Package: bcgwcat (via r-universe)

August 18, 2024

Type Package

Title BC Groundwater Chemistry Analysis Tool

Version 0.5.1

Description Set of tools for working with groundwater chemstry data from the BC Government Environmental Monitoring System (EMS). Download EMS data and a) export for use in AquaChem, b) view water quality summaries, c) create piper and stiff plots.

License file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.3.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports dplyr (>= 1.1.0), DT (>= 0.5), ggrepel, gridGraphics (>= 0.5.0), lubridate (>= 1.7.4), openxlsx (>= 4.1.0.1), purrr (>= 0.3.4), readr (>= 1.3.1), rems (>= 0.8.0), rlang (>= 0.1.2), shiny (>= 1.3.2), shinydashboard (>= 0.7.1), shinyjs (>= 1.0), smwrBase (>= 1.1.5), smwrGraphs (>= 1.1.4.9000), stringr (>= 1.4.0), tidyr (>= 1.0.0), ggplot2 (>= 3.3.2), viridisLite (>= 0.3.0)

Remotes bcgov/rems, USGS-R/smwrBase, USGS-R/smwrGraphs

Suggests roxygen2, testthat (>= 2.1.0), knitr, rmarkdown, shinytest2

VignetteBuilder knitr

Config/testthat/edition 3

URL https://bcgov.github.io/bcgwcat/, https://github.com/bcgov/bcgwcat/

Repository https://steffilazerte.r-universe.dev

RemoteUrl https://github.com/bcgov/bcgwcat

RemoteRef HEAD

RemoteSha a9910dd0d0abb8f6ab1b6222365c73edab64eb96

Contents

`charge_balance` *Calculate charge balance*

Description

Calculates charge balances based on ALS formula. **Note:** Original EMS charge balances, anion sums and cation sums have been **omitted**.

Usage

```
charge_balance(d)
```

Arguments

`d`	Data set formatted for AquaChem (output of `rems_to_aquachem()`)

Details

Potential changes in workflows over the years have made it difficult to ascertain exactly how charge balances were calculated in older samples. This resulted in discrepancies between EMS and locally calculated charge balances. Therefore for consistency, we calculate charge balances for all samples using the ALS formula below.

One difference between this calculation and that of ALS, is that we use more significant digits when calculating MEQ.

anion sum = Cl_meq + SO4_meq + F_meq + NO3_meq + NO2_meq + Means_Alk_meq

cation sum = Ca_meq + Mg_meq + Na_meq + K_meq + Al_diss_meq + Cu_diss_meq + Fe_diss_meq + Mn_diss_meq + Zn_diss_meq + NH4_meq + (10 ^ (-pH_lab)) * 1000

Charge balance = 100 x (Cation Sum - Anion sum) / (Cation Sum + Anion Sum)

Missing values are ignored (ie. generally treated as 0). However, if all values for cations or anions are missing the charge balance is `NA`.

Value

Data frame

`gw_app` *Launch Shiny app*

Description

Interactive interface for filtering and formatting data, summarizing water quality and creating piper and stiff plots.

Usage

```
gw_app()
```

Examples

```
## Not run: gw_app()
```

`meq` *Calculate MEQ values*

Description

Calculates MEQ values for 'long' data in mg/L. Expects columns: "aqua_code" and "RESULT", where "aqua_code" is the parameter type (e.g., "Zn_diss") and RESULT is the numeric concentration in mg/L.

Usage

```
meq(d, drop_na = FALSE)
```

Arguments

`d`	Data frame. Long data containing parameters and results
`drop_na`	Logical. Whether to omit missing parameters

Details

For conversion details see the included data frame, `meq_conversion`.

Also see `?meq_conversion` for a description of the data.

MEQs are calculated by dividing the parameter concentration in mg/L by the conversion factor.

Note: This is an internal function, exported for clarity in calculations

Examples

```
d <- data.frame(aqua_code = c("Cl", "HCO3"), RESULT = c(5.7, 38.3))
d
meq(d, drop_na = TRUE)
meq(d)
```

`meq_conversion` *Details used to calculate milli-equivalents per litre*

Description

A dataset containing the atomic mass, valency state and corresponding MEQ conversion factor for relevant parameters.

Usage

```
meq_conversion
```

Format

A data frame 4 variables:

param Parameter name (corresponds to AquaChem names)

mass Atomic mass (https://en.wikipedia.org/wiki/List_of_elements_by_atomic_properties)

valency_state Valency

conversion Conversion constant (mass/valency_state)

`piper_plot` *Create Piper plot*

Description

Create Piper plot

Usage

```
piper_plot(
  d,
  ems_id = NULL,
  group = "ems_id",
  legend = TRUE,
  legend_position = "topleft",
  legend_title = group,
  valid = TRUE,
  plot_data = FALSE,
  point_colour = "viridis",
  point_size = 0.1,
  point_filled = TRUE,
  point_shape = "circle"
)
```

Arguments

d	Data frame. AquaChem formatted dataset
ems_id	Character. Ids to plot if dataset includes more than one
group	Character. Column by which to group data for colour, shape, filled and size.
legend	Logical. Whether to show the legend
legend_position	Character or Numeric.. Location of legend. Must be one of "topleft", "topright", etc. (see ?legend for more options), OR a vector of two numeric values x, and y to specify an exact position.
legend_title	Character. Title of legend. Defaults to group.
valid	Logical. Keep only valid data (charge balances <=10)
plot_data	Logical. Whether to return plot data rather than a plot
point_colour	Character. Colour or colours by which to colour points. Either a single value (applied to all), or a vector of values the same length as the number of groups. Can also be "viridis", which will use the viridis colour scale.
point_size	Numeric. Point size. Either a single value (applied to all), or a vector of values the same length as the number of groups.
point_filled	Logical. Whether to fill point shapes or not. Either a single value (applied to all), or a vector of values the same length as the number of groups.
point_shape	Character. Shape of points to use. Valid options are "circle", "square" or "triangle". Either a single value (applied to all), or a vector of values the same length as the number of groups.

`rems_to_aquachem` *Fetch and convert data from rems to AquaChem format*

Description

Fetch and convert data from rems to AquaChem format

Usage

```
rems_to_aquachem(
  ems_ids,
  date_range = NULL,
  save = TRUE,
  out_folder = "./",
  out_file = NULL,
  interactive = TRUE,
  dont_update = TRUE
)
```

Arguments

`ems_ids`	Character vector. Unique EMS ids
`date_range`	Character vector. Start and end dates (YYYY-MM-DD)
`save`	Logical. Whether or not to save the data as csv.
`out_folder`	Character. Where to save data for AquaChem
`out_file`	Character. What to call data file for AquaChem. Default is aquachem_DATE.csv
`interactive`	Logical. Whether or not to ask when caching data.
`dont_update`	Logical. Whether or not to avoid updating EMS if `interactive` is FALSE

Details

`out_folder` This is how you tell the function where to store the aquachem data.

Value

Outputs an Excel and a CSV file in the format to import into AquaChem.

Examples

```
## Not run:
# Convert one well and save the data in the working directory
rems_to_aquachem(ems_ids = "E289551")

# Convert several wells (this will overwrite the previous data!)
rems_to_aquachem(ems_ids = c("1401030", "1401377"))
```

```
# To specify a date range (Year-Month-Day)
rems_to_aquachem(ems_ids = c("1401030", "1401377"),
                 date_range = c("2000-01-01", "2015-01-01"))

# To name the ouput (extension should be .csv, .txt, or .dat)
rems_to_aquachem(ems_ids = "1401030", out_file = "water_quality01.csv")

# To save the output to a specific folder
# (here, the Outputs folder in the Rcode folder on the H drive)
# Note that the out_folder must exist or you'll get an error
rems_to_aquachem(ems_ids = c("1401030", "1401377"),
                 out_folder = "H:\\Rcode\\Outputs/")

# All together now!
rems_to_aquachem(ems_ids = c("1401030", "1401377"),
                 date_range = c("2000-01-01", "2015-01-01"),
                 out_file = "water_quality05.csv",
                 out_folder = "H:\\Rcode\\Outputs/")

# Clean up
unlink("water_quality01.csv")
unlink(paste0("aquachem_", Sys.Date(), ".csv"))

## End(Not run)
```

`stiff_plot` *Create Stiff plot*

Description

Create Stiff plot

Usage

```
stiff_plot(d, ems_id = NULL, colour = TRUE, legend = TRUE, valid = TRUE)
```

Arguments

`d`	AquaChem formatted dataset
`ems_id`	Ids to plot if dataset includes more than one
`colour`	Whether to add colour by ems_id
`legend`	Whether to show the legend
`valid`	Logical. Keep only valid data (charge balances <=10)

units_remove *Remove units*

Description

The main `rems_to_aquachem()` function downloads EMS data and formats it for use in the external program, AquaChem. However, occasionally you may wish to work with this formatted EMS data in R. This function removes the extra 'units' row and then converts the columns to make the data usable in R.

Usage

```
units_remove(d)
```

Arguments

`d`	Data frame output from `rems_to_aquachem()`

Value

Data frame

Examples

```
## Not run:
# Get and format one well for use in AquaChem
r <- rems_to_aquachem(ems_ids = "E289551", save = FALSE)

# Remove units and convert columns to appropriate formats for use in R
r <- units_remove(r)


## End(Not run)
```

water_quality *Assess problems with water quality*

Description

Assess problems with water quality

Usage

```
water_quality(d)
```

Arguments

d	EMS data formated for AquaChem as returned by `rems_to_aquachem()`

Value

Tibble of sites, samples, parameters, water quality limits and problems.

---

water_type *Calculate water type*

Description

Water type based on anions Cl, SO4, HCO3 and cations Ca, Mg, Na and K. Elements are ranked by proportion MEQ, all greater than 10% are listed in descending order of presence, cations first. Water type is only calculated for samples with valid charge balances. Missing ions are ignored (i.e. treated as 0).

Usage

```
water_type(d)
```

Arguments

d	Data frame. Must contain columns `Sample_Date`, `SampleID`, `StationID`, `Cl_meq`, `SO4_meq`, `HCO3_meq`, `Ca_meq`, `Mg_meq`, `Na_meq`, `K_meq`, and `charge_balance`.

Value

Data frame with added column `water_type`.

Examples

```
d <- data.frame(Sample_Date = "2022-01-01", SampleID = "999990-01", StationID = 000,
                Cl_meq = 0.0226, SO4_meq = 0.0208, HCO3_meq = 1.54,
                Ca_meq = 0.187, Mg_meq = 0.490, Na_meq = 0.465, K_meq = 0.0665,
                charge_balance = 0.5)

d <- water_type(d)
d
```

Index