Setup Julia + Jupyter notebooks

Bruno Amorim

7th April 2021

1 Install Julia

1.1 Download Julia

1. Go to the Downloads page of the Julia official page: https://julialang.org/downloads/

2. Download the latest version (currently version 1.6.0) of Julia for your plataform:

Current stable release: v1.6.0 (March 24, 2021)

Checksums for this release are available in both MD5 and SHA256 formats.

Windows [help]	64-bit (installer), 64-bit (portable)	32-bit (installer), 32-bit (portable)	
macOS [help]	64-bit		
Generic Linux on x86 [help]	64-bit (GPG), 64-bit (musl)[1] (GPG)	32-bit (GPG)	
Generic Linux on ARM [help]	64-bit (AArch64) (GPG)		
Generic Linux on PowerPC [help]	64-bit (little endian) (GPG)		
Generic FreeBSD on x86 [help]	64-bit (GPG)		
Source	Tarball (GPG)	Tarball with dependencies (GPG)	GitHub

1.2 Install Julia

After downloading the Julia binaries, install them. General plataform specific instructions can be found here: https://julialang.org/downloads/platform/.

In the following, I provide some instructions on how to install Julia.

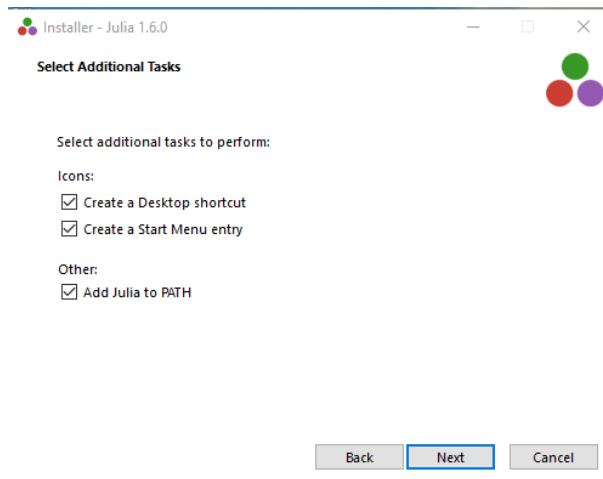#### 1.2.1 Install Julia on Linux

On Linux, the downloaded Julia binary is a compressed `tar` as file.

For Julia 1.6.0 the file name is `julia-1.6.0-linux-x86_64.tar.gz` .

Installing Julia is just a matter of extracting the tar file into some chosen location. I recommend following the steps:

1. Create a folder named `julia` on your personal directory. For example, create the folder `/home/username/julia/` (my username is `bamorim`, so the folder name would be `/home/bamorim/julia/`).

2. Extract the downloaded file into `/home/username/julia/`. This creates a new folder `/home/username/julia/julia-1.6.0`

In the folder `/home/username/julia/julia-1.6.0/bin/` you have an executable named `julia`. To start Julia, you simply run that executable in the command line.

To avoid having to type this every single time, you can edit the `.bashrc` file. This is an hidden file located on the personal folder: `/home/bamorim/.bashrc`. Open this file in a text editor and add the line:

`export PATH="$PATH:/home/username/julia/julia-1.6.0/bin"`

or

`alias julia="/home/username/julia/julia-1.6.0/bin/julia"`

Then, open a terminal and run `bash`.

After this, you can start Julia by simply running `julia` in a terminal.

1.2.2 Install Julia on Windows

The downloaded file is an installer. For Julia 1.6.0, the file is named `julia-1.6.0-win64` To install Julia you simply have to run the installer and follow the instructions. At a certain point, the option to add Julia to the PATH is available. Make sure to select it:

To run Julia, you can simply use the start menu, use the created desktop icon or by running `julia` in a command line (provided you added the Julia PATH during installation).

1.2.3   Install Julia on MacOS

The downloaded file in a `.dmg` file. For Julia 1.6.0, the file is named `julia-1.6.0-mac64.dmg`, that contains a `.app` file `Julia-1.6.app`. Move this file to the Applications Folder.

You can now run Julia from the application launcher.

To launch Julia from the terminal, you must first add Julia to PATH. To do this, open the terminal and run:

```
rm -f /usr/local/bin/julia
ln -s /Applications/Julia-1.6.app/Contents/Resources/julia/bin/julia /usr/local/bin/julia
```

2   Install Jupyter notebooks

In case you have not Jupyter installed on your system, I recommend doing it to through miniconda (a minimal version of anaconda). To do this:

1. Go to the miniconda downloads page: https://docs.conda.io/en/latest/miniconda.html

2. Download installer of the latest Python version (currently 3.9) for your plataform.

3. Follow the installation instructions for your plataform:

   (a) On Linux: https://conda.io/projects/conda/en/latest/user-guide/install/linux.html

   (b) On Windows: https://conda.io/projects/conda/en/latest/user-guide/install/windows.html

   (c) On MacOS: https://conda.io/projects/conda/en/latest/user-guide/install/macos.html

4. After installing miniconda, install jupyter. Open a terminal and run `conda install -c conda-forge jupyterlab`.

3   Use Julia from Jupyter notebooks

In order to use Julia from a Jupyter notebook, you mus install the IJulia kernell, which enables Jupyter notebooks to talk to the Julia installation. To do this:

1. Start a Julia command line (REPL)

2. Install the IJulia package:

   (a) Run `using Pkg`

   (b) Run `Pkg.add("IJulia")`

   (c) Run `Pkg.build("IJulia")`

After doing this you can use Julia from within jupyter notebooks. To do this: open a terminal and run `jupyter lab`.

4 Enable Julia multithreading from within Jupyter notebooks

In order to use multithreading in Julia, one must add the Julia threads before starting it. With the REPL, this is achieved by starting Julia from the command line by running:

```
julia -t n
```

where `n` is the number of threads. You should use the number of physical cores you want to use. My laptop has 4 physical cores, so I can run:

```
julia -t 4
```

In order to use multithreading in Julia from jupyter, you must install a Julia kernel with the desired number of threads. Following the instructions from https://julialang.github.io/IJulia.jl/stable/manual/installation/, to create a Julia kernel with 4 threads, open the Julia terminal and run:

```
using IJulia
installkernel("Julia (4 threads)", env=Dict("JULIA_NUM_THREADS"=>"4"))
```

After doing this, if you start jupyter, besides the `Julia` kernel, you can select a kernel named `Julia (4 threads)`.