Phylum Taxonomic rank, Phylum

SubPhylum Taxonomic rank, SubPhylum

Class Taxonomic rank, Class

SubClass Taxonomic rank, SubClass

Order Taxonomic rank, Order

SuperFamily Taxonomic rank, SuperFamily#'

Family Taxonomic rank, Family

Tribe Taxonomic rank, Tribe

Genus Taxonomic rank, Genus

SubGenus Taxonomic rank, SubGenus

Species Taxonomic rank, Species

BCG_Attr BCG Attribute

NonTarget Non-Target True/False

Thermal_Indicator thermal tolerance indicator

Long_Lived Long-lived indicator

FFG Function Feeding Group

Habit Habit

Life_Cycle Voltinism

TolVal Tolerance Value, all 7's, demonstration only

Details

To export to file use the code below.

```
write.csv(TaxaMaster_Ben_BCG_PugLowWilVal , "TaxaMaster_Ben_BCG_PugLowWilVal_20180314.csv")
```

Index

* **datasets**

metrics_values, [11](#)

TaxaMaster_Ben_BCG_PugLowWilVal,
[24](#)

BCG.Level.Assignment, [3](#)

BCG.Level.Membership, [6](#)

BCG.Metric.Membership, [9](#)

BCGcalc (BCGcalc-package), [2](#)

BCGcalc-package, [2](#)

metrics_values, [11](#)

TaxaMaster_Ben_BCG_PugLowWilVal, [24](#)