Appia: a web interface for chromatography

Richard Posert[1,2], Kimberly Hartfield[2], Isabelle Baconguis[2,*]

1. Department of Chemical Physiology and Biochemistry, Oregon Health & Science University, 3181 SW Sam Jackson Park Road, Portland, Oregon 97239, USA.

2. Vollum Institute, Oregon Health & Science University, 3181 SW Sam Jackson Park Road, Portland, Oregon 97239, USA.

* Correspondence to Isabelle Baconguis: bacongui@ohsu.edu

Abstract

Chromatography is used ubiquitously in biology and chemistry to prepare, purify, and assess a wide variety of samples. Software written by chromatography instrument manufacturers is proprietary, which limits its use to licensed computers, and complex, which increases the burden of cognitive load on the user. Here we present Appia, a free, open-source chromatography processing and visualization package focused on making analysis, collaboration, and publication quick, easy, and aesthetically appealing.

Main

Chromatography is an essential and nearly omnipresent technique in molecular biology, biochemistry, and synthetic chemistry. This extremely versatile technique can be used, to name only a few applications, to identify molecules, assay for sample stability and approximate size, determine fluorescence parameters, and purify material. Basic chromatography readouts are peak size, peak shape, and retention time or volume (peak "position"). Most researchers need only this simple information to interpret and communicate their results[1]. To streamline analysis of simple chromatograms, we present Appia, a free, open-source system written in Python for molecular biology and chemistry labs. Appia speeds and simplifies analysis of chromatography data by removing three major barriers: cognitive load, a lack of support for secondary analysis, and difficulty in sharing data.

Underlying all chromatographic assays system is the same fundamental mode of action (Figure 1). Samples are separated by some means, typically involving flowing sample in solvent of some kind over a column. Sample leaving the column is then detected by some property (e.g., UV absorbance, fluorescence, or radioactivity) and optionally collected into fractions which can be further analyzed. It is conventional in the modern laboratory for this process to be automated using instrument control software provided by the instrument manufacturer. The output is then formatted for display and analysis within the same software package or by exporting the data to a separate plotting software. The former option presents certain barriers, such as physical access restrictions (for example, if the instrument belongs to a core facility or collaborator or if a busy instrument is often unavailable for data analysis) and data management inconveniences (such as local data storage on multiple instrument control computers making it difficult to compare data).

Appia was developed to accommodate multi-instrument workflows and provide a consistent look and feel for chromatographic results from preparative and analytical chromatography, with a focus on the needs of the protein biochemistry lab. For example, we regularly purify protein using Size Exclusion Chromatography (SEC), then further analyze samples using Fluorescence-detection SEC (FSEC)[1]. The two techniques are performed on different specialized instruments with mutually incomprehensible output formats. In order to harmonize the displays for efficient analysis and communication of results, it was necessary to export each data set, import them to plotting software, and grapple with the display parameters which were no longer specialized for chromatography data. Appia was developed to consolidate into a single user-friendly interface the chromatographs from the preparative FPLC and the various analytical chromatographs which may have been used to characterize homogeneity, thermostability, protein-protein interaction, or other biochemical parameters of interest. Despite this focus, Appia can be used to analyze and visualize any chromatography data due to the shared structure of chromatograms. Adding unsupported manufacturer's devices requires only writing a method of processing the output data into Appia's very simple internal tables of Retention Volume, Signal, and Channel.

Even without the specific challenges associated with multiple chromatography instruments, Appia was designed for improved user experience by facilitating access to chromatography results with a simple and clean interface. Full-featured manufacturer software packages tend to use complicated interfaces in order to present specialized tools to general users. For example, while some workflows have critical need for peak detection and integration or advanced baseline subtraction, in many cases largely qualitative comparisons—taller or broader peaks; shifted retention times; or simply results consistent with the previous experiment—are sufficient for rapid evaluation of experimental results. Consequently, what could be a rapid comparison instead becomes an exercise in menu navigation. This is not merely a

matter of preference --- attempting to integrate data from multiple complex, physically distant sources accessed under uncomfortable conditions (i.e., wearing PPE to use an instrument computer) has repeatedly been shown to be more difficult than the same task performed in a simple, integrated interface accessed in a comfortable environment[2-6].

Appia simplifies the user experience by representing all data in the same way and presenting the data in a modern, interactive, browser-based interface (Appia Web; Figure 2, Supplementary Video, Supplementary Note) built with Plotly Dash. A clean layout with minimal distraction reduces costs associated with switching between data analysis and software manipulation, especially if the user has to analyze data from different chromatography systems or identical systems which do not share a database. Additionally, data can be analyzed from the user's own desk, removing the requirement for personal protective equipment (which would be required at any instrument computer) and reducing instrument scheduling conflicts resulting from the requirement that data is analyzed and viewed at the acquisition machine.

Data from any number of instruments is presented as a set of plots. Channels (e.g., UV absorbance, radioactivity, % Buffer B) are faceted out in the plots, with a separate plot for fractionated chromatography (Figure 2A, B). The channels are also normalized, useful for assessing analyte concentration and heterogeneity respectively (Figure 2C). With manufacturer software, peak height and area comparison requires integration, which takes time and training to set up properly and may fail in the case of a poorly-resolved shoulder. With Appia, users can zoom into a region on the un-normalized plot and re-normalize each trace locally. This enables visual and approximate numeric comparison of position and monodispersity of peaks even when they are not the highest peak in their respective chromatograms (Figure 2D).

Appia also provides researchers with a variety of simple ways to share their data. Eventually, no matter the system, users will have to export their data in some form to present or publish. Most manufacturers allow for export of some kind of pre-formatted report, but for cohesive publication-quality styling users export the raw data and process it in their plot building software of choice. Appia combines output files and generates simple, appealing default plots automatically during processing, all with a single click. Moreover, Appia includes optional manual R scripts for users familiar with ggplot2 to fine tune the outputs. Even users who prefer other plot-building software still benefit from Appia's data processing pipeline thanks to the plain-text format of Appia's data.

Once the data has been processed, Appia Web lets researchers analyze data at their desk, download plots at home for their grant application, and share links to exciting results directly with their collaborators. Of course, Appia Web can also be installed behind a firewall and/or login screen, or with an entirely separate database for each user when data security is of utmost concern. Appia Web produces consistent, appealing plots which can be saved as .png images with a single click, ready for inclusion in reports, presentations, or manuscripts. Appia can therefore take data from first-impression analysis, through comparison and collaboration, and all the way to publication, all using the same interface.

Appia is designed to be lightweight and approachable. To achieve these aims, it necessarily loses a great deal of advanced analytical power, and so is not a complete replacement for manufacturer-provided software. Rather, the ease of viewing, comparing, and sharing chromatograms allows for faster and easier day-to-day chromatograph. Moreover, pharmacology cores could use Appia to quickly and easily share data directly with core users via URL. Appia's speed and versatility make it ideal for early stages of

a project, and the clear and clean aesthetic of its plots make them strong candidates for final publication figures. If users prefer their own method of generating publication-quality figures, the built-in data standardization provided by Appia accelerates that process as well. Ultimately, Appia aims to focus the user on their data, yielding quicker insights with less time spent at the computer.

Methods

Appia is currently written to analyze data from Waters, Shimadzu, and Agilent HPLCs and AKTA FPLCs, and is under active development at https://github.com/PlethoraChutney/Appia. Support of a new chromatography manufacturer requires writing a parser to convert the manufacturer export format to the Appia format; this typically takes between one and two hours for someone moderately experienced in python, and only has to be done once per manufacturer. Requests for additional manufacturer support are very welcome, and require only submission of some basic information as well as a few representative chromatogram export files. The web user interface (UI) is optional, but highly recommended. It requires installation of both Appia and an Apache CouchDB database, along with some basic networking. However, up-to-date Docker images for the web UI are available at https://hub.docker.com/repository/docker/plethorachutney/appia, and the GitHub repo contains a Docker Compose YML template. Docker Compose automatically builds both the Appia web UI and the CouchDB database (discussed below) and appropriately networks them together, meaning that installation can be performed with minimal technical expertise.

Use of Appia is divided into two modes: processing and visualization. Both of these modes can occur at any computer with a local installation of Appia. Appia Web requires only a browser, no additional software is required at the user's end. Processing involves the conversion of exported manufacturer-format data files into the Appia format (discussed in detail below) and can be performed using a command line interface (CLI), or a graphical user interface (GUI). Visualization encompasses producing static or interactive plots of data, either single chromatograms or combinations thereof.

To produce Appia data files, raw exported chromatograms from each chromatography manufacturer are first provided by the user. Each manufacturer reports the chromatogram using different columns and data formats, part of the original justification for the development of Appia. Appia uses a "parser" to read the sample name, sample set name, channel type, retention time or volume, flow rate, and signal from these disparate file formats and converts that information into a consistent, simple data frame using the python Pandas library. Some manufacturer do not export data files with all the information necessary for Appia to reconstruct a complete chromatogram (e.g., sample name or flow rate). In these cases, the user is prompted to provide that information during processing. The raw files are moved into their own subdirectories, and the HPLC and FPLC data frames saved as .csv files in long (one row per observation, multiple rows per sample) and wide (one row per sample) format tables.

Appia then makes an Experiment, which is the internal representation for a collection of chromatograms. When multiple files are processed at the same time, they are all grouped into an Experiment, named explicitly by the user or implicitly by file metadata. Data processed later can be added to an existing experiment by user specification. Appia Experiments are not otherwise manipulated by the user directly. Rather, Appia uses them to store and work with chromatograms.

Experiments contain the methods for combining, re-normalizing, and plotting the underlying data. If the user provides a username, password, and host for a CouchDB database, Experiments are uploaded to the database during processing; they otherwise are not saved anywhere other than the .csv files. By default, all Appia Web Experiments (but not the saved processed .csv files or raw chromatograms) are downsampled to 1000 points per channel per injection to reduce bandwidth requirements. This temporal downsampling can be modified or disabled by the user. Appia provides wrappers around common CouchDB operations, such as listing the existing experiments and inspecting their contents, downloading them as .csv files, or deleting them. Just as with initial processing, these commands can be accessed in the GUI or the CLI.

For basic non-interactive visualization, Appia can save default plots as PDF files for quick inspection. These plots are moderately customizable through options selected during processing. Templates for manual production of fine-tuned and aesthetically-customized static plots using R and ggplot2 are bundled with Appia and can be copied to the data directory during processing. Users can write their own manual templates as well, which Appia can then copy during processing.

The web UI provides users with a searchable list of all Experiments in the database. The web UI uses a plotly dash frontend to display data pulled from the CouchDB. It is a lightweight application, easily served by the included python script (used in the Docker images). Data is stored and transmitted in a semi-wide format with Sample Name pivoted out into columns to minimize storage and bandwidth requirements. Once loaded, the plots are responsive even on underpowered machines, including mobile phones.

When users select an Experiment, the URL of the web view changes to reflect the selection. The Experiment is loaded and plots produced as follows. HPLC data is plotted in two separate plots, one each for normalized and unnormalized data. Each of these plots has channels separated into facets. Retention time or volume (user-selected) is plotted on the x-axis, while signal for the respective channel is plotted on the y-axis. Samples are separated by color. For Experiments with a single FPLC chromatogram, retention volume and mAU are plotted on the x- and y-axes respectively. Fractions appear in the figure legend if present. The user can click these legend entries to turn on and off a fill from the mAU curve down to the x-axis for the selected fractions, indicating which regions of the trace belong to which of the selected fractions by color. To avoid ambiguity and complexity, Experiments with two or more FPLC chromatograms do not display fraction fills. Instead, a normalized view is added to ease comparison of peak shape and position. Hovering over any plot gives the exact x and y coordinates for the nearest point, as well as fraction information for FPLC traces. Users can select multiple Experiments in the web UI to create a temporary combined Experiment for comparison. When multiple Experiments are combined, the URL path changes to reflect the complete list, separated by "+". When the selected Experiments contain multiple sets of HPLC or FPLC data, all sample names are prepended by their sample set name to prevent sample name collisions.

When users zoom in on a particular region of the unnormalized HPLC data, the URL query string updates to indicate the zoomed region, in minutes (e.g., "?view-range=2.0-5.0" shows the region spanning retention times between two and five minutes). If the user then clicks "Renormalize HPLC", the normalized plots will be re-normalized over the indicated view range, rather than the over the complete trace. That is, by default the normalized traces have a minimum of 0 and a maximum of 1, with linear scale between. When they are renormalized, they have a minimum of 0 globally and a maximum of 1 over the indicated range, meaning the complete trace may have a maximum greater than 1. When the

chromatogram is re-normalized, a second URL query string is updated to reflect the normalization range (e.g., "?norm-range=2.0-10.0" scales samples such that 0 is the global minimum, and 1 is the maximum between retention times of two and ten minutes). Renormalization can be reversed by clicking "Reset Normalization". The entire plot (normalization and view) can be reset by clicking "Reset HPLC". URL modification allows users to share specific combinations of samples, zoomed and normalized over specific ranges, with a single link, reducing barriers to collaborative analysis of chromatography data.

Figures

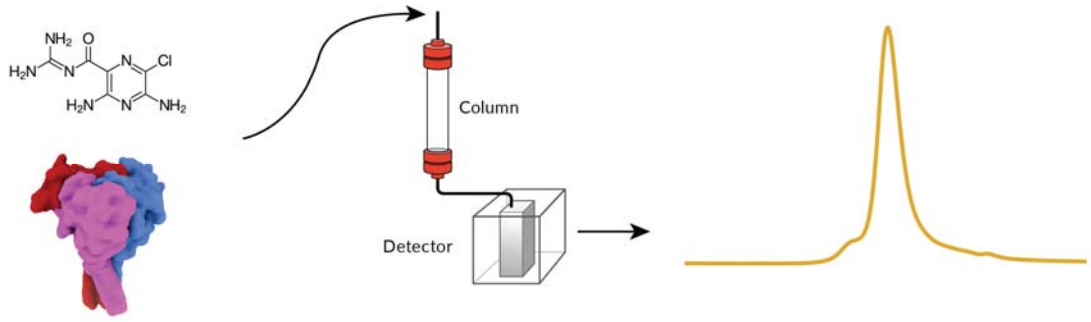

Figure 1. Chromatography schematic. Samples are injected onto a column. As material elutes from the column, some signal is recorded. From the shape and position of this peak, various properties of the sample can be inferred, based on the specific technique used.

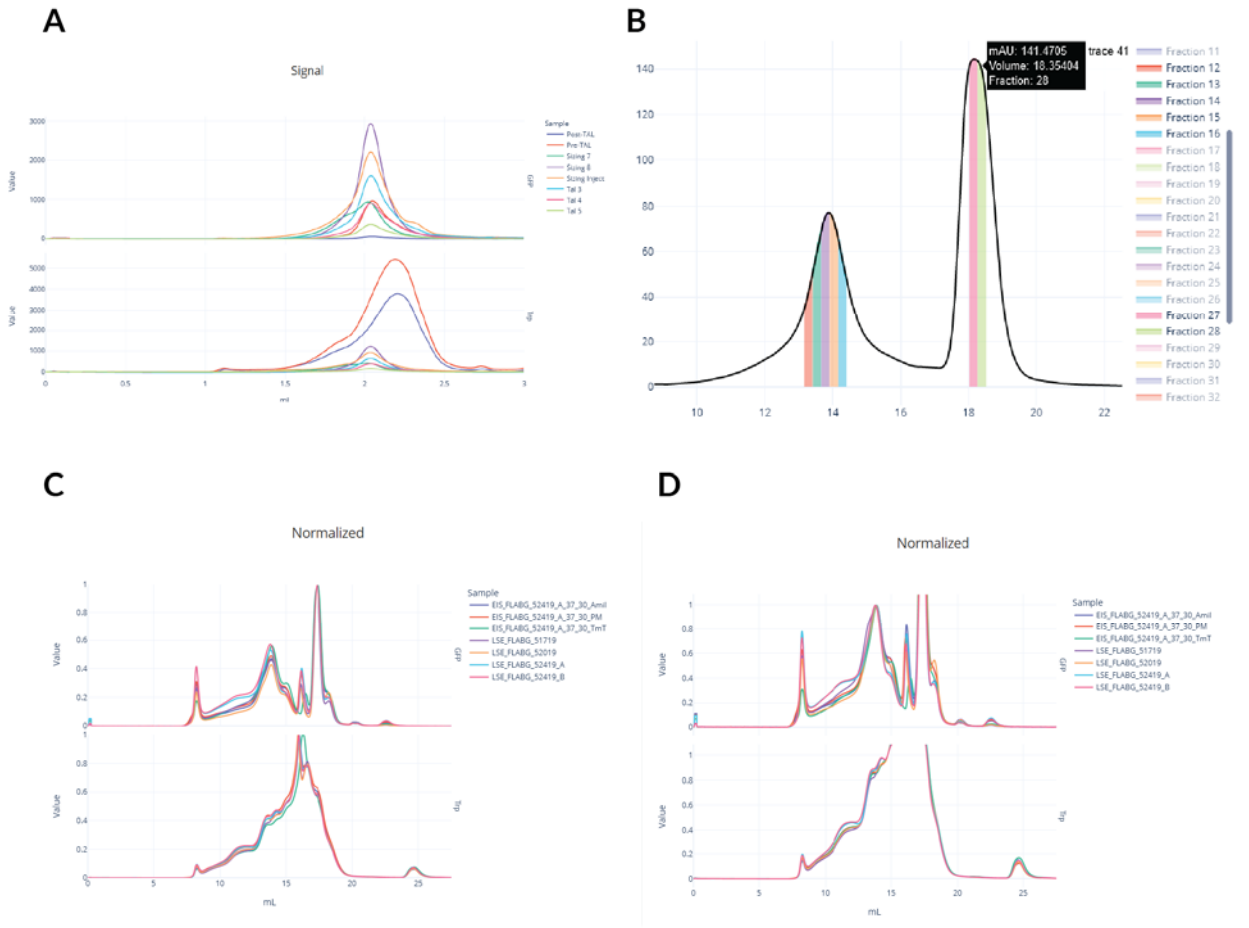

Figure 2. Example Appia plots. A: non-fractionated chromatography displays a separate plot for each channel. Samples can be selected and de-selected by clicking their names in the figure legend. B: Fractionated chromatography. User-selected fractions are highlighted by fill. Hovering over the trace gives more detailed information. C-D: Renormalization. Normalized traces (C) are useful for comparing peak position and shape, but are sometimes made more complicated by large peaks which are not the subject of analysis. Re-normalizing over the region of interest (D) facilitates comparison of specific peaks, regardless of what else is present in the chromatogram.

Supplementary Note

A live demo of the Appia web interface is available at traces.baconguislab.com

Acknowledgements

We are extremely thankful to the OHSU Medicinal Chemistry Core for providing sample Agilent data during development, and for helpful comments and discussion. Thanks also to Eric Gouaux and Steve Mansoor for Shimadzu data. We would also like to thank the lively community of python, plotly, R, and

ggplot users. Without the wealth of knowledge and help provided by these communities, Appia could never have been written. Finally, we would like to thank Sigrid Noreng, Arpita Bharadwaj, and Alex Houser, and James Cahill for valuable feature suggestions and frequent bug reports. This work was supported by the National Institute of General Medical Sciences (T32GM071338, initially to support RP, and R01GM138862 to IB).

Author Information

Affiliations

Department of Biochemistry and Molecular Biology, Oregon Health & Science University, United States

R.P.

Vollum Institute, Oregon Health & Science University, United States

R.P., K.H., I.B.

Contributions

R.P. conceived, wrote, and developed Appia. R.P., K.H., and I.B. wrote the manuscript.

Corresponding Author

Correspondence to Isabelle Baconguis.

Code Availability

Source code is available at https://github.com/PlethoraChutney/Appia under the MIT license.

References

1 Mallik, B., Chakravarti, B. & Chakravarti, D. N. Overview of chromatography. *Current Protocols Essential Laboratory Techniques*, 6.1. 1-6.1. 22 (2008).

2 Monsell, S. Task switching. *Trends in Cognitive Sciences* **7**, 134-140, doi:https://doi.org/10.1016/S1364-6613(03)00028-7 (2003).

3 Sweller, J. Element Interactivity and Intrinsic, Extraneous, and Germane Cognitive Load. *Educational Psychology Review* **22**, 123-138, doi:10.1007/s10648-010-9128-5 (2010).

4 Sweller, J., Chandler, P., Tierney, P. & Cooper, M. Cognitive load as a factor in the structuring of technical material. *Journal of experimental psychology: general* **119**, 176 (1990).

5 Sweller, J., van Merrienboer, J. J. G. & Paas, F. G. W. C. Cognitive Architecture and Instructional Design. *Educational Psychology Review* **10**, 251-296, doi:10.1023/A:1022193728205 (1998).

6 Sweller, J. Cognitive load during problem solving: Effects on learning. *Cognitive science* **12**, 257-285 (1988).