Package: hydrochem (via r-universe)

October 23, 2026

Title Analysis and Visualization Tools for Hydrochemical Analysis

Version 0.1.0

Description Provides a comprehensive suite of tools for processing, analyzing, and visualizing hydrochemical data in a reproducible and programmatic workflow. Implements unit conversion, censored data management, ionic balance calculation, water type classification, and a range of hydrochemical indices relevant to water quality assessment. Offers advanced visualization functions for generating Piper (Piper, 1944, <doi:10.1029/TR025i006p00914>), Durov (Durov, 1948), Stiff (Stiff, 1951, <doi:10.2118/951376-G>), Collins (Collins, 1923, <doi:10.1021/ie50160a030>), Schoeller (Schoeller, 1935), Gibbs (Gibbs, 1970, <doi:10.1126/science.170.3962.1088>), and ternary diagrams — the standard graphical tools used in hydrochemical interpretation. Unlike existing solutions that rely on spreadsheets or proprietary graphical software, 'hydrochem' is designed for scalability and reproducibility, lowering the technical barrier for hydrochemists working with datasets of varying complexity. It integrates seamlessly into modern R workflows and supports rigorous water resource management and research.

License GPL-2

Depends R (>= 4.1.0)

Imports cowplot (>= 1.2.0), data.table (>= 1.16.2), ggplot2 (>= 3.5.1), grDevices, scales (>= 1.4.0), stats

Suggests spelling

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Guillaume Cinkus [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2877-6551>), Raphaël Bondu [aut]
(ORCID: <https://orcid.org/0000-0003-3838-6693>)

Maintainer Guillaume Cinkus <guillaume.cinkus@gmail.com>

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

Date/Publication 2026-06-24 08:37:08 UTC

RemoteUrl https://github.com/cran/hydrochem

RemoteRef HEAD

RemoteSha 690fa930c20662a4430b126562ee5125c77c01d4

Contents

cai1 . . . 3
cai2 . . . 3
convert_unit . . . 4
data_qc . . . 5
hc_data . . . 6
ionic_balance . . . 6
kelly_ratio . . . 7
magnesium_hazard . . . 8
percent_sodium . . . 9
permeability_index . . . 9
plot_biplot . . . 10
plot_collins . . . 14
plot_durov . . . 16
plot_gibbs . . . 18
plot_piper . . . 20
plot_schoeller . . . 23
plot_stabler . . . 25
plot_stiff . . . 26
plot_ternary . . . 29
replace_bdl . . . 32
rsc . . . 33
sar . . . 33
save_plot . . . 34
stat_summary . . . 35
tds . . . 36
total_alkalinity . . . 37
total_hardness . . . 38
water_type . . . 39

Index **40**

cai1 *Calculate Chloro-Alkaline Index 1 (CAI-1)*

Description

Computes the Chloro-Alkaline Index 1 (CAI-1) from hydrochemical data using sodium, potassium, and chloride concentrations. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"` and are internally converted to `"meq/L"` prior to calculation.

Usage

```
cai1(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

`data`	A `data.frame` or `data.table` containing the required ion columns `"Na"`, `"K"`, and `"Cl"`.
`base_unit`	Unit of the input ion concentrations. One of `"mg/L"`, `"mmol/L"`, or `"meq/L"`. Default is `"mg/L"`.

Details

The function checks that all required ions (`Na`, `K`, `Cl`) are present in the input dataset and throws an error if any are missing.

Value

A numeric vector of CAI-1 values, one per row of `data`.

Examples

```
data("hc_data", package = "hydrochem")
cai1(hc_data, base_unit = "mg/L")
```

cai2 *Calculate Chloro-Alkaline Index 2 (CAI-2)*

Description

Computes the Chloro-Alkaline Index 2 (CAI-2) from hydrochemical data using sodium, potassium, chloride, sulfate, bicarbonate, nitrate, and carbonate concentrations. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"` and are internally converted to `"meq/L"` prior to calculation.

Usage

```
cai2(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Na", "K", "Cl", "SO4", "HCO3", and "NO3".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Details

The function checks that all required ions (Na, K, Cl, SO4, HCO3, NO3) are present in the input dataset and throws an error if any are missing. If carbonate (CO3) is also provided, its contribution is added to bicarbonate after conversion to milliequivalents (meq/L). If CO3 is absent, alkalinity is considered to be represented solely by bicarbonate (HCO3).

Value

A numeric vector of CAI-2 values, one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
cai2(hc_data, base_unit = "mg/L")
```

---

convert_unit *Convert Ion Concentrations Between Units (mg/L, mmol/L, meq/L)*

Description

Converts concentrations of selected ions (cations and anions) between supported units: "mg/L", "mmol/L", and "meq/L". Input columns must be named using plain ion names (e.g., "Ca", "Cl"). Output columns are added with suffixes: _mg, _mmol, and _meq depending on the to_unit function parameter.

Usage

```
convert_unit(
  data,
  ions = c("Ca", "Mg", "Na", "K", "Cl", "SO4", "HCO3"),
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  to_unit = c("mg/L", "mmol/L", "meq/L"),
  convert_to_species = FALSE
)
```

Arguments

data	A data.frame or data.table with numeric columns named after ions (e.g., "Ca", "Cl").
ions	A character vector of ions names. Default is c("Ca", "Mg", "Na", "K", "Cl", "SO4", "HCO3").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
to_unit	Unit of the output unit. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
convert_to_species	Logical. If TRUE, converts known elemental forms (e.g., "NO3_N", "PO4_P", "H2S_HS_S") to their compound species ("NO3", "PO4", "H2S").

Value

A data.table with original columns and new columns suffixed based on the to_unit function parameter.

Examples

```
data("hc_data", package = "hydrochem")
convert_unit(hc_data, base_unit = "mg/L", to_unit = "meq/L")
```

---

data_qc *Hydrochemical dataset from Quebec*

Description

A dataset containing ion concentrations in mg/L for applying hydrochemical tools.

Usage

```
data_qc
```

Format

A data frame with 146 rows of hydrochemical data.

Source

Cloutier V. (2004) Origin and goechemical evolution of groundwater in the Paleozoic Basses-Laurentides sedimentary rock aquifer system, St. Lawrence Lowlands, Québec, Canada. PhD Thesis, INRS-Eau, Terre & Environnement, Québec, Canada (electronic version on https://espace.inrs.ca/id/eprint/1445/)

Examples

```
data(data_qc)
head(data_qc)
```

`hc_data` *Example hydrochemical dataset*

Description

A sample dataset containing ion concentrations in mg/L for applying hydrochemical tools.

Usage

```
hc_data
```

Format

A data frame with 59 rows of hydrochemical data.

Source

Anonymized field data

Examples

```
data(hc_data)
head(hc_data)
```

`ionic_balance` *Calculate Ionic Balance Error (%)*

Description

Computes the ionic balance error for each row of a hydrochemical dataset.

Usage

```
ionic_balance(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  method = c("major", "major_minor")
)
```

Arguments

data	A data.frame or data.table with numeric columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
method	A character string specifying which ions to include. "major" uses only major ions; "major_minor" includes both major and minor ions.

Details

The ionic balance error is calculated in percent:

$$IB = ((T_c - T_a)/(T_c + T_a)) * 100$$

where T_c is the total concentration of cations in meq/L, and T_a is the total concentration of anions in meq/L.

Value

A numeric vector of ionic balance error (in %), one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
hc_data$ib <- ionic_balance(hc_data,
                            base_unit = "mg/L")
```

---

kelly_ratio *Calculate Kelly Ratio*

Description

Computes the Kelly Ratio from hydrochemical data using sodium, calcium, and magnesium concentrations. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L", and are internally converted to "meq/L" before calculation.

Usage

```
kelly_ratio(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Na", "Ca", and "Mg".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Details

The function checks that all required ions (Na, Ca, Mg) are present in the input dataset and throws an error if any are missing.

Value

A numeric vector of Kelly Ratio values, one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
kelly_ratio(hc_data, base_unit = "mg/L")
```

---

magnesium_hazard *Calculate Magnesium Hazard (MH)*

Description

Computes the Magnesium Hazard (MH) index from hydrochemical data using calcium and magnesium concentrations. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation.

Usage

```
magnesium_hazard(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Ca" and "Mg".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Details

The function checks that both required ions (Ca, Mg) are present in the input dataset and throws an error if either is missing.

Value

A numeric vector of Magnesium Hazard values (in %), one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
magnesium_hazard(hc_data, base_unit = "mg/L")
```

percent_sodium *Calculate Percent Sodium (%Na)*

Description

Computes the Percent Sodium (%Na) index from hydrochemical data using sodium, potassium, calcium, and magnesium concentrations. Input values may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation.

Usage

```
percent_sodium(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Na", "K", "Ca", and "Mg".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Details

The function checks that all required ions (Na, K, Ca, Mg) are present in the input dataset and throws an error if any are missing.

Value

A numeric vector of Percent Sodium values (in %), one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
percent_sodium(hc_data, base_unit = "mg/L")
```

permeability_index *Calculate Permeability Index (PI)*

Description

Computes the Permeability Index (PI) from hydrochemical data using sodium, bicarbonate, calcium, and magnesium concentrations. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation.

Usage

```
permeability_index(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Na", "HCO3", "Ca", and "Mg".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Details

The function checks that all required ions (Na, HCO3, Ca, Mg) are present in the dataset and throws an error if any are missing.

Value

A numeric vector of Permeability Index values (in %), one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
permeability_index(hc_data, base_unit = "mg/L")
```

---

plot_biplot *Plot Biplots from Hydrochemical Data*

Description

Generates simple XY biplots from hydrochemical data using user-defined variable pairs. Pairs are provided as character strings using the "y ~ x" syntax and #' can include arithmetic expressions (e.g., "Ca + Mg ~ Cl" or "Na / Cl ~ SO4"). Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_biplot(
  data,
  pairs,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  to_unit = c("meq/L", "mmol/L", "mg/L"),
  ncol = NULL,
  group = NULL,
  group_custom = FALSE,
  grid_lines = TRUE,
  slope = NA,
  intercept = NA,
  slope_color = "black",
  xmin = NA,
```

```
  xmax = NA,
  ymin = NA,
  ymax = NA,
  log_x = FALSE,
  log_y = FALSE,
  shape = 21,
  color = "black",
  fill = "grey40",
  size = 2,
  label = NULL,
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0,
  legend.position = "bottom",
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A `data.frame` or `data.table` with columns named after ions (e.g., `"Ca"`, `"Cl"`, `"SO4"`).
pairs	Character vector of biplot definitions using the `"y ~ x"` syntax. Each element defines one panel. Arithmetic expressions are allowed on either side, e.g. `"Ca + Mg ~ Cl"`, `"Na * K ~ Cl"`, `"Na - K ~ Cl"`, or `"SO4 / Cl ~ NO3"`. Whitespace is ignored.
base_unit	Unit of the input ion concentrations. One of `"mg/L"`, `"mmol/L"`, or `"meq/L"`. Default is `"mg/L"`.
to_unit	Unit of the output ion concentrations. One of `"meq/L"`, `"mmol/L"`, or `"mg/L"`. Default is `"meq/L"`.
ncol	Integer. Number of columns for the cowplot grid layout. Default is `NULL` (automatic layout).
group	Optional column name in `data` used to group points. If `NULL`, a single style is used.
group_custom	Logical. If `TRUE`, enables manual aesthetics per group defined in `group`. In that case, at least one of `color`, `fill`, `shape`, or `size` must be a vector of length equal to the number of groups. Missing or wrongly sized vectors are filled with defaults (`"black"`, `"transparent"`, `21`, `2`). If `FALSE` (default), these aesthetics must be length 1.
grid_lines	Logical. Draw grid lines on panels. Default `TRUE`.
slope	Numeric or numeric vector. Slope of a reference line drawn on the biplots. If length 1, the same slope is applied to all panels. If a vector, it must have the same length as `pairs`, with `NA` indicating that no line is drawn for a given panel. Must be provided together with `intercept`. Default is `NULL` (no reference line). Horizontal and vertical lines can be displayed using 0 and Inf, respectively.

intercept	Numeric or numeric vector. Intercept of a reference line drawn on the biplots. If length 1, the same intercept is applied to all panels. If a vector, it must have the same length as pairs, with NA indicating that no line is drawn for a given panel. Must be provided together with slope. Default is NULL (no reference line).
slope_color	Character. Color of the reference line. Default is "black".
xmin	Numeric or numeric vector. Minimum value of the x-axis. If length 1, the same limit is applied to all biplots. If a vector, it must have the same length as pairs, with NA indicating that the limit is derived from the data for a given panel. Default is NA.
xmax	Numeric or numeric vector. Maximum value of the x-axis. If length 1, the same limit is applied to all biplots. If a vector, it must have the same length as pairs, with NA indicating that the limit is derived from the data for a given panel. Default is NA.
ymin	Numeric or numeric vector. Minimum value of the y-axis. If length 1, the same limit is applied to all biplots. If a vector, it must have the same length as pairs, with NA indicating that the limit is derived from the data for a given panel. Default is NA.
ymax	Numeric or numeric vector. Maximum value of the y-axis. If length 1, the same limit is applied to all biplots. If a vector, it must have the same length as pairs, with NA indicating that the limit is derived from the data for a given panel. Default is NA.
log_x	Logical or logical vector. If TRUE, the x-axis is displayed on a base-10 logarithmic scale. If length 1, the same setting is applied to all biplots. If a vector, it must have the same length as pairs, with FALSE indicating a linear scale for a given panel. Default is FALSE.
log_y	Logical or logical vector. If TRUE, the y-axis is displayed on a base-10 logarithmic scale. If length 1, the same setting is applied to all biplots. If a vector, it must have the same length as pairs, with FALSE indicating a linear scale for a given panel. Default is FALSE.
shape	Integer or integer vector of point shapes (ggplot2 standards). Default 21.
color	Character. Point outline color. Default "black".
fill	Character. Point fill color. Default "grey40".
size	Numeric or numeric vector. Point size(s). Default 2.
label	Optional column name in data used to label points. Default is NULL (no labels).
label_size	Numeric. Size of the label text. Default is 3.
label_nudge_x	Numeric. Horizontal offset of labels relative to points. Default is 0.
label_nudge_y	Numeric. Vertical offset of labels relative to points. Default is 0.
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A combined cowplot object containing the biplots (and legend if grouping is used).

Examples

```
data("data_qc", package = "hydrochem")

# Replace below detection limit values
data <- replace_bdl(data_qc)

# Basic biplots
plot_biplot(
  data,
  pairs = c("Ca ~ Mg", "SO4 ~ Cl", "Cl + SO4 ~ NO3_N")
)

# Grouped biplots
plot_biplot(
  data,
  pairs = c("Ca ~ HCO3", "Na ~ Cl"),
  group = "Cluster"
)

# Custom aesthetics per group
plot_biplot(
  data,
  pairs = c("Ca ~ HCO3", "Na + K ~ Cl"),
  group = "Hydro_cond",
  group_custom = TRUE,
  fill = c("steelblue", "tomato", "darkgreen", "yellow"),
  color = c("black", "black", "black", "black"),
  shape = c(21, 22, 24, 25),
  size  = c(2.5, 3, 3.5, 4),
  ncol = 2
)

# Same reference line (1:1) on all biplots
plot_biplot(
  data,
  pairs = c("Ca ~ Ca", "Na ~ Na"),
  slope = 1,
  intercept = 0
)

# Reference line on a single panel only
plot_biplot(
  data,
  pairs = c("Ca ~ Ca", "Ca + Mg ~ HCO3", "Cl ~ Na"),
  slope     = c(NA, NA, 1),
  intercept = c(NA, NA, 0)
)
```

```
# Log scale on x-axis for all biplots
plot_biplot(
  data,
  pairs = c("Ca ~ Mg", "Cl ~ Na"),
  log_x = TRUE
)

# Log scale on selected panels only
plot_biplot(
  data,
  pairs = c("Ca ~ Mg", "Ca + Mg ~ HCO3", "Cl ~ Na"),
  log_x = c(FALSE, TRUE, FALSE),
  log_y = c(FALSE, FALSE, TRUE)
)
```

`plot_collins` *Plot Collins Diagrams from Hydrochemical Data*

Description

Generates Collins diagrams from hydrochemical data. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"` and are internally converted to `"meq/L"` prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_collins(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  include_NO3 = FALSE,
  ratio = 0.05,
  idcol = NULL,
  nrow = NULL,
  ncol = NULL,
  facet_dir = c("v", "h"),
  ylab = "%meq/L",
  base_size = 12,
  border_color = "transparent",
  border_width = 0.5,
  axis_title_size = 14,
  label_border_color = "transparent",
  plot_title = NULL,
  legend.position = "right",
  legend.title = NULL,
  show_label = TRUE,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
include_NO3	Logical. If TRUE, nitrate (NO3) is included. The NO3 column must be present in the input data. Default is FALSE.
ratio	Numeric. Aspect ratio between x and y axes in the plot. Default is 0.05.
idcol	Character or NULL. Column name used to identify and facet individual diagrams. If NULL, synthetic sample IDs are generated. Default is NULL.
nrow, ncol	Integers. Number of rows and columns for the facet layout. Default is NULL (automatic layout).
facet_dir	Character. Facet layout direction. Either "v" for vertical or "h" for horizontal. Default is "v".
ylab	Character. Label for the y-axis. Set NULL for no axis. Default is "%meq/L".
base_size	Numeric. Base font size for plot text. Default is 12.
border_color	Color of the bar border. Default is "transparent".
border_width	Width of the bar border. Default is 0.5.
axis_title_size	Numeric. Font size for strip labels. Default is 14.
label_border_color	Character. Border color for facet strip labels. Use "transparent" for no border. Default is "transparent".
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
show_label	Logical. If TRUE, displays strip labels. Default is TRUE.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object representing the Collins diagram(s).

Examples

```
data("hc_data", package = "hydrochem")
hc_data <- hc_data[1:20, ]
plot_collins(hc_data, idcol = "id")
```

plot_durov *Plot Durov Diagram from Hydrochemical Data*

Description

Generates a Durov diagram from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. If carbonate (CO3) is also provided, its contribution is added to bicarbonate after conversion to milliequivalents (meq/L). If CO3 is absent, alkalinity is considered to be represented solely by bicarbonate (HCO3). Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_durov(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  top_ions = c("Cl", "SO4", "HCO3"),
  left_ions = c("Mg", "Na", "Ca"),
  tds_method = c("major", "major_si", "all_ions"),
  group = NULL,
  group_custom = FALSE,
  grid_lines = TRUE,
  plot_pH = TRUE,
  plot_TDS = TRUE,
  tds_log = FALSE,
  axis_title_size = 3.5,
  axis_tick_label_size = 3,
  shape = 21,
  color = "black",
  fill = "grey40",
  size = 2,
  plot_title = NULL,
  label = NULL,
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0,
  legend.position = "bottom",
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

top_ions	Character vector (length 3) of anions used on the top triangle, default c("Cl","SO4","HCO3"). If "CO3" exists, it is combined with "HCO3" unless "HCO3" is absent.
left_ions	Character vector (length 3) of cations used on the left triangle, default c("Mg","Na","Ca"). If "K" is present, it is typically merged with "Na".
tds_method	TDS estimation for the Durov square: "major" (sum of major ions), "major_si" (major ions incl. silica if present), or "all_ions" (sum of all available ions).
group	Optional column name in data used to group points. If NULL, a single style is used.
group_custom	Logical. If TRUE, enables manual aesthetics per group defined in group. In that case, at least one of color, fill, shape, or size must be a vector of length equal to the number of groups. Missing or wrongly sized vectors are filled with defaults ("black", "transparent", 21, 2). If FALSE (default), these aesthetics must be length 1.
grid_lines	Logical. Draw dashed grid lines on the ternaries and Durov square. Default TRUE.
plot_pH	Logical. If TRUE, adds the optional pH scale plot below the main Durov diagram. Default TRUE.
plot_TDS	Logical. If TRUE, adds the optional TDS scale plot to the right of the main Durov diagram. Default TRUE.
tds_log	Logical. If TRUE, the TDS axis on the right panel is plotted on a base-10 logarithmic scale. Default FALSE.
axis_title_size	Numeric. Size for ion-group labels. Default 3.5.
axis_tick_label_size	Numeric. Size for tick labels. Default 3.
shape	Integer or integer vector of point shapes (ggplot2 standards). Default 21.
color	Character. Point outline color. Default "black".
fill	Character. Point fill color. Default "grey40".
size	Numeric or numeric vector. Point size(s). Default 2.
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
label	Optional column name in data used to label points. Default is NULL (no labels).
label_size	Numeric. Size of the label text. Default is 3.
label_nudge_x	Numeric. Horizontal offset of labels relative to points. Default is 0.
label_nudge_y	Numeric. Vertical offset of labels relative to points. Default is 0.
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object showing the Durov diagram.

Examples

```
data("data_qc", package = "hydrochem")

# Replace below detection limit values
data <- replace_bdl(data_qc)

# Basic Durov plot
plot_durov(data)

# Grouped Durov plot
plot_durov(data, group = "Cluster")

# Custom aesthetics per group
plot_durov(
  data,
  group = "Hydro_cond",
  group_custom = TRUE,
  fill = c("steelblue", "tomato", "darkgreen", "yellow"),
  color = c("black", "black", "black", "black"),
  shape = c(21, 22, 24, 25),
  size  = c(2.5, 3, 3.5, 4)
)
```

---

plot_gibbs *Plot Gibbs Diagram from Hydrochemical Data*

Description

Generates a Gibbs diagram from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_gibbs(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  tds_method = c("major", "major_si", "all_ions"),
  show_guide = TRUE,
  show_text = FALSE,
  group = NULL,
  group_custom = FALSE,
  base_size = 12,
```

```
  shape = 21,
  color = "black",
  fill = "grey40",
  size = 2,
  plot_title = NULL,
  label = NULL,
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0,
  legend.position = c("bottom", "right", "none"),
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
tds_method	Character. Method to compute total dissolved solids (TDS). "major" uses only Na, Ca, Cl, HCO3; "major_si" includes additional major ions (e.g. Mg, SO4); "all_ions" uses all columns. Default is "major".
show_guide	Logical. If TRUE, displays dashed reference lines approximating domain boundaries and transitions within the Gibbs diagram (e.g., evaporation dominance, rock weathering, precipitation influence). Default is TRUE.
show_text	Logical. If TRUE, labels for the dominance zones are displayed on the plot.
group	Optional character string. Column name in data used to group points. If NULL, all points are styled the same. Default is NULL.
group_custom	Logical. If TRUE, allows custom aesthetics (color, fill, shape, size) per group. Must supply vectors of appropriate length. If FALSE, all aesthetics must be scalars. Default is FALSE.
base_size	Numeric. Base font size for the plot. Default is 12.
shape	Integer or vector of integers. Shape(s) for plotting points (ggplot2 standard). Default is 21.
color	Character. Outline color of points. Default is "black".
fill	Character. Fill color(s) for points. Default is "grey40".
size	Numeric. Size of points. Default is 2.
plot_title	Optional character string. Plot title. Default is NULL.
label	Optional column name in data used to label points. Default is NULL (no labels).
label_size	Numeric. Size of the label text. Default is 3.
label_nudge_x	Numeric. Horizontal offset of labels relative to points. Default is 0.
label_nudge_y	Numeric. Vertical offset of labels relative to points. Default is 0.

legend.position
: Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".

legend.title
: A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.

bg_color
: Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object showing the Gibbs diagram.

Examples

```
data("hc_data", package = "hydrochem")

# Basic Gibbs diagram
plot_gibbs(hc_data)

# With grouping
plot_gibbs(hc_data, group = "type")

# With custom fill and shape
plot_gibbs(
  hc_data,
  group = "type",
  group_custom = TRUE,
  fill = c("skyblue", "tomato", "green"),
  shape = c(21, 22, 23)
)

# With all ions included in TDS computation
plot_gibbs(hc_data, tds_method = "all_ions")
```

---

plot_piper *Plot Piper Diagram from Hydrochemical Data*

Description

Generates a Piper diagram from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. If carbonate (CO3) is also provided, its contribution is added to bicarbonate after conversion to milliequivalents (meq/L). If CO3 is absent, alkalinity is considered to be represented solely by bicarbonate (HCO3). Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_piper(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  include_NO3 = FALSE,
  group = NULL,
  group_custom = FALSE,
  grid_lines = TRUE,
  axis_title_size = 3.5,
  axis_tick_label_size = 3,
  bold_label = FALSE,
  shape = 21,
  color = "black",
  fill = "grey40",
  size = 2,
  plot_title = NULL,
  label = NULL,
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0,
  legend.position = "bottom",
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
include_NO3	Logical. If TRUE, nitrate (NO3) is included. The NO3 column must be present in the input data. Default is FALSE.
group	Optional character string. Name of the column in data used to group points in the Piper diagram. If NULL, all points are plotted with the same style. Default is NULL.
group_custom	Logical. If TRUE, enables manual aesthetics for each group defined in group. In this case, at least one of color, fill, shape, or size must be provided as a vector of length equal to the number of groups. Any missing or incorrectly sized vectors will be automatically filled with default values (e.g., "black", "transparent", 21, 2). If FALSE (default), all these aesthetics must be of length 1.
grid_lines	Logical. Whether to include dashed grid lines for the ternary fields and diamond. Default is TRUE.
axis_title_size	Numeric. Text size for the ion group labels. Default is 3.5.

axis_tick_label_size
: Numeric. Text size for the tick value labels. Default is 3.

bold_label
: Logical. If TRUE, renders the ion group labels in bold. Default is FALSE.

shape
: Integer or vector of integers. Shape(s) for plotting points. Must follow ggplot2 shape standards. Default is 21.

color
: Character. Outline color for the plotted points. Default is "black".

fill
: Character. Fill color for the plotted points. Default is "grey40".

size
: Numeric. Size of the plotted points. Default is 2.

plot_title
: Optional character string. Title of the plot. Default is NULL (no title displayed).

label
: Optional column name in data used to label points. Default is NULL (no labels).

label_size
: Numeric. Size of the label text. Default is 3.

label_nudge_x
: Numeric. Horizontal offset of labels relative to points. Default is 0.

label_nudge_y
: Numeric. Vertical offset of labels relative to points. Default is 0.

legend.position
: Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".

legend.title
: A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.

bg_color
: Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object displaying the Piper diagram.

Examples

```
data("hc_data", package = "hydrochem")

# Basic Piper plot without grouping
plot_piper(hc_data)

# Piper plot with simple grouping
plot_piper(hc_data, group = "type")

# Piper plot with custom aesthetics per group
plot_piper(
  hc_data,
  group = "type",
  group_custom = TRUE,
  fill = c("steelblue", "tomato", "darkgreen"),
  color = c("black", "black", "black"),
  shape = c(21, 22, 24),
  size  = c(2.5, 3, 3.5)
)
```

plot_schoeller *Plot Schoeller Diagrams from Hydrochemical Data*

Description

Generates Schoeller diagrams from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_schoeller(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  include_NO3 = FALSE,
  mg_guide = c("full", "simple", "none"),
  group = NULL,
  group_custom = FALSE,
  base_size = 12,
  bold_label = FALSE,
  line_color = "black",
  line_width = 1,
  plot_title = NULL,
  legend.position = "bottom",
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
include_NO3	Logical. If TRUE, nitrate (NO3) is included. The NO3 column must be present in the input data. Default is FALSE.
mg_guide	Character. Display mode for mg/L horizontal guide lines in the Schoeller diagram. "full" shows lines with labels, "simple" shows only dashed lines, and "none" hides them completely. Default is "full". Major ticks (0.01, 0.1, 1, 10) are emphasized. Labels appear to the left for Na and HCO3, to the right for K and CO3 (if present).
group	Optional character string. Name of the column used for coloring groups. Default is NULL.
group_custom	Logical. If TRUE, enables manual aesthetics for each group defined in group. Requires a vector color of the same length as number of groups. Default is FALSE.

base_size	Numeric. Base font size for the plot. Default is 12.
bold_label	Logical. If TRUE, displays axis labels in bold. Default is FALSE.
line_color	Character or vector. Line color(s). If group_custom = TRUE, must be a vector. Default is "black".
line_width	Numeric. Line width. Default is 1.
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object representing the Schoeller diagram.

Examples

```
data("hc_data", package = "hydrochem")
hc_data <- hc_data[1:20, ] # keep first 20 rows

# Basic Schoeller plot
plot_schoeller(hc_data)

# Remove legend
plot_schoeller(hc_data, legend.position = "none")

# Schoeller plot with mg/L guides hidden
plot_schoeller(hc_data, mg_guide = "none")

# Specify a group by unique id
plot_schoeller(hc_data, group = "id")

# Schoeller plot with group coloring
plot_schoeller(hc_data, group = "type")

# Schoeller plot with custom group colors
plot_schoeller(
  hc_data,
  group = "type",
  group_custom = TRUE,
  line_color = c("red", "blue")
)

# Schoeller plot with NO3 included
plot_schoeller(hc_data, include_NO3 = TRUE)
```

plot_stabler *Plot Stabler Diagrams from Hydrochemical Data*

Description

Generates Stabler diagrams from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_stabler(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  include_NO3 = FALSE,
  ratio = 10,
  idcol = NULL,
  nrow = NULL,
  ncol = NULL,
  facet_dir = c("v", "h"),
  xlab = "%meq/L",
  base_size = 12,
  border_color = "transparent",
  border_width = 0.5,
  axis_title_size = 14,
  label_border_color = "transparent",
  plot_title = NULL,
  legend.position = "bottom",
  legend.title = NULL,
  show_label = TRUE,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
include_NO3	Logical. If TRUE, nitrate (NO3) is included. The NO3 column must be present in the input data. Default is FALSE.
ratio	Numeric. Aspect ratio between x and y axes in the plot. Default is 10.
idcol	Character or NULL. Column name used to identify and facet individual diagrams. If NULL, synthetic sample IDs are generated. Default is NULL.
nrow, ncol	Integers. Number of rows and columns for the facet layout. Default is NULL (automatic layout).

facet_dir	Character. Facet layout direction. Either "v" for vertical or "h" for horizontal. Default is "v".
xlab	Character. Label for the x-axis. Set NULL for no axis Default is "%meq/L".
base_size	Numeric. Base font size for plot text. Default is 12.
border_color	Color of the bar border. Default is "transparent".
border_width	Width of the bar border. Default is 0.5.
axis_title_size	Numeric. Font size for strip labels. Default is 14.
label_border_color	Character. Border color for facet strip labels. Use "transparent" for no border. Default is "transparent".
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
show_label	Logical. If TRUE, displays strip labels. Default is TRUE.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object representing the Stabler diagram(s).

Examples

```
data("hc_data", package = "hydrochem")
hc_data <- hc_data[1:20, ]
plot_stabler(hc_data, idcol = "id")
```

---

plot_stiff *Plot Stiff Diagrams from Hydrochemical Data*

Description

Generates Stiff diagrams from hydrochemical data. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation. Optionally groups data points and allows manual customization of plotting aesthetics.

Usage

```
plot_stiff(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  include_NO3 = NULL,
  idcol = NULL,
  group = NULL,
  group_custom = FALSE,
  nrow = NULL,
  ncol = NULL,
  scales = c("fixed", "free_x"),
  facet_dir = c("v", "h"),
  xlab = "meq/L",
  base_size = 12,
  bold_label = FALSE,
  color = "black",
  fill = "grey40",
  axis_title_size = 14,
  label_border_color = "transparent",
  plot_title = NULL,
  legend.position = "bottom",
  legend.title = NULL,
  show_label = TRUE,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
include_NO3	Character or NULL. If set to "Cl" or "SO4", nitrate (NO3) is included in the Stiff diagram by being added to the specified anion. Default is NULL.
idcol	Character or NULL. Column name used to identify and facet individual diagrams. If not provided, a sequential ID is created. If provided, it is converted to a factor with levels ordered by first appearance in the data, ensuring facet order matches data order. Default is NULL.
group	Optional character string. Name of the column used for coloring or filling groups. Default is NULL.
group_custom	Logical. If TRUE, enables manual aesthetics for each group defined in group. Requires vectors color, fill (each of length equal to the number of groups). Default is FALSE.
nrow, ncol	Integers. Number of rows and columns for the facet layout. Default is NULL (automatic layout).
scales	Character. Facet scaling behavior. Must be "fixed" or "free_x". Default is "fixed".

facet_dir	Character. Facet layout direction. Either "v" for vertical or "h" for horizontal. Default is "v".
xlab	Character. Label for the x-axis. Set NULL for no label. Default is "meq/L".
base_size	Numeric. Base font size for the plot. Default is 12.
bold_label	Logical. If TRUE, displays y-axis group labels in bold. Default is FALSE.
color	Character or vector. Outline color(s) for polygons. If group_custom = TRUE, must be a vector. Default is "black".
fill	Character or vector. Fill color(s) for polygons. If group_custom = TRUE, must be a vector. Default is "grey40".
axis_title_size	Numeric. Font size for facet strip labels. Default is 14.
label_border_color	Character. Border color of strip labels. Use "transparent" for no border. Default is "transparent".
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
legend.position	Position of the legend. Accepts "top", "bottom", "left", "right", "none", or a numeric vector of length 2 for relative coordinates, e.g., c(1, 0.5). Default is "bottom".
legend.title	A character string for the title of the legend. If NULL, no legend title is displayed. Default is NULL.
show_label	Logical. If TRUE, displays strip labels in each facet. Default is TRUE.
bg_color	Character. Background color of the plot area and panel. Accepts valid color names or hex codes (e.g., "red", "#FFFFFF"). Default is "#FFFFFF".

Value

A ggplot object representing the Stiff diagram(s).

Examples

```
data("hc_data", package = "hydrochem")
hc_data <- hc_data[1:20, ] # keep first 20 rows

# Basic Stiff plot
plot_stiff(hc_data)

# Stiff plot with sample IDs
plot_stiff(hc_data, idcol = "id")

# Stiff plot for selected samples only
sel_data <- hc_data[hc_data$id %in% c("id_1", "id_5", "id_20"), ]
plot_stiff(sel_data)
plot_stiff(sel_data, group = "type") # with grouping

# Stiff plot with controlled facet order
sel_data <- hc_data[hc_data$id %in% c("id_1", "id_5", "id_20"), ]
```

```
sel_data <- sel_data[order(sel_data$type, decreasing = TRUE), ]
plot_stiff(sel_data, idcol = "id", group = "type")

# Stiff plot with group and custom fill
plot_stiff(
  hc_data,
  idcol = "id",
  group = "type",
  group_custom = TRUE,
  fill = c("lightblue", "orange"),
  color = c("red", "red")
)
```

---

plot_ternary *Plot Custom Ternary Diagram from Hydrochemical Data*

Description

Generates a ternary diagram using ggplot2 based on three user-specified solutes (e.g., Ca, Mg, Na). Converts input data from mg/L, mmol/L, or meq/L to normalized percentages, and displays the data in a ternary plot. No constraint is applied regarding ion type (e.g., cation/anion), allowing full flexibility for exploratory plotting.

Usage

```
plot_ternary(
  data,
  first,
  second,
  third,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  to_unit = c("mmol/L", "mg/L"),
  group = NULL,
  group_custom = FALSE,
  first_label = NULL,
  second_label = NULL,
  third_label = NULL,
  grid_lines = TRUE,
  axis_title_size = 3.5,
  axis_tick_label_size = 3,
  bold_label = FALSE,
  shape = 21,
  color = "black",
  fill = "grey40",
  size = 2,
  plot_title = NULL,
  label = NULL,
```

```
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0,
  legend.position = "bottom",
  legend.title = NULL,
  bg_color = "#FFFFFF"
)
```

Arguments

data	A data.frame or data.table containing the solute concentration data.
first, second, third	Character strings. Names of the three solute columns to be plotted on the ternary diagram.
base_unit	Character. Unit of the input concentrations. Must be one of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
to_unit	Character. Unit to which the data should be converted before normalization. Must be one of "mg/L", "mmol/L", or "meq/L". Default is "mmol/L".
group	Optional character string. Name of the grouping column in data. Default is NULL.
group_custom	Logical. If TRUE, enables manual aesthetics for each group defined in group. In this case, at least one of color, fill, shape, or size must be provided as a vector of length equal to the number of groups. Any missing or incorrectly sized vectors will be automatically filled with default values (e.g., "black", "transparent", 21, 2). If FALSE (default), all these aesthetics must be of length 1.
first_label, second_label, third_label	Optional character strings. Custom axis labels for the first, second, and third solutes. If NULL, the original column names are used.
grid_lines	Logical. Whether to include ternary grid lines. Default is TRUE.
axis_title_size	Numeric. Size of the axis/group labels. Default is 3.5.
axis_tick_label_size	Numeric. Size of the axis tick labels. Default is 3.
bold_label	Logical. Whether to render axis labels in bold. Default is FALSE.
shape	Integer or vector of integers. Shape(s) for plotting points. Must follow ggplot2 shape standards. Default is 21.
color	Character. Outline color for the plotted points. Default is "black".
fill	Character. Fill color for the plotted points. Default is "grey40".
size	Numeric. Size of the plotted points. Default is 2.
plot_title	Optional character string. Title of the plot. Default is NULL (no title displayed).
label	Optional column name in data used to label points. Default is NULL (no labels).
label_size	Numeric. Size of the label text. Default is 3.
label_nudge_x	Numeric. Horizontal offset of labels relative to points. Default is 0.

label_nudge_y	Numeric. Vertical offset of labels relative to points. Default is 0.
legend.position	Character or numeric vector. Position of the legend. Default is "bottom".
legend.title	Optional character. Title of the legend. Default is NULL.
bg_color	Character. Background color of the plot. Accepts named colors or hex codes. Default is "#FFFFFF".

Details

If transformations are needed (e.g., amplifying signal, applying powers, or any other custom scaling), users should perform these operations beforehand. The column name must still match the same name as the ion component (e.g., Mg). Users may customize axis labels via dedicated function arguments (first_label, second_label, third_label).

Value

A ggplot object representing the ternary diagram.

Examples

```
data("hc_data", package = "hydrochem")

# Basic ternary plot
plot_ternary(hc_data, first = "Ca", second = "Mg", third = "Na")

# Basic ternary plot without grouping
plot_ternary(hc_data, first = "Ca", second = "Mg", third = "Na")

# Ternary plot with simple grouping
plot_ternary(hc_data, first = "Ca", second = "Mg", third = "Na", group = "type")

# Ternary plot with custom aesthetics per group
plot_ternary(
  hc_data, first = "Ca", second = "Mg", third = "Na",
  group = "type",
  group_custom = TRUE,
  fill = c("steelblue", "tomato", "darkgreen"),
  color = c("black", "black", "black"),
  shape = c(21, 22, 24),
  size  = c(2.5, 3, 3.5)
)

# Custom transformation: amplify Mg signal by a factor of 10 (must still be named "Mg")
hc_mod <- hc_data
hc_mod$Mg <- hc_mod$Mg * 10
plot_ternary(hc_mod, first = "Ca", second = "Mg", third = "Na",
             first_label = "Ca", second_label = "10*Mg", third_label = "Na")
```

replace_bdl *Replace Values Below Detection Limit (BDL)*

Description

Detects and replaces values below detection limits (e.g., "<0.01", "< 0.005", "< .5", " < 0.1 ") in a hydrochemical dataset. Only columns including at least one BDL value are processed. Values are replaced with a numeric approximation based on the specified method.

Usage

```
replace_bdl(
  data,
  cols = NULL,
  method = c("DL/2", "DL/sqrt(2)", "DL", "0", "NA")
)
```

Arguments

data	A data.frame or data.table.
cols	A character vector of cols names to summarize. If NULL, columns are detected automatically.
method	Character. Method used to replace values below detection limits. One of: • "DL/2" (default): replaces with half the detection limit, • "DL/sqrt(2)": replaces with DL divided by $\sqrt{x}2$, • "DL": replaces with the detection limit itself, • "0": replaces with zero. • "NA": replaces with NA.

Value

A data.table where BDL character values have been replaced by numeric values according to the selected method.

Examples

```
df <- data.frame(Ca = c(" <0.01", "5.2", "< 0.005"), Cl = c("3.1", "<0.02", "2.5"))
df <- replace_bdl(df, method = "DL/2")
print(df)

data <- replace_bdl(data_qc, method = "0")
print(data)
```

rsc *Calculate Residual Sodium Carbonate (RSC)*

Description

Computes the Residual Sodium Carbonate (RSC) index from hydrochemical data using calcium, magnesium, bicarbonate, and carbonate concentrations. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"`, and are internally converted to `"meq/L"` before calculation.

Usage

```
rsc(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

`data`	A `data.frame` or `data.table` containing the required ion columns `"HCO3"`, `"Ca"`, and `"Mg"`.
`base_unit`	Unit of the input ion concentrations. One of `"mg/L"`, `"mmol/L"`, or `"meq/L"`. Default is `"mg/L"`.

Details

The function checks that all required ions (`CO3`, `HCO3`, `Ca`, `Mg`) are present in the input dataset and throws an error if any are missing. If carbonate (`CO3`) is also provided, its contribution is added to bicarbonate after conversion to milliequivalents (`meq/L`). If `CO3` is absent, alkalinity is considered to be represented solely by bicarbonate (`HCO3`).

Value

A numeric vector of Residual Sodium Carbonate values, one per row of `data`.

Examples

```
data("hc_data", package = "hydrochem")
rsc(hc_data, base_unit = "mg/L")
```

sar *Calculate Sodium Adsorption Ratio (SAR)*

Description

Computes the Sodium Adsorption Ratio (SAR) from hydrochemical data using sodium, calcium, and magnesium concentrations. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"`, and are internally converted to `"meq/L"` before calculation.

Usage

```
sar(data, base_unit = c("mg/L", "mmol/L", "meq/L"))
```

Arguments

data	A data.frame or data.table containing the required ion columns "Na", "Ca", and "Mg".
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".

Value

A numeric vector of Sodium Adsorption Ratio values, one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
sar(hc_data, base_unit = "mg/L")
```

---

save_plot *Save a plot Diagram to File*

Description

Saves a plot diagram (or any ggplot object) to a file using ggsave().

Usage

```
save_plot(
  filename,
  type = NULL,
  plot = last_plot(),
  dpi = 300,
  width = NULL,
  height = NULL,
  ...
)
```

Arguments

filename	Character. Path to the output file (e.g., "piper_plot.png").
type	Character. Diagram type used to choose default width/height if not supplied. One of c("piper", "ternary", "collins", "stabler", "durov", "gibbs", schoeller", "stiff
plot	A ggplot object to save. Default is the last plot (last_plot()).
dpi	Resolution of the output image in dots per inch. Default is 300.
width, height	Numeric. Output size. If NULL, defaults depend on type.
...	Additional arguments passed to ggsave().

Value

Invisibly returns the filename.

Examples

```
p <- plot_piper(hc_data)

  save_plot(file.path(tempdir(), "piper.png"), type = "piper", plot = p)


p <- plot_stiff(hc_data[1:10, ], ncol = 5)

  if (interactive()) {
   save_plot(file.path(tempdir(), "stiff.png"), type = "stiff", plot = p, width = 10, height = 4)
  }
```

---

stat_summary *Summary Statistics*

Description

Computes summary statistics for selected numeric columns in a dataset, including: mean, standard deviation, minimum, maximum, median, percentiles (1%, 5%, 25%, 75%, 95%, 99%), geometric mean, mode, median absolut deviation (MAD), coefficient of variation (CV), range, interquartile range (IQR), number of non-missing values (n), percentage of non-missing values (n_percent), number of NAs (NA), and percentage of NAs (NA_percent).

Usage

```
stat_summary(data, params = "all", group = NULL, stats = NULL, na.rm = FALSE)
```

Arguments

data	A data.frame or data.table with numeric columns.
params	A character vector of cols names to summarize. If params = "all", all numerical columns are used.
group	Optional character string. Name of the grouping column in data. Default is NULL.
stats	Character vector of statistics to select (e.g., c("min", "max", "mean")). Must be among: c("min", "P1", "P5", "P10", "P25", "med", "P75", "P90", "P95", "P99", "max", "mean", "gmean", "mode", "range", "IQR", "sd", "mad", "CV", "n", "n_percent", "NA", "NA_percent"). If NULL, all available statistics are returned.
na.rm	Logical. Should missing values be removed before computing statistics? Default is FALSE.

Value

A data.table with summarized statistics for the selected columns.

Examples

```
data("hc_data", package = "hydrochem")
stat_summary(hc_data, params = c("Ca", "Mg", "Na"))
stat_summary(hc_data, group = "type", params = c("Ca", "Mg", "Na"))
```

---

tds *Calculate Total Dissolved Solids (TDS)*

Description

Computes the total dissolved solids (TDS) of a hydrochemical dataset by summing the concentrations of selected ions in mg/L. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "mg/L" prior to calculation.

Usage

```
tds(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  method = c("major", "major_si", "all_ions")
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
method	Character string indicating the calculation method. One of: • "major" (default): uses major ions (Ca, Mg, Na, K, HCO3, CO3, Cl, SO4, NO3), • "major_si": adds SiO2 to the major ions (can be calculated with Si if only available), • "all_ions": uses all available ions in the dataset and SiO2.

Value

A numeric vector of TDS values, one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
tds(hc_data, base_unit = "mg/L")
```

`total_alkalinity` *Calculate Total Alkalinity*

Description

Computes total alkalinity from hydrochemical data using bicarbonate/carbonate concentrations and pH. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"` and are internally converted to `"meq/L"` prior to calculation.

Usage

```
total_alkalinity(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  output = c("mg/L_CaCO3", "meq/L")
)
```

Arguments

`data`	A `data.frame` or `data.table` containing the required columns `"HCO3"` and `"pH"`.
`base_unit`	Unit of the input ion concentrations. One of `"mg/L"`, `"mmol/L"`, or `"meq/L"`. Default is `"mg/L"`.
`output`	Unit of the returned hardness. One of: • `"mg/L_CaCO3"` — milligrams per liter as CaCO3 (default) • `"meq/L"` — milliequivalents per liter

Details

The function checks that both required variables (`HCO3`, `pH`) are present in the input dataset and throws an error if either are missing. If carbonate (`CO3`) is also provided, its contribution is added to bicarbonate after conversion to milliequivalents (`meq/L`). If `CO3` is absent, alkalinity is considered to be represented solely by bicarbonate (`HCO3`).

Value

A numeric vector of Total Alkalinity values (in mg/L as CaCO3), one per row of `data`.

Examples

```
data("data_qc", package = "hydrochem")
total_alkalinity(data_qc, base_unit = "mg/L")
```

`total_hardness` *Calculate Total Hardness*

Description

Computes total water hardness from hydrochemical data using calcium and magnesium concentrations. Input concentrations may be provided in `"mg/L"`, `"mmol/L"`, or `"meq/L"` and are internally converted to `"mg/L"` prior to calculation.

Usage

```
total_hardness(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  output = c("mg/L_CaCO3", "fH", "dH", "e")
)
```

Arguments

`data`	A `data.frame` or `data.table` containing the required ion columns `"Ca"` and `"Mg"`.
`base_unit`	Unit of the input ion concentrations. One of `"mg/L"`, `"mmol/L"`, or `"meq/L"`. Default is `"mg/L"`.
`output`	Unit of the returned hardness. One of: • `"mg/L_CaCO3"` — milligrams per liter as CaCO3 (default) • `"fH"` — French degrees • `"dH"` — German degrees • `"e"` — English degrees

Details

The function checks that both required ions (`Ca`, `Mg`) are present in the input dataset and throws an error if either is missing.

Value

A numeric vector of Total Hardness values (in mg/L as CaCO3), one per row of `data`.

Examples

```
data("hc_data", package = "hydrochem")
total_hardness(hc_data, base_unit = "mg/L")
```

water_type *Determine Water Type*

Description

Classifies each sample in a hydrochemical dataset by water type based on the dominant cations and anions. Input concentrations may be provided in "mg/L", "mmol/L", or "meq/L" and are internally converted to "meq/L" prior to calculation.

Usage

```
water_type(
  data,
  base_unit = c("mg/L", "mmol/L", "meq/L"),
  format = c("long", "short"),
  threshold = 20
)
```

Arguments

data	A data.frame or data.table with columns named after ions (e.g., "Ca", "Cl", "SO4").
base_unit	Unit of the input ion concentrations. One of "mg/L", "mmol/L", or "meq/L". Default is "mg/L".
format	Output format: "short" for dominant ions only, "long" to include all ions above a threshold. Default is "long".
threshold	Numeric. Minimum percentage for an ion to be considered dominant in "long" format. Default is 20.

Details

Two formats are available:

- "short": returns the single most dominant cation and anion (e.g., "Ca-HCO3").
- "long": returns all ions above a specified threshold, joined by "+" (e.g., "Ca+Mg-HCO3+SO4").

Value

A character vector of water types sorted in decreasing order, one per row of data.

Examples

```
data("hc_data", package = "hydrochem")
water_type(hc_data, base_unit = "mg/L", format = "short")
water_type(hc_data, base_unit = "mg/L", format = "long", threshold = 25)
```

Index

∗ **datasets**
data_qc, 5
hc_data, 6

cai1, 3
cai2, 3
convert_unit, 4

data_qc, 5

hc_data, 6

ionic_balance, 6

kelly_ratio, 7

magnesium_hazard, 8

percent_sodium, 9
permeability_index, 9
plot_biplot, 10
plot_collins, 14
plot_durov, 16
plot_gibbs, 18
plot_piper, 20
plot_schoeller, 23
plot_stabler, 25
plot_stiff, 26
plot_ternary, 29

replace_bdl, 32
rsc, 33

sar, 33
save_plot, 34
stat_summary, 35

tds, 36
total_alkalinity, 37
total_hardness, 38

water_type, 39