(A)

hplc-py/io.py

`fn: load_chromatogram()`

Given a chromatogram `.txt` file, parse header and identify measurements given names of "time" and "signal" columns.

hplc-py/quant.py

`class: Chromatogram()`

`mth: crop()`

Restricts time-dimension of chromatogram in-place, given start and stop bounds.

`mth: fit_peaks()`

Infers and subtracts baseline using the SNIP algorithm, chunks chromatogram into regions with isolated and overlapping peaks, fits a mixture of weighted skew-normal distributions, and computes peak properties (e.g. integrated area, retention time).

`mth: assess_fit()`

Prints a report card scoring how well the fit mixture reconstructs the raw chromatogram.

`mth: map_peaks()`

Assigns a compound identity to each peak given user-supplied dictionary of identities and retention time. Given a linear calibration curve, the concentration of each compound is also computed.

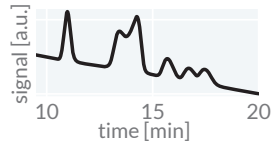
`mth: show()`

Plots the baseline-corrected chromatogram, estimated baseline signal, individual peaks, and inferred mixture of all fit peaks.

(B)

instantiation

```
data = load_chromatogram(fname,cols)
chrom = Chromatogram(data)
```

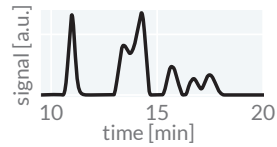


(C)

`chrom.fit_peaks()`

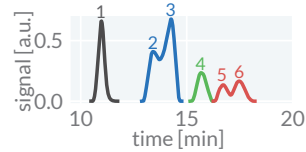
i **background subtraction**

`chrom.correct_baseline()`



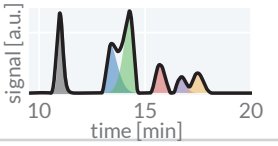
ii **peak detection**

`chrom._assign_windows()`



iii **peak fitting**

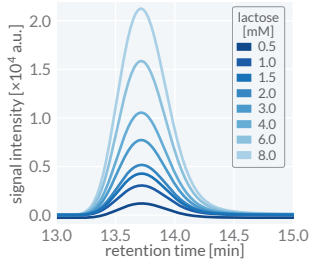
`chrom.deconvolve_peaks()`



(D) **peak measurements**

ret_time	amp.	scale	skew	area	peak
10.90	2.3e+4	0.16	0.70	2.8e+6	1
13.19	2.5e+4	0.42	3.19	3.0e+6	2
14.45	3.7e+4	0.38	-3.53	4.4e+6	3
⋮	⋮	⋮	⋮	⋮	⋮

(E)



(F)