**Generalization of chromatoPy package**

**No changes:**

1. run
2. multigaussian
3. gaussian\_decay
4. individual\_gaussian
5. estimate\_initial\_gaussian\_params
6. find\_valleys
7. calculate\_boundaries
8. smoother
9. forward\_derivative
10. extrapolate\_gaussian
11. calculate\_gaus\_extension\_limits
12. peak\_area\_distribution
13. add\_window\_controls
14. baseline
15. highlight\_subplot

**Removed:**

1. clear\_all\_peaks
2. collect\_peak\_data
3. append\_peak\_data

**chromatopy\_gen.py:**

SignalAnalyzer class and constructor:

1. traces removed.
2. GDGT\_dict removed.
3. axs\_to\_traces removed.
4. is\_reference moved towards the end; run wasn’t working for some reason.
5. fig and axs were initialized twice; removed redundancy.
6. peaks is now a list; used to be a dictionary.
7. called removed.
8. t\_pressed is now r\_pressed.

find\_peak\_neighborhood\_boundaries:

1. trace removed from function signature.
2. [trace] removed for l\_lim and r\_lim initialization.

find\_peak\_boundaries:

1. trace removed from function signature; trace wasn’t being used anywhere in this function to begin with.

fit\_gaussians:

1. trace removed from function signature; trace wasn’t being used anywhere in this function to begin with.

handle\_peak\_selection:

1. peak\_results setup changed because traces aren’t there anymore.
2. find\_ peak\_neighbourhood\_boundaries and fit\_gaussians calls has trace removed

plot\_data:

1. if statements at start are removed because traces aren’t there anymore.
2. datasets and peaks\_indices were initialized to [None] \* no. of traces. No more needed.
3. if statement removed from the loop because no traces.
4. setup\_subplot call only takes one argument now

setup\_subplot:

1. trace\_idx removed from signature.
2. x\_values is retrieved using “RT (min)” as the key; used to be “rt\_corr.”
3. trace is removed.
4. y is retrieved using “Signal” as the key.
5. hasattr if statements are removed.
6. full\_data used to be a dictionary; now it is a tuple (x\_values, y\_filtered).
7. line\_objects used to be a dictionary; now it is line\_object storing a line object.
8. axs\_to\_traces doesn’t exist anymore.
9. datasets used to be a dictionary; now it is a list storing a tuple (x\_values, y\_bcorr).
10. Population of peaks, peak\_properties and peaks\_indices is different because no traces.

on\_click:

1. event.inaxes check in axs or list of axes now instead of axs\_to\_traces.
2. trace retrieval removed.
3. ax\_idx = 0 because one subplot.
4. Loop for plt.draw() is removed.
5. One plt.draw() towards the end which his now outside the if statement.
6. For non-peak clicks, that is, no peak found, additional no peak mark was added to peak\_key tuple. It is a string that says “no\_peak\_at\_{retention time of the click}.”
7. For non-peak clicks, in integrated\_peaks, area\_ensemble is now stored as [0] for consistency.
8. Added non-peak clicks to peak\_results to get non-peak clicks in automation.

auto\_select\_peaks:

1. Changed to work without traces. Many changes. Mostly similar to on\_click function.

on\_key:

1. Not emptying peak\_results now.
2. collect\_peak\_data matching\_peaks functioning incorporated into “enter” key functioning.
3. Up and down movement key functioning removed because one subplot.
4. “r” functioning changed because no traces.
5. “t” functioning incorporated within “r” key, that is, action\_stack clearance, r\_pressed initilalization and print statement.
6. Added “e” key functioning for exit process. User presses “e” to exit the session. Basically, end the run.

undo\_last\_action:

1. Peak\_results updation added.
2. But there was a bug in the old version which only allowed to undo the latest non-peak click.
3. This bug was cleared by making changes in on\_click.

clear\_peaks\_subplot:

1. ax\_idx removed from signature.
2. Axis index is 0 because one subplot.
3. setup\_subplot call parameters are changed due to change in signature of setup\_subplot.

**hplc\_integration\_gen.py:**

1. User is prompted to enter a folder\_path.
2. All files in folder a handled in parallel and dataframe is made for each file.
3. windows, by default is true. If true, default value for bounds is taken. If user passes windows as false in the argument, user will be prompted to input the window bounds.
4. First sample follow the selection process to setup the reference. After that automation follows. If user presses “r”, the current sample is used to make a new reference, and the reference peak is updated.
5. Reference peak is handled in the wrapper. Set to None, changes to the reference that is set and when “r” is pressed, a new reference is set based on the current sample’s selection results within the wrapper.
6. If user presses “e” at any point, it exits the session.
7. Takes in a compounds list.
8. Added functioning for output saving.
9. Added abortion handling to handle output cleanup when aborted.

**Notes:**

1. matching\_peaks in on\_key “enter” key functioning isn’t used anywhere nor returned. Probably not useful in our new generalized version. Might be removable.
2. waiting\_for\_input in on-key for “enter” key functioning also seems to have no use. Probably not useful in our new generalized version. Might be removable.

**Functioning:**

1. User inputs parameters otherwise default parameters are taken.
2. If folder\_path is not inputted, i.e., it is None, user is prompted. It is checked if the folder exists for the given path. csv files are extracted into a list from the folder, and they are read in parallel into separate dataframes. Dataframes are returned as a list called data. Done by folder\_handling and read\_data\_concurrently
3. If windows parameter is inputted as false, prompts user for custom window bounds. Done by window\_handling.
4. Initializes some variable necessary for the analyzer.
5. Output directories and paths are setup and returned. Done by output\_handling.
6. If results csv file already exists, it is read into the output dataframe, otherwise a new output dataframe is created.
7. Each sample’s dataframe is analyzed from data. Sample name is extracted from the dataframe. If the sample is already analyzed, i.e., already present in the results dataframe, sample is skipped. If not, analyzer is run.
8. Analyzer returns results dict called peaks, reference peaks called ref\_new, the figure called fig and booleans to track ‘r’ and ‘e’ key presses.
9. Results dataframe population begins. If no. of peak clicks are more than no. of compounds, an error is raised. If not, output dataframe starts getting populated.
10. Figure and individual sample data is saved for each sample.
11. Now reference is handled, if iref was true, i.e., this was the first sample, reference is set.
12. If r was pressed, reference is set to the results of the latest sample at which r was clicked.
13. If e was pressed, it means the program was aborted at sample ‘x’.
14. abortion\_handling handles cleanup of the folders and files and production of accurate results based on abortion. Figures, individual data and sample area data for results csv file for samples after the abortion sample are removed.
15. Output dataframe is sorted by sample name before being read into the results csv file.
16. Final message is displayed depending on abortion.