**Imports:**

* from ...misc import create\_report
* from ...signal import signal\_sanitize
* from .eda\_clean import eda\_clean
* from .eda\_methods import eda\_methods
* from .eda\_peaks import eda\_peaks
* from .eda\_phasic import eda\_phasic
* from .eda\_plot import eda\_plot

**Explanation of Functions:**

* **create\_report**: A function to generate a report of the processing.
* **signal\_sanitize**: A utility to ensure the input signal is in a standard format.
* **eda\_clean**: A function specifically for cleaning the EDA signal.
* **eda\_methods**: A helper function to get the correct methods and parameters for processing.
* **eda\_peaks**: A function to find peaks in the EDA signal.
* **eda\_phasic**: A function to separate the phasic component of the EDA signal.
* **eda\_plot**: A function to create plots of the EDA signal.

**Optional Arguments:**

* **report = None**: This is an optional argument. If you provide a filename (e.g., 'my\_eda\_report.html'), the function will generate an HTML report with plots and descriptions of the analysis.
* **\*\*kwargs**: This allows you to pass additional arguments to the underlying functions.

**Data Dictionary:**

* **info (dictionary)**: A dictionary containing detailed information about each detected SCR peak, such as timing and amplitude, as well as the sampling rate.

**Sanitize Input:**

**eda\_signal = signal\_sanitize(eda\_signal)**

This line takes the raw EDA signal and makes sure it's in a consistent format (1-D array), so the rest of the function can work with it.

methods = eda\_methods(sampling\_rate, method="neurokit", \*\*kwargs)

This line calls the eda\_methods function to get a dictionary containing the specific cleaning, decomposition, and peak detection methods and their parameters based on the chosen method (e.g., "neurokit").

**Preprocess**

**# clean signal**

**eda\_cleaned = eda\_clean(eda\_signal, sampling\_rate=" ",**

**method=methods["method\_cleaning"],**

**\*\*methods["kwargs\_cleaning"])**

Here, the raw EDA signal is cleaned to remove noise and artifacts using the eda\_clean function. The specific cleaning method and its settings are retrieved from the methods dictionary.

**If/Else Logic**

* The if statement checks if a decomposition method is specified. If not, it assumes the cleaned signal is the phasic component.
* The else block calls the eda\_phasic function to perform the decomposition, which typically splits the eda\_cleaned signal into its tonic (EDA\_Tonic) and phasic (EDA\_Phasic) components.

**Find Peaks**

**peak\_signal, info = eda\_peaks(eda\_decomposed["EDA\_phasic"],**

**sampling\_rate=sampling\_rate,**

**method=methods["method\_peaks"],**

**amplitude\_min=0.01,**

**\*\*methods["kwargs\_peaks"])**

This is where the function identifies the Skin Conductance Responses (SCRs).

* It takes the phasic component of the signal.
* It uses the eda\_peaks function to find the onsets, peaks, and recovery times of the responses.
* amplitude\_min=0.1 sets the threshold so that very small fluctuations are not counted as significant responses.
* It returns peak\_signal, a DataFrame containing columns for SCR onsets and SCR peaks, and info, a dictionary with details about each peak.

# Store

signals = pd.DataFrame({"EDA\_Raw": eda\_signal, "EDA\_Clean": eda\_cleaned})

A new pandas DataFrame called signals is created. Initially, it contains the raw and cleaned EDA signals.

signals = pd.concat([signals, eda\_decomposed, peak\_signal], axis=1)

* **concat**: A pandas function to join tables.
* **axis=1**: They are joined side-by-side as new columns.
* It merges the signals DataFrame (with raw and clean data), the eda\_decomposed DataFrame (with tonic and phasic data), and the peak\_signal DataFrame (with SCR event data).

if report is not None:

# Generate report containing description & figures of processing

if ".html" in str(report):

fig = eda\_plot(signals, info, static=False)

else:

fig = None

create\_report(file=report, signals=signals, info=methods, fig=fig)

* It checks if a report filename was provided.
* If the filename ends in .html, it creates an interactive plot using eda\_plot.
* It then calls create\_report to generate the final HTML file with the plot and other processing details.