NumFocus - Julia Google Summer of Code proposal: Sputink, a modern tool for data exploration based on JuliaDB and WebIO

Pietro Vertechi, mentored by Shashi Gowda and Josh Day

March 27, 2018

Introduction

The recent JuliaDB package implements effective data wrangling algorithms on large datasets, potentially stored across different processors.

The package is complemented by a set of plotting recipes based on OnlineStats.jl as well as a macro from StatPlots.jl to simplify statistical visualizations of the data: `http://juliadb.org/latest/api/plotting.html`.

I plan to build a web app, tentatively called Sputnik.jl, to allow users to access algorithms from JuliaDB, OnlineStats and StatPlots (also incorporating some of my prior work - GroupedErrors.jl for analysis of population data) from a friendly user interface.

While a web app will never grant the same flexibility as coding a Julia script, I believe it has the following two advantages:

- It is more inviting for users not very comfortable with coding.
- It simplifies completely exploratory data analysis on a dataset with a large number of columns where doing all plots by hand would be too time consuming.

I intend to integrate ideas from my previous experience building a QML-based GUI for data visualization: PlugAndPlot.jl. Despite having enjoyed the flexibility and features of QML while developing PlugAndPlot, this time I'd prefer to focus on a web app (based on WebIO) as it can be easily deployed:

- On the plot pane in Juno (a popular Julia IDE)
- In a Jupyter notebook
- In an electron window
- Served in the browser

If the time allows it, I'd like to investigate whether it is feasible - for users whose data is stored on a server - to deploy such app from the server and analyze the data remotely. In that way, researchers who are willing to open-source their data could quickly set up a website where everybody can consult their data interactively (in my view, this is an excellent way to accompany a publication where only some analyses of the data are accessible).

Plan

As a first step, I inted to port PlugAndPlot.jl from QML to WebIO. This will involve adding some functionality to the WebIO, InteractNext, CSSUtils stack as not all widgets and features of QML are implemented there yet. It will also be a learning opportunity for me as, despite having some experience with traditional GUI toolkits (Gtk and QML), I'm not as familiar with the recently developed WebIO stack. I will be under @shashi mentoring who is one of the main developer of the WebIO stack and will help me familiarize myself with this software. I've already ported some of the functionality, see this video for a demo.

As a second step, I intend to optimize the analytical core of PlugAndPlot, GroupedErrors.jl, a package which accepts any table that can iterate data, in the case where the input data is a JuliaDB table. This should be possible without sacrificing the "iterator based interface", and relies on a set of PRs (on which I'm currently working) to collect iterators as a set of columns efficiently, see:
`https://github.com/JuliaComputing/IndexedTables.jl/pull/137` and
`https://github.com/JuliaComputing/IndexedTables.jl/pull/135`.

As a third step, I intend to rethink the UI design, adding features specific to JuliaDB (such as the powerful set of online statistical analysis and visualizations - or the TableView.jl package, also WebIO based, to visualize the data in a spreadsheet format), incorporating feedback from the JuliaDB developers. For extra flexibility, I intend to add a textbox where users can type in calls to functions from

JuliaDB or JuliaDBMeta to do some preprocessing on their data before visualizing it.

Throughout this process, I will try, as much as possible, to keep the components of the app modularized so that it would also be possible, for a user, to recombine these components to build GUIs with a different design or calling different algorithms and visualization techniques in the background.

Finally, if time permits, I will investigate whether it is feasible to deploy this app from a server where the data is stored, thus simplifying online interactive visualizations of shared data.

Timeline

An approximate planned timeline is as follows:

Week 1	Port PlugAndPlot to WebIO+InteractNext (those parts where the functionality exists already)
Week 2-3	Add missing functionality to InteractNext
Week 4-5	Specialize GroupedErrors for JuliaDB input
Week 6-9	Rethink UI, incorporating OnlineStats and TableView
Week 10-12	Modularize design to allow different parts of the GUI to be reused by different projects
Stretch goal	Test and deploy from a server for online interactive visualizations

About me

Even though my background (bachelor and master) is in mathematics, I'm currently enrolled in a PhD in neuroscience.

For my work I enjoy using open source tools, particularly the Julia programming language to model and analyze data and the Godot game engine to design videogames to be used as psychophysical tasks in a lab environment or online.

My work using Godot or developing Julia can be found at my GitHub account:
`https://github.com/piever`

I've recently started a blog about Godot and Julia, which so far includes a series of introductory blog posts to the Julia language for beginners: `https://piever.github.io/simpleblog/`.

I've chosen to apply for this project (a GUI to simplify data visualization) as I've seen in my research institution that many researchers in neuroscience, especially during master or early PhD, struggle to analyze their data and often resort to proprietary graphical tools that are quite limiting, in that once the user becomes more confident there is no way to "dig deeper". By building a app based on JuliaDB I hope that the beginner user will gradually learn how to translate a sequence of clicks on the GUI into a JuliaDB command. I also think Julia is reaching a stage where it is stable enough also for inexperienced user and it is therefore a good moment to build tools that also target them as a potential audience.

My work in a neurocience institute has also tought me what kind of analysis are more often needed and what type of software is considered intuitive or unintuitive by researchers in this field.

For further information feel free to contact me at pietro.vertechi@neuro.fchampalimaud.org or, if you have a suggestion about this new project, feel free to open a "feature request" issue at this repository.