The Journal of Open Source Software

hplc-py: A Python Utility For Rapid Quantification of Complex Chemical Chromatograms

Griffin Chure [1,2] **and Jonas Cremer** [1]

1 Department of Biology, Stanford University, CA, USA **2** For correspondence, contact griffinchure@gmail.com

DOI: 10.21105/joss.06270

Software

- Review
- Repository
- Archive

Editor: Jeff Gostick

Reviewers:

- @florian-huber
- @Kastakin

Submitted: 05 October 2023
Published: 10 February 2024

Summary

High-Performance Liquid Chromatography (HPLC) and Gas Chromatography are analytical techniques which allow for the quantitative characterization of the chemical components of mixtures [Figure 1(A)]. Technological advancements in sample preparation and mechanical automation have allowed HPLC to become a high-throughput tool (Broeckhoven et al., 2019; Kaplitz et al., 2020) which poses new challenges for reproducible and rapid analysis of the resulting chromatograms. Here we present `hplc-py`, a Python package that permits rapid and reliable quantitation of component signals within a chromatogram for pipelined workflows. This is achieved by a signal detection and quantitation algorithm which i) identifies windows of time which contain peaks and ii) infers the parameters of a mixture of amplitude-weighted skew-normal distributions which sum to reconstruct the observed signal. This approach is particularly effective at deconvolving highly overlapping signals, allowing for precise absolute quantitation of chemical constituents with similar chromatographic retention times.

Statement of Need

Chromatography has become a gold-standard method across diverse fields for precise quantitation and separation of chemical mixtures. A key objective in the analysis of chromatographic data is determining the time-integrated signal of each component, a process which becomes challenging when chemically-similar components result in strongly overlapping signals [such as the blue and green symbols in Figure 1(B)]. As of this writing, many of the available tools for signal quantification, such as the open source Python 2.7 software `HappyTools` (Jansen et al., 2018), Microsoft Excel applications (Cruz Villalon, 2023), or proprietary solutions such as Chromeleon by Thermo-Fisher and Empower by Waters, rely on manual processing of the chromatograms and curation of the resulting quantitative data. Furthermore, we are unaware of any tools that can reliably deconvolve highly overlapping signals. `hplc-py` provides a programmatic interface by which users can quickly and reliably quantify components of complex chromatograms in a few lines of code [Figure 1(C)]. Importantly, the peak detection and fitting algorithm of `hplc-py` is able to deconvolve completely overlapping signals, allowing for the accurate quantification of mixtures otherwise not separable without extensive experimental optimization.

The Journal of Open Source Software

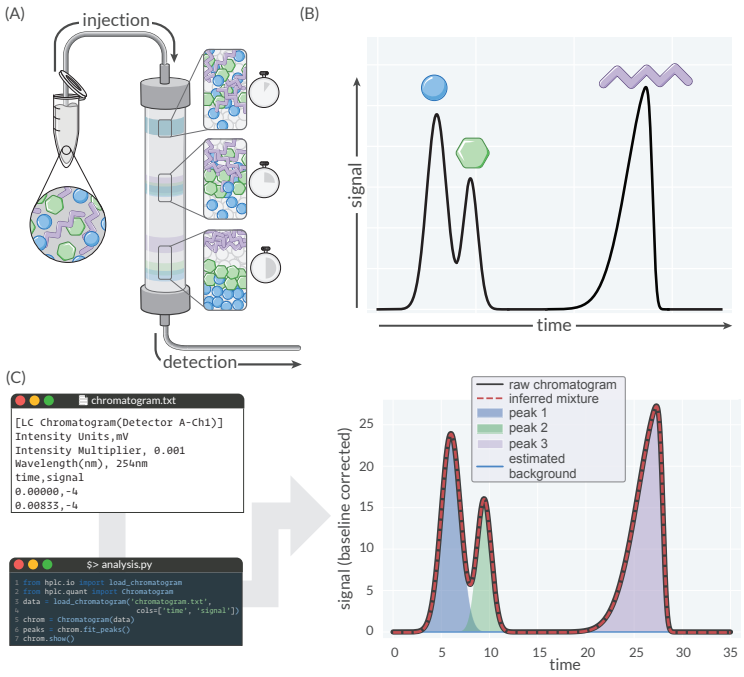

Figure 1: Chromatographic separation of chemical compounds and their detection with `hplc-py`. (A) Diagrammatic view of the chromatographic principle. (B) A simulated chromatogram of the three separated compounds diagrammed in panel A. (C) Passing this simulated chromatogram through the methods of the `Chromatogram` object of `hplc-py` allows for deconvolution and quantification of individual signals which sum to reconstruct the observed chromatogram. Code used to generate panels (B) and (C) is available on the GitHub repository publication branch

Methodology

The major components of `hplc-py` are diagrammed in Figure 2(A). A helper function, `load_chromatogram`, can be used to read a raw text file, filter through the metadata in the header, and retrieve the time and signal data, given user-supplied column names [Figure 2(B)]. The resulting pandas DataFrame object can be passed to the `Chromatogram` object, which has a slew of methods for cropping, fitting, scoring, quantifying, and plotting the chromatogram. The core algorithmic steps employed by `hplc-py` are diagrammed in Figure 2(C) and presented in detail on the package documentation. Once a `Chromatogram` has been instantiated, automated detection and quantification of peaks which compose the observed chromatogram can be executed by calling the `.fit_peaks` method. Under the hood, this method calls three helper functions [diagrammed in Figure 2(C)] which preform the following steps:

`i)` Estimation of and correction for a variable baseline. A common challenge in the analysis of HPLC data is the identification and removal of spurious background signal. While the physicochemical basis for baseline variance is complex (Choikhet et al., 2003; Felinger & Káré, 2004), numerous methods have been developed for their correction (Macko & Berek, 2001; Mecozzi, 2014). In `hplc-py`, this is implemented using the Sensitive Nonlinear Iterative Peak (SNIP) method originally developed for smoothing of spectroscopic data (Morháč & Matoušek, 2008).

`ii)` Peak identification and separation of the chromatogram into region windows. After any variable background has been identified and corrected, peak-filled regions of the chromatogram are identified through the application of topographic prominence thresholds, a method common in the signal processing of neuron action potentials (Choi et al., 2017). With peak locations

identified, the chromatogram is further clipped into windows–regions in time when chemical species co-elute and therefore overlap.

iii) Fitting a mixture of amplitude-weighted skew-normal distributions to each peak window. For an assigned peak window with N peaks, `hplc-py` fits a convolution of N amplitude-weighted skew-normal distributions to the observed signal S within that window. A weighted skew-normal distribution is parameterized by an amplitude A, location and scale parameters τ and σ, and a skew parameter α and has the form

$$S(t) = \frac{A}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(t-\tau)^2}{2\sigma^2}\right]\left[1 + \text{erf}\left(\frac{\alpha(t-\tau)}{\sqrt{2\sigma^2}}\right)\right], \tag{1}$$

where t is the time point and erf is the error function. The skew-normal distribution is useful in fitting chromatogram signals as peaks are often asymmetric with high skewness, a property described by a single parameter α.

The `.fit_peaks` method returns a Pandas DataFrame [Figure 2(D)] which reports the best-fit values for each parameter for each peak. Importantly, it also returns the integral of Equation 1 for each compound over a given time window which is linearly proportional to the concentration of the analyte (Moosavi & Ghassabian, 2018). Figure 2(E-F) demonstrates that the peak quantification algorithm of `hplc-py` yields a linear relationship between concentration and integrated area for a standard curve of a lactose sugar solution across a decade of concentrations.

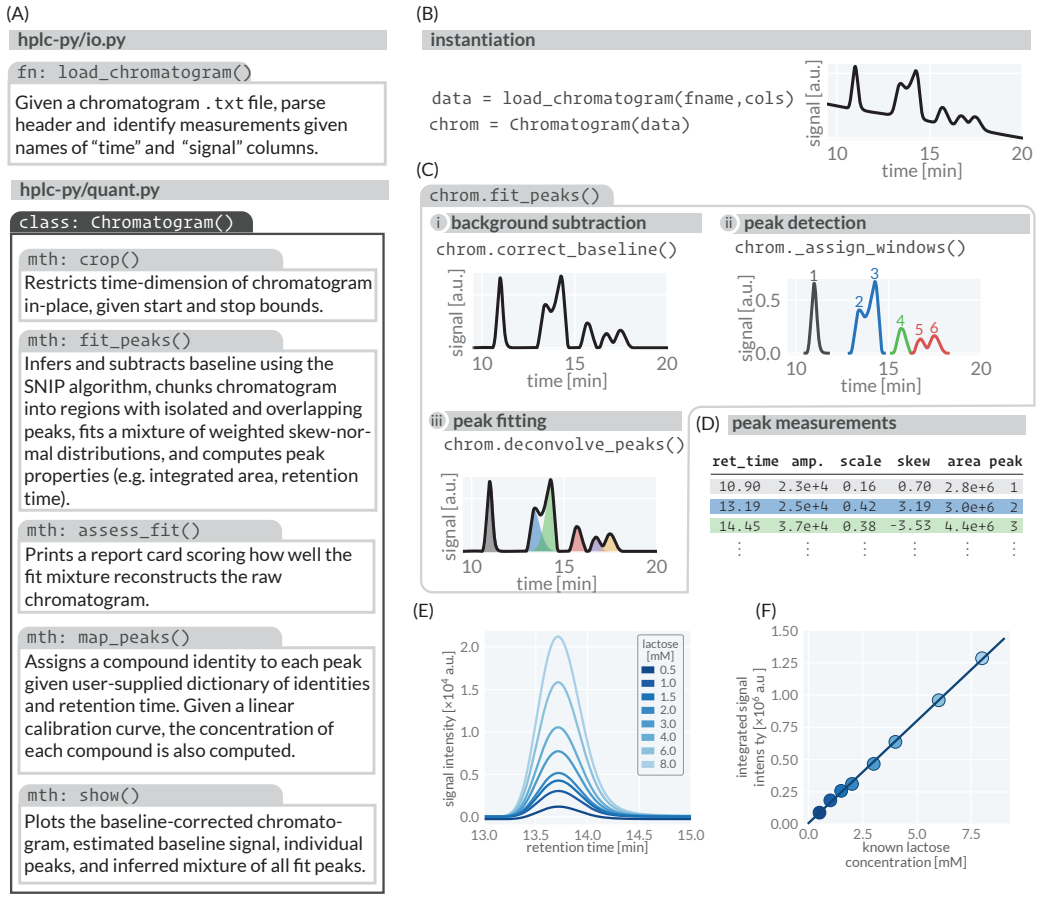

Figure 2: The peak quantification algorithm implemented in `hplc-py` as applied to a real chromatogram. (A) The `hplc-py` data model. (B) A `Chromatogram` object is instantiated by loading a raw chromatogram text file as a Pandas DataFrame. (C) The peak quantification operations undertaken by the `fit_peaks()` method of a `Chromatogram` object. (D) A representative peak quantification table returned by `.fit_peaks()`. (E) Representative signals of a lactose solution with different concentrations. (F) A calibration curve generated from panel E using `hplc-py`. Code used to generate these figure panels are available on the GitHub repository publication branch.

Constraining Peak Parameters and Overlapping Signals

The separation efficiency of different chemical species through HPLC is dependent on myriad variables, including chemical properties of the column, the solvent, the operational temperature, and column dimensions. It is common for some chemical species to co-elute in a given experimental configuration. For example, the sugar lactose [Figure 3(A, blue)] and inorganic ion phosphate [Figure 3(A, purple)] have almost identical elution times on a Rezex Organic Acid H+ 8% column with a 2.5 mM H_2SO_4 mobile phase, resulting in a convolution which can be mistaken for a single peak [Figure 3(A, dashed line)]. As a consequence, these signals would be classified as inseparable using other HPLC data analysis programs and further experimental optimization would be needed to resolve them.

However, as `hplc-py` fits mixtures of weighted distributions instead of empirically summing over the signal itself, it is possible to quantitatively resolve these signals. This can be performed by tightly constraining the parameters of one of the two confounding signals, such as phosphate. As an example, we have considered a use case where phosphate is present in a fixed concentration across samples whereas the lactose concentration can vary. Under such a scenario, `hplc-py` can be used to independently characterize the parameters which define the size and shape of

the phosphate peak [Figure 3(B)]. With these parameters in hand, one can tightly constrain the parameter regimes (as described in the hplc-py documentation) for the phosphate signal within a mixture [Figure 3(C)], allowing for effective estimation of parameter values for the lactose peak. Using this approach and the lactose calibration curve shown in Figure 2(E), we were able to measure the lactose concentration within a wide range of lactose-phosphate mixtures with quantitative accuracy [Figure 3(D)]. Such accuracy is lost when parameters for both the phosphate and lactose peaks are allowed to be freely estimated [Figure 3(F)], even though the inferred chromatogram reconstruction is in agreement with the observed signal [Figure 3(E)].

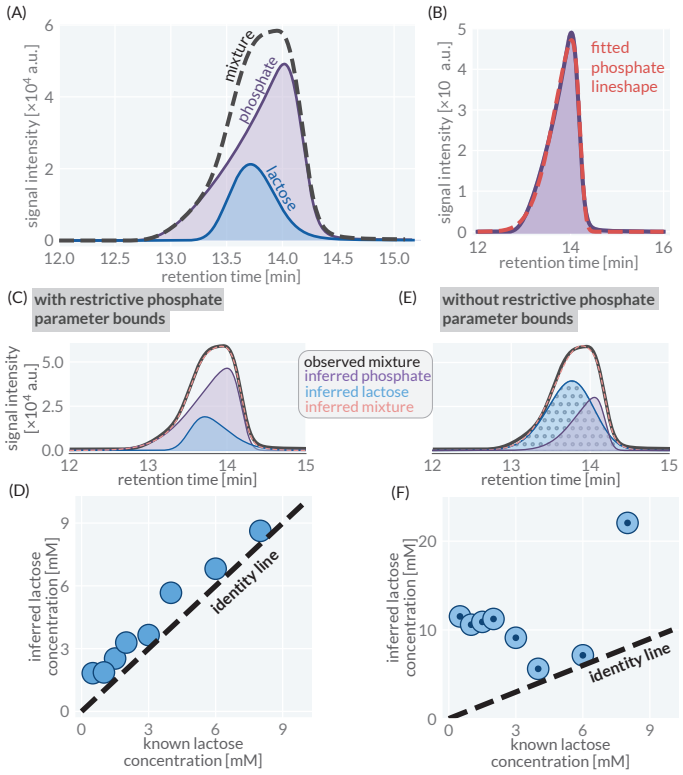

Figure 3: hplc-py decomposes completely overlapping signals of phosphate and lactose in a real chromatogram with quantitative accuracy. (A) Three overlaid chromatograms of lactose (blue), phosphate-based buffer (purple), and a lactose-phosphate mixture as detected on a Rezex Organic Acid H+(8%) column with 2.5 mM H_2SO_4 mobile phase. (B) The best-fit lineshape (red) of the phosphate signal (purple) as computed by hplc-py. The inferred underlying distributions of lactose (blue) and phosphate (purple) with constrained or unconstrained phosphate parameters in (C) and (D), respectively. Inferred lactose concentration compared to the known concentration in the mixture for the constrained and unconstrained phosphate parameters are shown in (E) and (F), respectively. Code used to perform this analysis and generate these figures is available on the GitHub repository publication branch

In total, hplc-py provides a programmatic interface that allows experimentalists to rapidly quantify chemical signals from chromatograms, even when there is exceedingly high overlap between analytes. While we have tailored the default parameters of the hplc-py methods to be amenable to typical HPLC chromatographic outputs, we have also made it simple to manually adjust different aspects of the peak quantification algorithm, including parameters controlling the degree of background subtraction, the constraint of fitting parameters, and even enforcing the estimation of peaks with low topographic prominence. Additionally, we have developed heuristics the user can employ to assess quality of the reconstruction, though we emphasize that this is not to be used as a measure of uncertainty. We hope that hplc-py can act as a tool that can make scientific interpretation of results, rather than their generation,

the next bottleneck in HPLC-based experiments.

Data & Code Availability

All experimental data and code used to process data schematized in the Figures are publicly available on the `hplc-py` GitHub repository `publication` branch. Data for Figure 3 was collected as described in the `README.md` file of the experimental data folder of the repository publication branch.

Acknowledgements

We thank Markus Arnoldini and Richa Sharma for extensive discussion of software needs and for prototyping early releases on various types of HPLC data. We thank Olivia Warren for advice in software design and implementation. Griffin Chure acknowledges financial support by the NSF Postdoctoral Research Fellowships in Biology Program (grant no. 2010807).

References

Broeckhoven, K., Shoykhet, K., & Dong, M. (2019). Modern HPLC Pumps: Perspectives, Principles, and Practices. *LCGC North America*, *37*(6), 374–384. https://www.chromatographyonline.com/view/modern-hplc-pumps-perspectives-principles-and-practices

Choi, M.-H., Ahn, J., Park, D. J., Lee, S. M., Kim, K., Cho, D. D., Senok, S. S., Koo, K., & Goo, Y. S. (2017). Topographic Prominence Discriminator for the Detection of Short-Latency Spikes of Retinal Ganglion Cells. *Journal of Neural Engineering*, *14*(1), 016017. https://doi.org/10.1088/1741-2552/aa5646

Choikhet, K., Glatz, B., & Rozing, G. (2003). The Physicochemical Causes of Baseline Disturbances in HPLC: TFA-containing eluents. *LCGC International*, *6*(2), 96–105. https://api.semanticscholar.org/CorpusID:19173011

Cruz Villalon, G. (2023). Characterization of Chromatographic Peaks with Excel. *Journal of Chemical Education*, *100*(2), 928–932. https://doi.org/10.1021/acs.jchemed.2c00588

Felinger, A., & Káré, M. (2004). Wavelet Analysis of the Baseline Noise in HPLC. *Chemometrics and Intelligent Laboratory Systems*, *72*(2), 225–232. https://doi.org/10.1016/j.chemolab.2004.01.018

Jansen, B. C., Hafkenscheid, L., Bondt, A., Gardner, R. A., Hendel, J. L., Wuhrer, M., & Spencer, D. I. R. (2018). HappyTools: A Software for High-Throughput HPLC Data Processing and Quantitation. *PLOS ONE*, *13*(7), e0200280. https://doi.org/10.1371/journal.pone.0200280

Kaplitz, A. S., Kresge, G. A., Selover, B., Horvat, L., Franklin, E. G., Godinho, J. M., Grinias, K. M., Foster, S. W., Davis, J. J., & Grinias, J. P. (2020). High-Throughput and Ultrafast Liquid Chromatography. *Analytical Chemistry*, *92*(1), 67–84. https://doi.org/10.1021/acs.analchem.9b04713

Macko, T., & Berek, D. (2001). Pressure Effects in Hplc: Influence of Pressure and Pressure Changes on Peak Shape, Base Line, and Retention Volume in Hplc Separations. *Journal of Liquid Chromatography & Related Technologies*, *24*(9), 1275–1293. https://doi.org/10.1081/JLC-100103447

Mecozzi, M. (2014). A Polynomial Curve Fitting Method for Baseline Drift Correction in the Chromatographic Analysis of Hydrocarbons in Environmental Samples. *APCBEE Procedia*, *10*, 2–6. https://doi.org/10.1016/j.apcbee.2014.10.003

Moosavi, S. M., & Ghassabian, S. (2018). Linearity of Calibration Curves for Analytical Methods: A Review of Criteria for Assessment of Method Reliability. In *Calibration and Validation of Analytical Methods - A Sampling of Current Approaches*. IntechOpen. https://doi.org/10.5772/intechopen.72932

Morháč, M., & Matoušek, V. (2008). Peak Clipping Algorithms for Background Estimation in Spectroscopic Data. *Applied Spectroscopy*, *62*(1), 91–106. https://doi.org/10.1366/000370208783412762