Independent Research Project

Release 1.0.0

Dongzi Ding

Aug 18, 2023

CONTENTS:

CHAPTER

ONE

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()

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: 2023-08-17

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

Attributes:

- filename (str): Name of the input Excel file containing data.
- data (tuple): Processed data from the Excel file.
- log_initial_concentration (array): Logged values of initial concentrations.
- log_initial_rate (array): Logged values of initial rates.
- pH (array): pH values from the data.
- params (tuple): Parameters of the fitted plane.
- r_squared (float): R-squared value of the fitted model.

`create_3D_plot`(*ax=None*)

Creates a 3D plot with data points.

Args:

- ax (Axes3D, optional): Existing 3D axes if provided. Defaults to None.

Returns:
fig, ax: Figure and axes objects of the plot.

`fig_to_pixmap`(*fig*)

Converts a matplotlib figure to a QPixmap.

Args:

- fig (Figure): Matplotlib figure to be converted.

Returns:
QPixmap: Converted QPixmap of the figure.

`fit_plane`(*log_initial_concentration*, *pH*, *log_initial_rate*)

Fits a plane to the given 3D data.

Args:

- log_initial_concentration (array): Logged values of initial concentrations.
- pH (array): pH values.
- log_initial_rate (array): Logged values of initial rates.

Returns:
tuple: Parameters of the fitted plane and R-squared value.

`get_results`()

Returns the parameters of the fitted plane and R-squared value.

Returns:
dict: Parameters of the fitted plane and R-squared value.

`perform_analysis`()

Performs 3D analysis on the data.

Returns:
tuple: Parameters of the fitted plane and R-squared value.

`plot_3D_data`(*ax=None*)

Plots the 3D data and fitted plane.

Args:

- ax (Axes3D, optional): Existing 3D axes if provided. Defaults to None.

Returns:
fig, ax: Figure and axes objects of the plot.

`plot_fitted_plane`(*ax*, *params*)

Plots the fitted plane on the given 3D axes.

Args:

- ax (Axes3D): 3D axes object to plot on.
- params (tuple): Parameters of the fitted plane.

`read_data`(*filename*)

Reads data from the specified Excel file and returns logged values.

Args:

- filename (str): Name of the Excel file to read from.

Returns:
tuple: Logged initial concentration, logged initial rate, and pH values.

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: Pixmap 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: 2023-08-17

`src.utils.regression_analysis.`**`calculate_log_values`**(*initial_concentration*, *initial_rate*)

Calculates the log values of the initial concentration and rate.

Args:

- initial_concentration (array-like): Initial concentrations.
- initial_rate (array-like): Initial rates.

Returns:
Log values of the initial concentration and rate.

src.utils.regression_analysis.**calculate_regression**(*log_concentration*, *log_rate*)

Calculates the linear regression of the log values using sklearn.

Args:

- log_concentration (array-like): The log concentration data.
- log_rate (array-like): The log rate data.

Returns:
Slope (reaction order), intercept, and the R-squared value.

src.utils.regression_analysis.**plot_regression**(*log_concentration*, *log_rate*, *slope*, *intercept*, *r_squared*, *label*, *color*, *ax*, *fig*)

Plots the data and the regression line.

Args:

- log_concentration (array-like): The log concentration data.
- log_rate (array-like): The log rate data.
- slope (float): Slope of the regression line.
- intercept (float): Intercept of the regression line.
- 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.

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_changed

src.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.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX

R

S

T

U

V