Julia Programming Workshop

ACEMS Retreat, Nov, 2017

Prof Matthew Roughan

University of Adelaide

Intro

Julia is a free, open-source, programming language that is portable (it runs on Linux, Mac and Windows, at least), and it is very high-performance for numerical coding.

It is a high-level, imperative, procedural language, much like Matlab or R in many respects, but superior in to them in performance for many tasks.

It has many other advantages (over Matlab at least), and has performance approaching that of C.

Pre-workshop tasks

The expectation is that partipants will have their own laptop, as much as is possible, as we won't have significant computing facilities at the hotel. And we might not have good Internet connections at the hotel, so we don't want to be doing big downloads there. Thus, we will need people to do a few things before attending.

This is a short list of things to do before you get to the workshop, in order to make the limited time we have most productive.

1. Download and install Julia. Platform specific instructions can be found at
   https://julialang.org/downloads/platform.html
   Its pretty simply these days. The version we will be using is 0.6.x. I will be using the simple default of a command-line version of Julia.

   You could also try JuliaPro, but I have little experience with this yet. It includes "Juno", and atom-based IDE which should be nice, but you need to register to get this version (it's still free).
   https://juliacomputing.com/products/juliapro.html

2. Install the following packages: DataFrames, DataStructures, Distributions, JSON, and Iterators.
   This is easily accomplished by opening Julia and typing

```
Pkg.update()
Pkg.add("DataFrames")
Pkg.add("DataStructures")
Pkg.add("Distributions")
Pkg.add("JSON")
Pkg.add("Iterators")
```

   Note that this will download the packages, and their dependencies, and so it would be far preferable to do it BEFORE you arrive at the retreat, as it can take some time, even with a decent Internet connection.

3. Install the graphics package PyPlot. In principle this can be done as with any other package, e.g.,

```
Pkg.update()
Pkg.add("PyPlot")
```

However, I have found installing the graphics packages to be a little finicky on my Mac laptop. I can't advise you on all systems, but the underlying issues seem to be

- PyPlot mimics Matlab's plotting routines by calling Python's `matplotlib`. Hence, you must have a working install of Python, with `matplotlib` installed. This isn't too hard, but on the Mac's you have to make sure Julia knows which install to use. You can set this by (in Julia) typing (note use your path to python):

```
ENV["PYTHON"]="/usr/local/bin/python2.7"
Pkg.build("PyCall")
```

- You need to have the Mac command-line tools installed.
- You may need to install XQuartz

As noted, there are many online guides that can help with the exact details for your particular system.

JuliaPro has PyPlot installed by default, but I found I had to update it (using `Pkg.update()`) before it worked.

As a last resort, there are other plotting packages available that may be easier to install, for instance Plotly, but these use somewhat different commands.

4. Make sure you have a text editor installed, for instance Notepad++, Vi, Emacs, …
If you install JuliaPro, it is based around the Atom text editor/IDE, and so you won't need to do this step.

5. It wouldn't hurt to have a play with Julia before you arrive. Its syntax is highly reminiscent of Matlab, so it is pretty easy to get going.

Schedule:

Part I – Julia for Matlab Users

We have only a little time (1 hour) to learn Julia basics, and so it will be a lot easier by comparison to your existing knowledge of Matlab. If you don't know Matlab, you probably have an extra step to add to the previous list :)

Main references:

- https://learnxinyminutes.com/docs/julia/
- https://docs.julialang.org/en/release-0.6/manual/noteworthy-differences/
- https://cheatsheets.quantecon.org/
- https://docs.julialang.org/en/stable/

h all applicable ethical standards of their home institution(s), including, but not limited to privacy policies and policies on experiments involving humans. Note that submitting research for approval by one's institution's ethics review body is necessary, but not sufficient – in cases where the PC has concerns about the ethics of the work in a submission, the PC will have its own discussion of the ethics of that work. The PC takes a broad view of wha

Part II – Julia for Data Sciences

The second component of the workshop will look at some of the facilities that Julia provides for data science.

Topics:

Assuming we have time I intend to talk about

- Plotting and exploratory data analysis
- Some useful packages
- Parallel processing features

Main References:

This component will be (very loosely) based on the book *Julia for Data Science*, Z.Voulgaris, Technics Publications, 2016. However, I won't expect participants to have access to this book.

Other refs

- https://docs.julialang.org/en/release-0.6/manual/performance-tips/#man-performance-tips-1