
Independent Research Project

Release 1.0.0

Dongzi Ding

Aug 14, 2023

CONTENTS:

1	src package	1
1.1	Subpackages	1
1.2	mainwindow module	10
2	Indices and tables	13
	Python Module Index	15
	Index	17

SRC PACKAGE

1.1 Subpackages

1.1.1 gui package

button_area module

button_area.py

Author: Dongzi Ding Created: 2023-06-25 Modified: 2023-08-14

class src.gui.button_area.**ButtonArea**(parent=None)

Bases: QWidget

Main button area of the application which provides the necessary buttons for the user to interact with the application.

Attributes:

- result: A dictionary storing the results.
- figures: A dictionary storing the generated figures.
- main_window: Reference to the main application window.

calculate()

Calculate functionality of the application.

reset()

Reset functionality of the application.

save_result()

Save result functionality of the application.

show_result()

Show result functionality of the application.

show_visual()

Show visual functionality of the application.

update_start_button()

class src.gui.button_area.**OptionDialog**(selected_features, parent=None)

Bases: QDialog

A dialog for selecting analysis options.

get_options(feature)

`use_both()`

`use_sklearn()`

help_window module

help_window.py

Author: Dongzi Ding Created: 2023-07-06 Modified: 2023-08-14

class `src.gui.help_window.HelpWindow`(*parent=None*)

Bases: `QMainWindow`

A `QMainWindow` class that represents the help window.

Attributes:

parent (`QWidget`)

The parent widget of the help window.

open_pdf_manual()

Opens the PDF manual located in the assets directory. The method of opening depends on the OS.

input_window module

input_window.py

Author: Dongzi Ding Created: 2023-06-27 Modified: 2023-08-14

class `src.gui.input_window.InputWindow`(*parent=None*)

Bases: `QWidget`

A `QWidget` class that represents the input window for user data.

Attributes:

- **input_changed** (`pyqtSignal`): Signal emitted when the input changes.

browse_file()

Handles the file browsing and data extraction for the selected features.

clear_data()

Clears the stored data.

emit_input_changed()

Emits the `input_changed` signal.

input_changed

manual_input()

Handles the manual input of data by the user.

reset()

Resets the input window to its default state.

update_content()

Updates the input method based on the user's selected option.

result_window module

result_window.py

Author: Dongzi Ding Created: 2023-06-28 Modified: 2023-08-14

class src.gui.result_window.**ResultWindow**(parent=None)

Bases: QMainWindow

A QMainWindow class that represents the result window for displaying analysis results.

Attributes:

- tab_widget (QTabWidget): Widget to manage multiple result tabs.

add_result(title, result, feature_name)

Adds a result to a new tab based on the feature_name.

Args:

- title (str): The title for the new tab.
- result (dict or tuple): The result data.
- feature_name (str): The name of the analysis feature.

settings_window module

settings_window.py

Author: Dongzi Ding Created: 2023-06-26 Modified: 2023-08-14

class src.gui.settings_window.**SettingsWindow**(parent=None)

Bases: QWidget

A window displaying user settings and associated instructions.

Attributes:

- main_window (QWidget): Reference to the main application window.
- guide_textedit (QTextEdit): Text area displaying guidance for selected settings.
- scroll_area (QScrollArea): Scroll area housing the guide text.
- option_label (QLabel): Label showing current selected options.

update_content()

Update the content displayed based on the user's selected settings.

visual_window module

visual_window.py

Author: Dongzi Ding Created: 2023-07-01 Modified: 2023-08-14

class src.gui.visual_window.**VisualWindow**(pixmap, parent=None)

Bases: QMainWindow

A QMainWindow class to display an image in a window for visualization purposes.

Attributes:

- image_label (QLabel): A label widget to display the image.
- central_widget (QWidget): The central widget containing the image label.

1.1.2 utils package

initial_rate module

initial_rate.py

Author: Dongzi Ding Created: 2023-06-25 Modified: 2023-08-14

`src.utils.initial_rate.calculate_rate(time, conc, threshold)`

Calculates the rate of a reaction using linear regression on a subset of data.

Parameters:

- time (array): Time data.
- conc (array): Concentration data.
- threshold (float): Percentage of data to use for regression.

Returns:

Dictionary containing time, concentration, slope, intercept, and R squared values.

`src.utils.initial_rate.calculate_rate_compare(time, conc)`

Calculates rates using different thresholds and compares the fits.

Parameters:

- time (array): Time data.
- conc (array): Concentration data.

Returns:

Dictionary containing time, concentration, slopes, intercepts, and R squared values for each threshold.

`src.utils.initial_rate.cut_data(time, conc, threshold)`

Filters time and concentration data based on a threshold.

Parameters:

- time (array): Time data.
- conc (array): Concentration data.
- threshold (float): Threshold value for filtering.

Returns:

Filtered arrays of time and concentration values.

`src.utils.initial_rate.plot_initial_rate(time, conc, slope, intercept, r_squared)`

Generates a plot of the initial reaction rate.

Parameters:

- time (array): Time data.
- conc (array): Concentration data.
- slope (float): Slope from linear regression.
- intercept (float): Intercept from linear regression.
- r_squared (float): R squared value from linear regression.

Returns:

A QPixmap object containing the plot.

`src.utils.initial_rate.plot_rate_comparison(time, conc, slopes, intercepts, r_squared_values)`

Generates a plot comparing reaction rates for different thresholds.

Parameters:

- `time` (array): Time data.
- `conc` (array): Concentration data.
- `slopes` (list): List of slopes from linear regressions.
- `intercepts` (list): List of intercepts from linear regressions.
- `r_squared_values` (list): List of R squared values from linear regressions.

Returns:

A QPixmap object containing the comparison plot.

`src.utils.initial_rate.read_data(filename)`

Reads experimental data from an Excel file.

Parameters:

- `filename` (str): Path to the Excel file.

Returns:

Arrays of time and concentration values.

input_help module

input_help.py

Author: Dongzi Ding Created: 2023-07-28 Modified: 2023-08-14

class `src.utils.input_help.DataInputDialog`(*parent=None*)

Bases: `QDialog`

A custom dialog for the user to input experimental data.

Attributes:

- `main_window` (`QWidget`): Reference to the main application window.
- `input_data` (dict): Dictionary storing input data after confirmation.
- `data_types` (dict): Dictionary defining the expected data types for each tab.
- `tab_widget` (`QTabWidget`): Widget containing tabs for each analysis type.
- `list_widgets` (dict): Dictionary storing list widgets for each tab.
- `input_fields` (dict): Dictionary storing input fields for each data type and tab.
- `check_boxes` (dict): Dictionary storing checkboxes indicating readiness for each data type and tab.

confirm_input()

Validates and confirms the input data from the current tab.

get_input_data()

Converts input strings to numeric data and returns a dictionary.

select_all(*state*)

Selects all items in the current tab's `QListWidget`.

update_input_fields(*item*)

Enables or disables the input field and checkbox for a clicked item.

plane3D_plot module

plane3D_plot.py

Author: Dongzi Ding Created: 2023-08-12 Modified: 2023-08-14

This file contains functions for performing 3D plotting and regression analysis on data. It includes functions for reading data, plotting 3D scatter points, and fitting a plane to the data.

class src.utils.plane3D_plot.Plane3DPlotter(*filename=None*)

Bases: object

A class for performing 3D plotting and regression analysis on data.

Attributes:

- filename (str): The path to the data file.
- data (tuple): Data values for pH, deltapH, logFe, deltalogFe, logR, deltalogR.
- params (tuple): Regression parameters.
- r_squared (float): R squared value of regression analysis.

Methods:

- read_data: Read data from a file.
- perform_analysis: Perform regression analysis.
- fit_plane: Fit a plane to the data.
- plot_3D_data: Plot the 3D data with a fitted plane.
- create_3D_plot: Create a 3D plot.
- plot_fitted_plane: Plot a fitted plane on 3D data.
- get_results: Retrieve analysis results.
- fig_to_pixmap: Convert a Matplotlib figure to a QPixmap.

create_3D_plot(*pH, logFe, logR, ax=None*)

Creates a 3D plot for the given data.

Args:

- pH (array-like): pH values.
- logFe (array-like): logFe values.
- logR (array-like): logR values.
- ax (matplotlib.axes._subplots.Axes3DSubplot, optional): 3D subplot. Defaults to None.

Returns:

Matplotlib figure and 3D subplot.

fig_to_pixmap(*fig*)

Converts a Matplotlib figure to a QPixmap.

Args:

- fig (matplotlib.figure.Figure): Matplotlib figure.

Returns:

PyQt5.QtGui.QPixmap: Pixmap representation of the figure.

fit_plane(*pH*, *logFe*, *logR*)

Fits a plane to the data.

Args:

- *pH* (array-like): pH values.
- *logFe* (array-like): logFe values.
- *logR* (array-like): logR values.

Returns:

Regression parameters and R squared value.

get_results()

Retrieve the regression analysis results.

Returns:

dict: Regression parameters and R squared value.

perform_analysis(*pH*, *logFe*, *logR*)

Perform regression analysis on data.

Args:

- *pH* (array-like): pH values.
- *logFe* (array-like): logFe values.
- *logR* (array-like): logR values.

Returns:

Regression parameters and R squared value.

plot_3D_data(*pH*, *logFe*, *logR*, *ax=None*)

Plot the 3D data along with the fitted plane.

Args:

- *pH* (array-like): pH values.
- *logFe* (array-like): logFe values.
- *logR* (array-like): logR values.
- *ax* (matplotlib.axes._subplots.Axes3DSubplot, optional): 3D subplot. Defaults to None.

Returns:

Matplotlib figure and 3D subplot.

plot_fitted_plane(*ax*, *pH*, *logFe*, *params*)

Plots a fitted plane on the 3D data.

Args:

- *ax* (matplotlib.axes._subplots.Axes3DSubplot): 3D subplot.
- *pH* (array-like): pH values.
- *logFe* (array-like): logFe values.
- *params* (tuple): Regression parameters.

Returns:

PyQt5.QtGui.QPixmap: QPixmap representation of the plot.

read_data(*filename*)

Reads data from a file.

Args:

- *filename* (str): Path to the data file.

Returns:

Data values for pH, ΔpH , $\log\text{Fe}$, $\Delta\log\text{Fe}$, $\log R$, $\Delta\log R$ or None if an error occurs.

rate_const module

rate_const.py

Author: Dongzi Ding Created: 2023-08-10 Modified: 2023-08-14

`src.utils.rate_const.calculate_rate(time, conc)`

Calculates the reaction rate using regression on logarithmic concentration.

Args:

- time (array-like): Array of time values.
- conc (array-like): Array of concentration values.

Returns:

Calculated values including time, logarithmic concentration, slope, intercept, and R squared value.

`src.utils.rate_const.plot(time, conc, slope, intercept, r_squared)`

Plots the given time and logarithmic concentration data with a linear fit.

Args:

- time (array-like): Array of time values.
- conc (array-like): Array of logarithmic concentration values.
- slope (float): Slope from linear regression.
- intercept (float): Intercept from linear regression.
- r_squared (float): R squared value from linear regression.

Returns:

PyQt5.QtGui.QPixmap: QPixmap representation of the plot.

`src.utils.rate_const.read_data(filename)`

Reads data from the given filename.

Args:

- filename (str): Path to the data file.

Returns:

Time and concentration values or None if an error occurs.

regression_analysis module

regression_analysis.py

Author: Dongzi Ding Created: 2023-06-28 Modified: 2023-08-14

This file contains functions for performing regression analysis on data. It includes functions for reading data, calculating LINEAR regression, and plotting the regression line.

`src.utils.regression_analysis.calculate_regression(x, y, sdx_absolute=None,
sdy_absolute=None, use_sklearn=False)`

Calculates the linear regression of the data.

Args:

- x (array-like): The x data.

- `y` (array-like): The y data.
- `sdx_absolute` (array-like, optional): Absolute standard deviations of the x data. Defaults to None.
- `sd_y_absolute` (array-like, optional): Absolute standard deviations of the y data. Defaults to None.
- `use_sklearn` (bool, optional): Whether to use sklearn for the regression. Defaults to False.

Returns:

Slope, intercept, standard error of the slope, standard error of the intercept, and the R-squared value.

```
src.utils.regression_analysis.plot_regression(x, y, sdx_lower, sdx_upper, sdy_lower, sdy_upper,  
                                              slope, intercept, se_slope, se_intercept, r_squared,  
                                              label, color, ax, fig)
```

Plots the data and the regression line.

Args:

- `x` (array-like): The x data.
- `y` (array-like): The y data.
- `sdx_lower` (array-like): Lower standard deviations of the x data.
- `sdx_upper` (array-like): Upper standard deviations of the x data.
- `sd_y_lower` (array-like): Lower standard deviations of the y data.
- `sd_y_upper` (array-like): Upper standard deviations of the y data.
- `slope` (float): Slope of the regression line.
- `intercept` (float): Intercept of the regression line.
- `se_slope` (float): Standard error of the slope.
- `se_intercept` (float): Standard error of the intercept.
- `r_squared` (float): R-squared value.
- `label` (str): Label for the plot.
- `color` (str): Color for the plot.
- `ax` (matplotlib.axes.Axes): Axes object to draw the plot onto.
- `fig` (matplotlib.figure.Figure): Figure object containing the Axes.

Returns:

PyQt5.QtGui.QPixmap: QPixmap representation of the plot.

```
src.utils.regression_analysis.read_data(filename)
```

Reads data from an Excel file.

Args:

- `filename` (str): Path to the Excel file.

Returns:

Data extracted from the file or None if an error occurs.

save module

save.py

Author: Dongzi Ding Created: 2023-06-25 Modified: 2023-08-14

`src.utils.save.save(result, dirname, figures)`

Saves the result data to CSV files and figures to PNG files.

Parameters:

- `result` (dict): Dictionary containing the analysis results.
- `dirname` (str): Directory path where the results will be saved.
- `figures` (dict): Dictionary containing figures for saving as PNG.

Returns:

None

1.2 mainwindow module

1.2.1 mainwindow.py

Author: Dongzi Ding Created: 2023-06-25 Modified: 2023-08-14

Main window for the application. This module provides the main application window for the PyQt5-based GUI application. It includes menu bars for feature selections, input settings, save settings, help, and developer contact.

`class src.mainwindow.MainWindow(parent=None)`

Bases: `QMainWindow`

The main window for the PyQt5-based GUI application.

`check_calculate_button_state()`

Checks if the 'Calculate' button should be enabled.

`open_contact()`

Opens the appropriate contact method based on the menu selection.

`open_help()`

Opens the Help window.

`select_option1()`

`select_option2()`

`select_option3()`

`select_option4()`

`select_option5()`

`select_option6()`

`select_option7()`

`select_option8()`

`toggle_option(checked, option)`

Toggles the current function option.

update_func_option(*checked*)

Updates the current function option based on the menu selection.

class src.mainwindow.**Settings**

Bases: QObject

Represents application settings.

Attributes:

- func_current_options (dict): Current functional options selected.

reset()

Resets the settings to default values.

set_func_option(*option*)

Sets the current function option.

set_input_option(*option*)

Sets the current input option.

set_save_option(*option*)

Sets the current save option.

settings_changedsrc.mainwindow.**resource_path**(*relative_path*)

Gets the absolute path to a resource, works in both development and PyInstaller contexts.

Args:

- relative_path (str): The relative path to the resource.

Returns:

The absolute path to the resource.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

S

- src, 1
- src.gui, 1
 - src.gui.button_area, 1
 - src.gui.help_window, 2
 - src.gui.input_window, 2
 - src.gui.result_window, 3
 - src.gui.settings_window, 3
 - src.gui.visual_window, 3
- src.mainwindow, 10
- src.utils, 4
 - src.utils.initial_rate, 4
 - src.utils.input_help, 5
 - src.utils.plane3D_plot, 6
 - src.utils.rate_const, 8
 - src.utils.regression_analysis, 8
 - src.utils.save, 10

INDEX

A

`add_result()` (*src.gui.result_window.ResultWindow* method), 3

B

`browse_file()` (*src.gui.input_window.InputWindow* method), 2

`ButtonArea` (class in *src.gui.button_area*), 1

C

`calculate()` (*src.gui.button_area.ButtonArea* method), 1

`calculate_rate()` (in module *src.utils.initial_rate*), 4

`calculate_rate()` (in module *src.utils.rate_const*), 8

`calculate_rate_compare()` (in module *src.utils.initial_rate*), 4

`calculate_regression()` (in module *src.utils.regression_analysis*), 8

`check_calculate_button_state()` (*src.mainwindow.MainWindow* method), 10

`clear_data()` (*src.gui.input_window.InputWindow* method), 2

`confirm_input()` (*src.utils.input_help.DataInputDialog* method), 5

`create_3D_plot()` (*src.utils.plane3D_plot.Plane3DPlotter* method), 6

`cut_data()` (in module *src.utils.initial_rate*), 4

D

`DataInputDialog` (class in *src.utils.input_help*), 5

E

`emit_input_changed()` (*src.gui.input_window.InputWindow* method), 2

F

`fig_to_pixmap()` (*src.utils.plane3D_plot.Plane3DPlotter* method), 6

`fit_plane()` (*src.utils.plane3D_plot.Plane3DPlotter* method), 6

G

`get_input_data()` (*src.utils.input_help.DataInputDialog* method), 5

`get_options()` (*src.gui.button_area.OptionDialog* method), 1

`get_results()` (*src.utils.plane3D_plot.Plane3DPlotter* method), 7

H

`HelpWindow` (class in *src.gui.help_window*), 2

I

`input_changed` (*src.gui.input_window.InputWindow* attribute), 2

`InputWindow` (class in *src.gui.input_window*), 2

M

`MainWindow` (class in *src.mainwindow*), 10

`manual_input()` (*src.gui.input_window.InputWindow* method), 2

module

src, 1

src.gui, 1

src.gui.button_area, 1

src.gui.help_window, 2

src.gui.input_window, 2

src.gui.result_window, 3

src.gui.settings_window, 3

src.gui.visual_window, 3

src.mainwindow, 10

src.utils, 4

src.utils.initial_rate, 4

src.utils.input_help, 5

src.utils.plane3D_plot, 6

src.utils.rate_const, 8

src.utils.regression_analysis, 8

src.utils.save, 10

O

`open_contact()` (*src.mainwindow.MainWindow* method), 10

`open_help()` (*src.mainwindow.MainWindow* method), 10

`open_pdf_manual()` (*src.gui.help_window.HelpWindow* method), 2

`OptionDialog` (class in *src.gui.button_area*), 1

P

`perform_analysis()`

(src.utils.plane3D_plot.Plane3DPlotter method), 7
Plane3DPlotter (class in src.utils.plane3D_plot), 6
plot() (in module src.utils.rate_const), 8
plot_3D_data() (src.utils.plane3D_plot.Plane3DPlotter method), 7
plot_fitted_plane() (src.utils.plane3D_plot.Plane3DPlotter method), 7
plot_initial_rate() (in module src.utils.initial_rate), 4
plot_rate_comparison() (in module src.utils.initial_rate), 4
plot_regression() (in module src.utils.regression_analysis), 9

R

read_data() (in module src.utils.initial_rate), 5
read_data() (in module src.utils.rate_const), 8
read_data() (in module src.utils.regression_analysis), 9
read_data() (src.utils.plane3D_plot.Plane3DPlotter method), 7
reset() (src.gui.button_area.ButtonArea method), 1
reset() (src.gui.input_window.InputWindow method), 2
reset() (src.mainwindow.Settings method), 11
resource_path() (in module src.mainwindow), 11
ResultWindow (class in src.gui.result_window), 3

S

save() (in module src.utils.save), 10
save_result() (src.gui.button_area.ButtonArea method), 1
select_all() (src.utils.input_help.DataInputDialog method), 5
select_option1() (src.mainwindow.MainWindow method), 10
select_option2() (src.mainwindow.MainWindow method), 10
select_option3() (src.mainwindow.MainWindow method), 10
select_option4() (src.mainwindow.MainWindow method), 10
select_option5() (src.mainwindow.MainWindow method), 10
select_option6() (src.mainwindow.MainWindow method), 10
select_option7() (src.mainwindow.MainWindow method), 10
select_option8() (src.mainwindow.MainWindow method), 10
set_func_option() (src.mainwindow.Settings method), 11
set_input_option() (src.mainwindow.Settings method), 11
set_save_option() (src.mainwindow.Settings method), 11

Settings (class in src.mainwindow), 11
settings_changed (src.mainwindow.Settings attribute), 11
SettingsWindow (class in src.gui.settings_window), 3
show_result() (src.gui.button_area.ButtonArea method), 1
show_visual() (src.gui.button_area.ButtonArea method), 1

src module, 1
src.gui module, 1
src.gui.button_area module, 1
src.gui.help_window module, 2
src.gui.input_window module, 2
src.gui.result_window module, 3
src.gui.settings_window module, 3
src.gui.visual_window module, 3
src.mainwindow module, 10
src.utils module, 4
src.utils.initial_rate module, 4
src.utils.input_help module, 5
src.utils.plane3D_plot module, 6
src.utils.rate_const module, 8
src.utils.regression_analysis module, 8
src.utils.save module, 10

T

toggle_option() (src.mainwindow.MainWindow method), 10

U

update_content() (src.gui.input_window.InputWindow method), 2
update_content() (src.gui.settings_window.SettingsWindow method), 3
update_func_option() (src.mainwindow.MainWindow method), 10
update_input_fields() (src.utils.input_help.DataInputDialog method), 5
update_start_button() (src.gui.button_area.ButtonArea method), 1

`use_both()` (*src.gui.button_area.OptionDialog*
 method), 1
`use_sklearn()` (*src.gui.button_area.OptionDialog*
 method), 2

V

`VisualWindow` (*class in src.gui.visual_window*), 3