Package 'biocharkitgui'

July 30, 2026

Title 'Shiny' GUI for the 'biocharkit' Biochar Analysis Toolkit

Version 0.3.0

Description A point-and-click 'Shiny' interface to the 'biocharkit' package. Lets a user upload Excel workbooks of biochar characterisation and batch adsorption data, map spreadsheet columns to the required variables via dropdown menus, and run sample-ID parsing, adsorption capacity and removal efficiency calculations, isotherm fitting (Langmuir, Freundlich, Temkin, Dubinin-Radushkevich, Sips), kinetics fitting (pseudo-first/second-order, Elovich, intraparticle diffusion), van't Hoff thermodynamics, batch fitting across many samples at once, FTIR baseline correction, automatic peak picking and functional-group analysis, XRD peak deconvolution and crystallinity index, BET surface area, TGA analysis (DTG curve with auto-detected decomposition peaks, moisture/volatile-matter/ash/fixed-carbon straight off a curve for a single sample or in batch across many, and Kissinger non-isothermal kinetics from multi-heating-rate data), proximate/ultimate analysis, and correlation matrices, without writing any R code. Results and 600 dpi TIFF figures can be downloaded directly from the browser, along with a combined analysis report.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 4.0)

Imports biocharkit (>= 0.3.0), shiny, readxl, DT, rmarkdown, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Sukamal Sarkar [aut, cre]

Maintainer Sukamal Sarkar <sukamal.sarkar@gm.rkmvu.ac.in>

Repository CRAN

Date/Publication 2026-07-30 16:30:14 UTC

Contents

`gui_adsorption` *Compute adsorption capacity and removal efficiency from an uploaded data frame*

Description

Compute adsorption capacity and removal efficiency from an uploaded data frame

Usage

```
gui_adsorption(df, c0_col, ce_col, v_col, m_col, id_col = NULL)
```

Arguments

`df`	Data frame.
`c0_col`, `ce_col`, `v_col`, `m_col`	Column names for initial concentration (mg/L), equilibrium concentration (mg/L), volume (L), and adsorbent mass (g) respectively.
`id_col`	Optional column name to carry through as a sample identifier in the output (not used in the calculation).

Value

A data frame with the original identifying column (if given), `qe_mgg` and `removal_pct`.

`gui_bet` *BET surface area from an uploaded data frame*

Description

BET surface area from an uploaded data frame

Usage

```
gui_bet(df, pp0_col, q_col)
```

Arguments

`df`	Data frame.
`pp0_col`	Column name for relative pressure (P/P0).
`q_col`	Column name for quantity adsorbed (cm^3/g STP).

Value

The list returned by `biocharkit::bet_surface_area()`.

gui_correlation *Correlation matrix for display, in long format*

Description

Correlation matrix for display, in long format

Usage

```
gui_correlation(df, cols, method = c("spearman", "pearson", "kendall"))
```

Arguments

df	Data frame.
cols	Character vector of numeric column names to correlate.
method	"spearman", "pearson", or "kendall".

Value

A list with wide_estimate (matrix), long (data frame with var1, var2, rho, p_value, one row per unique pair, for easy table display/download).

gui_datatable *Render a data frame as an interactive DT table*

Description

Thin wrapper around DT::datatable() with sensible defaults for this app (paginated, 10 rows per page).

Usage

```
gui_datatable(df, page_length = 10)
```

Arguments

df	Data frame to display.
page_length	Rows per page. Default 10.

Value

A DT::datatable htmlwidget.

gui_ftir_autopeaks *Automatic FTIR peak picking from an uploaded full spectrum*

Description

Automatic FTIR peak picking from an uploaded full spectrum

Usage

```
gui_ftir_autopeaks(df, wavenumber_col, intensity_col, min_prominence = 0)
```

Arguments

`df`	Data frame.
`wavenumber_col, intensity_col`	Column names.
`min_prominence`	Minimum prominence for a peak to be kept.

Value

A data frame from `biocharkit::find_ftir_peaks()`.

gui_ftir_baseline *Baseline-correct an FTIR spectrum from an uploaded data frame*

Description

Baseline-correct an FTIR spectrum from an uploaded data frame

Usage

```
gui_ftir_baseline(
  df,
  wavenumber_col,
  intensity_col,
  method = c("linear", "rolling_min")
)
```

Arguments

`df`	Data frame.
`wavenumber_col, intensity_col`	Column names.
`method`	`"linear"` or `"rolling_min"`.

Value

A two-column data frame (`wavenumber_cm1`, `intensity`), the baseline-corrected spectrum.

gui_ftir_density *Functional-group density from an uploaded full FTIR spectrum*

Description

Functional-group density from an uploaded full FTIR spectrum

Usage

```
gui_ftir_density(df, wavenumber_col, intensity_col)
```

Arguments

df	Data frame with a wavenumber column and an intensity column.
wavenumber_col, intensity_col	Column names.

Value

A list with `table` (from `biocharkit::functional_group_density()`) and `spectrum` (a two-column data frame usable with `biocharkit::plot_ftir_spectrum()`).

gui_ftir_peaks *Assign FTIR peak positions to functional groups*

Description

Assign FTIR peak positions to functional groups

Usage

```
gui_ftir_peaks(df, peak_col)
```

Arguments

df	Data frame with a column of peak wavenumbers.
peak_col	Column name.

Value

A data frame from `biocharkit::assign_ftir_peaks()`.

gui_isotherm *Fit an adsorption isotherm from an uploaded data frame*

Description

Fit an adsorption isotherm from an uploaded data frame

Usage

```
gui_isotherm(
  df,
  ce_col,
  qe_col,
  model = c("langmuir", "freundlich", "temkin", "dr", "sips")
)
```

Arguments

df	Data frame.
ce_col, qe_col	Column names for equilibrium concentration (mg/L) and adsorption capacity (mg/g).
model	"langmuir", "freundlich", "temkin", "dr", or "sips".

Value

A list with elements fit, params (a one-row data frame of fitted parameters for display), Ce, qe, and model.

gui_isotherm_batch *Fit an adsorption isotherm separately per group in an uploaded data frame*

Description

Fit an adsorption isotherm separately per group in an uploaded data frame

Usage

```
gui_isotherm_batch(
  df,
  group_col,
  ce_col,
  qe_col,
  model = c("langmuir", "freundlich", "temkin", "dr", "sips")
)
```

Arguments

df	Data frame.
group_col	Grouping column (e.g. sample ID).
ce_col, qe_col	Column names as in gui_isotherm().
model	One of "langmuir", "freundlich", "temkin", "dr", "sips".

Value

A data frame with one row per group (see biocharkit::fit_isotherm_batch()).

---

gui_kinetics *Fit adsorption kinetics from an uploaded data frame*

Description

Fit adsorption kinetics from an uploaded data frame

Usage

```
gui_kinetics(
  df,
  t_col,
  qt_col,
  model = c("pfo", "pso", "elovich", "intraparticle")
)
```

Arguments

df	Data frame.
t_col, qt_col	Column names for contact time and adsorption capacity at time t (mg/g).
model	"pfo", "pso", "elovich", or "intraparticle".

Value

A list with elements fit, params, t, qt, model.

gui_kinetics_batch *Fit adsorption kinetics separately per group in an uploaded data frame*

Description

Fit adsorption kinetics separately per group in an uploaded data frame

Usage

```
gui_kinetics_batch(
  df,
  group_col,
  t_col,
  qt_col,
  model = c("pfo", "pso", "elovich", "intraparticle")
)
```

Arguments

`df`	Data frame.
`group_col`	Grouping column (e.g. sample ID).
`t_col`, `qt_col`	Column names as in `gui_kinetics()`.
`model`	One of `"pfo"`, `"pso"`, `"elovich"`, `"intraparticle"`.

Value

A data frame with one row per group (see `biocharkit::fit_kinetics_batch()`).

gui_list_sheets *List sheet names in an uploaded Excel workbook*

Description

List sheet names in an uploaded Excel workbook

Usage

```
gui_list_sheets(path)
```

Arguments

`path`	Path to an `.xlsx`/`.xls` file.

Value

Character vector of sheet names.

gui_parse_ids *Parse sample IDs from an uploaded data frame*

Description

Parse sample IDs from an uploaded data frame

Usage

```
gui_parse_ids(df, id_col)
```

Arguments

df	Data frame (from gui_read_excel()).
id_col	Name of the column containing sample IDs.

Value

A data frame from biocharkit::parse_sbc_id().

gui_preview *Preview the first rows of an uploaded data frame*

Description

Thin wrapper around utils::head() used by the Shiny app's data preview.

Usage

```
gui_preview(df, n = 5)
```

Arguments

df	Data frame.
n	Number of rows to show. Default 5.

Value

A data frame: the first n rows of df.

gui_proximate *Proximate analysis from an uploaded data frame*

Description

Proximate analysis from an uploaded data frame

Usage

```
gui_proximate(df, moisture_col, vm_col, ash_col, id_col = NULL)
```

Arguments

df	Data frame.
moisture_col, vm_col, ash_col	Column names for moisture %, volatile matter %, and ash % (as-received basis).
id_col	Optional identifier column to carry through.

Value

A data frame from biocharkit::proximate_analysis().

gui_read_excel *Read a sheet of an uploaded Excel workbook into a data frame*

Description

Thin, defensive wrapper around readxl::read_excel() that returns plain data.frames (not tibbles) and gives an informative error rather than a cryptic one if the file or sheet can't be read.

Usage

```
gui_read_excel(path, sheet = 1)
```

Arguments

path	Path to an .xlsx/.xls file.
sheet	Sheet name or 1-based index. Defaults to the first sheet.

Value

A data.frame.

gui_render_report *Render the combined session report*

Description

Renders `report_template.Rmd` (shipped in `inst/shiny-app/report/`) against a snapshot of accumulated analysis results, producing a self-contained HTML report.

Usage

```
gui_render_report(store, output_file)
```

Arguments

`store`	A named list of accumulated results (see the Shiny app's server code for the expected structure); sections for `NULL` entries are omitted from the report.
`output_file`	Destination `.html` file path.

Value

Invisibly, `output_file`.

gui_snapshot_png *Snapshot a base-graphics plotting expression to a PNG file*

Description

Opens a PNG device, evaluates `expr` (a plotting call such as `biocharkit::plot_isotherm()`), closes the device, and returns the file path. Used to capture plots for inclusion in the combined session report.

Usage

```
gui_snapshot_png(expr, width = 900, height = 650, res = 130)
```

Arguments

`expr`	A plotting expression/call to evaluate with the device open.
`width, height`	Pixel dimensions. Defaults `900`, `650`.
`res`	Resolution (dpi). Default `130`.

Value

The path to the written PNG file (a temp file).

gui_tga_curve — *Build a canonical TGA curve data frame from uploaded columns*

Description

Build a canonical TGA curve data frame from uploaded columns

Usage

```
gui_tga_curve(df, temperature_col, weight_col)
```

Arguments

`df`	Data frame.
`temperature_col, weight_col`	Column names for temperature (degrees C) and weight percent.

Value

A data frame with columns `temperature_C`, `weight_pct`.

gui_tga_kissinger — *Kissinger non-isothermal kinetics from an uploaded multi-heating-rate table*

Description

Kissinger non-isothermal kinetics from an uploaded multi-heating-rate table

Usage

```
gui_tga_kissinger(df, beta_col, tpeak_col)
```

Arguments

`df`	Data frame with one row per heating-rate run (not a raw TGA curve): typically produced by running the TGA curve tab once per heating rate and noting the DTG peak temperature each time.
`beta_col`	Column name for heating rate (degrees C/min).
`tpeak_col`	Column name for the DTG peak temperature (degrees C) at that heating rate, for the same decomposition stage across rows.

Value

A list with elements `fit` (from `biocharkit::tga_kinetics_kissinger()`) and `params` (a one-row data frame for display).

gui_tga_stages — *Proximate analysis directly from an uploaded TGA curve*

Description

Proximate analysis directly from an uploaded TGA curve

Usage

```
gui_tga_stages(
  df,
  temperature_col,
  weight_col,
  moisture_end = 110,
  vm_end = 650
)
```

Arguments

`df`	Data frame.
`temperature_col, weight_col`	Column names as in `gui_tga_curve()`.
`moisture_end, vm_end`	Temperature breakpoints (degrees C); see `biocharkit::tga_stages()`.

Value

A data frame from `biocharkit::tga_stages()`.

gui_tga_stages_batch — *Proximate analysis from TGA curves, separately per sample in an uploaded long-format data frame*

Description

Proximate analysis from TGA curves, separately per sample in an uploaded long-format data frame

Usage

```
gui_tga_stages_batch(
  df,
  group_col,
  temperature_col,
  weight_col,
  moisture_end = 110,
  vm_end = 650
)
```

Arguments

df	Data frame containing multiple samples' curves stacked long-format.
group_col	Grouping column (e.g. sample ID).
temperature_col, weight_col	Column names as in gui_tga_curve().
moisture_end, vm_end	As in gui_tga_stages().

Value

A data frame from biocharkit::tga_stages_batch().

---

gui_ultimate *Ultimate (CHNS) analysis atomic ratios from an uploaded data frame*

Description

Ultimate (CHNS) analysis atomic ratios from an uploaded data frame

Usage

```
gui_ultimate(df, c_col, h_col, o_col, n_col = NULL, id_col = NULL)
```

Arguments

df	Data frame.
c_col, h_col, o_col	Column names for C%, H%, O% (mass basis).
n_col	Optional column name for N%.
id_col	Optional identifier column to carry through.

Value

A data frame from biocharkit::ultimate_ratios().

gui_vant_hoff *Van't Hoff thermodynamic analysis from an uploaded data frame*

Description

Van't Hoff thermodynamic analysis from an uploaded data frame

Usage

```
gui_vant_hoff(df, temp_col, kc_col)
```

Arguments

df	Data frame.
temp_col	Column name for absolute temperature (Kelvin).
kc_col	Column name for the equilibrium distribution coefficient (dimensionless, e.g. qe/Ce) at each temperature.

Value

The list returned by `biocharkit::fit_vant_hoff()`.

gui_write_csv *Write a results data frame to CSV (no row names)*

Description

Thin wrapper around `utils::write.csv()` used by the Shiny app's download handlers.

Usage

```
gui_write_csv(x, file)
```

Arguments

x	Data frame to write.
file	Destination file path.

Value

Invisibly, `NULL`. Called for its side effect of writing `file`.

gui_xrd_ci *XRD crystallinity index for one or more samples*

Description

XRD crystallinity index for one or more samples

Usage

```
gui_xrd_ci(df, crystalline_col, amorphous_col, id_col = NULL)
```

Arguments

df	Data frame.
crystalline_col, amorphous_col	Column names for crystalline and amorphous integrated peak areas.
id_col	Optional identifier column to carry through.

Value

A data frame with crystallinity index per row.

gui_xrd_deconvolve *XRD peak deconvolution from an uploaded data frame*

Description

XRD peak deconvolution from an uploaded data frame

Usage

```
gui_xrd_deconvolve(df, two_theta_col, intensity_col, peak_centers)
```

Arguments

df	Data frame.
two_theta_col, intensity_col	Column names for 2theta and intensity.
peak_centers	Numeric vector of approximate crystalline peak positions (2theta).

Value

The list returned by `biocharkit::xrd_deconvolve()`.

`run_biocharkit_gui` *Launch the biocharkit Shiny GUI*

Description

Starts a local Shiny app in your default web browser. Upload Excel (`.xlsx`/`.xls`) workbooks, map spreadsheet columns to the required variables from dropdown menus, and run biochar characterisation and adsorption analyses (sample ID parsing, adsorption capacity, isotherms, kinetics, FTIR, XRD crystallinity index, correlation matrices) without writing any R code.

Usage

```
run_biocharkit_gui(...)
```

Arguments

`...`	Additional arguments passed to `shiny::runApp()`, e.g. `launch.browser`, `port`.

Value

Called for its side effect of launching the Shiny app; does not return under normal use (control passes to the Shiny event loop until the app is closed).

Examples

```
if (interactive()) {
  run_biocharkit_gui()
}
```

Index