Wholebrain

Contents

Installing Wholebrain on Windows

Tested on Windows 10 64-bit. At the time of this writing (Nov 2018) the version of Wholebrain installed is 0.1.35.

Install R

Tested: Version 3.5.1 Probably works: Latest version Download R (https://cran.r-project.org/bin/windows/base/)

Under Select Components, choose "64-bit User Installation" Under Customize Startup, choose "SDI (separate windows)" [optional]

It should end up here:

```
C:\Program Files\R
```

Install FFTW

Tested: 3.3.5 Probably works: latest version

The Fastest Fourier Transform in The West (FFTW)! Always loved that name.

Download FFTW 3.3.5 (ftp://ftp.fftw.org/pub/fftw/fftw-3.3.5-dll64.zip) Unzip to download folder, then create this folder:

```
C:\fftw
```

And copy 2 files into it:

```
fftw3.h
libfftw3-3.dll
```

We install it here before Rtools, because we want to add FFTW to the System PATH and the Rtools installer has a step where you can manually edit the PATH directly which is a nice convenience.

If you for some reason already installed Rtools and didn't add C:\fftw to the PATH, you can use the instructions in #Update System Path to include R and ROpenCVLite to do it. You will be adding an environment variable in the next step anyway, which requires the same instructions, so should be easy to do.

Add the LIB_FFTW Environment Variable to Windows

Use the instructions in #Update System Path to include R and ROpenCVLite to get to the Environment Variables window. But rather than edit the PATH, an existing environment variable, we will create a new one called LIB_FFTW. Use the New button to create the variable, and then fill in its value: C:\fftw

Install Rtools

Tested: Version 3.4 Probably works: Latest "frozen" version. See table on download page.

Download Rtools (https://cran.r-project.org/bin/windows/Rtools/)

Under Select Components, choose "R toolset, Cygwin DLLs, and R 3.3.x + 64 bit toolchain" not 32-bit!

Install to

C:\Rtools.

Set System Path To include Rtools and FFTW

You will get the option to set the System Path from within the Rtools installer. Check the box. At the top of the list, it should list (add if not) the Rtools bin directories:

```
c:\Rtools\bin
c:\Rtools\mingw_64\bin
c:\fftw
... (leave everything below here alone!)
```

Be sure to add FFTW to the path above while we're at it!

If you forgot or skipped this convenient part of the Rtools installer, you can add those directories to the PATH via #Update System Path to include R and ROpenCVLite

Setting system path here will enable R & RStudio to find the compilers and fftw api when building Wholebrain

Note on setting system PATH

Be careful when changing the system PATH. In particular, do not change any lines below the ones we will add, especially any with %SYSTEM in them! You can render your system unusable by removing or altering %SYSTEM paths. Copy and paste all the text in the PATH window into a text editor such as Notepad.exe and save it, before changing the PATH, if you want to be extra careful.

#Update System Path to include R and ROpenCVLite shows where to find the Environment Variables window.

There is an upper and lower section to the Environment Variables window. The upper part changes user-specific variables, and below is the area for system variables that affect all users. You want to use the bottom pane. It seems safer to use the upper one, but it is not clear whether that would work for our purposes or not, so we will stick with the traditional system PATH setting.

If for some reason you've already launched R or RStudio (next step) you will need to quit and relaunch them. Any time you update variables RStudio needs from the system PATH, restart RStudio. I've changed the PATH to include FFTW while RStudio was running, and it failed to find the FFTW library. It succeeded after being quit and restarted, so the value when it launches is the value it uses (likely true for all programs on Windows.)

We will be updating the system PATH at least 3 times while installing Wholebrain which is the only reason we are looking into time-saving shortcuts. The PATH is how many system-wide services find critical components, the usual advice is to restart your system after changing the PATH. Doing so will ensure that all running processes are using the latest values.

Install RStudio

Tested: 1.1.463 Probably works: Latest version (https://www.rstudio.com/)

RStudio is not strictly necessary, one can launch R and use it from within a DOS-like command window. But installing and using RStudio makes R easier to use and has several conveniences. Also, the author of Wholebrain uses it, so when corresponding with the author or power users about Wholebrain, they will usually assume you are using RStudio when describing how to do things, so it is worth becoming familiar with it if you aren't already!

One can install the devtools conveniently by going into the "Tools" menu, and choosing "Install Packages...", then type "devtools"

RStudio can install Rtools as well, which in my "RStudio first" experiment was exciting, but ultimately unsuccessful. RStudio will install Rtools to C:\RBuildTools not C:\Rtools which Wholebrain expects. It also will install the latest version of Rtools (3.5 as of Nov 2018) not 3.4 which is more stable. Stick to installing Rtools manually rather than from within RStudio.

RStudio should find your 64-bit installation of R. RStudio will ask you where R is, if you didn't install R yet. Quit and install R first if that happens. I tried installing RStudio first, just to see if RStudio includes R. It does not! RStudio asked me to locate R the first time it ran, and operated normally afterwards.

RStudio should end up here:

```
C:\Program Files\RStudio
```

Install CMake

Tested: 3.13.0-rc3 Probably works: Latest version (https://cmake.org/download/)

Install to its default location.

Quit RStudio if you are running it, because we are about to update the system PATH. RStudio will need CMake to compile Wholebrain and the packages it depends upon, and it will look up CMake's location in the system PATH that RStudio stores when it launches.

Update System Path to include CMake

CMake should do this for you but if you forgot to check the box during CMake install, you can add it (or double check) yourself using the instructions for doing it manually #Update System Path to include R and ROpenCVLite

Check the box to update the System PATH ("for all users".) It should add:

```
C:\Program Files\CMake\bin
```

Install Wholebrain Part I: ROpenCVLite

We will go ahead and start the install of Wholebrain, which will download and install a bunch of dependencies. This includes ROpenCVLite which will take a little bit of time. But it is easier than compiling and installing OpenCV yourself as in the original instructions!

Start up RStudio again. If it was running, quit it first.

At the command line on the left, enter

```
devtools::install_github("tractatus/wholebrain", args="--no-multiarch")
```

The args don't seem to matter but we can keep these original args in there. It should download a bunch of dependencies, and then get to ROpenCVLite. It should build this successfully (it will require CMake be installed correctly.) Then it will attempt to install wholebrain, but fail. This is because the ROpenCVLite OpenCV DLLs need to be added to the System PATH, and we didn't know where they were until they existed! (well, we could have guessed I suppose.)

Quit RStudio.

Update System Path to include R and ROpenCVLite

This time we need to update the system PATH manually, no help from installers. I find it this way in Windows 10:

```
This PC -> Right click -> Properties -> Advanced System Settings -> Environment Variables... -> System variables -> [highlight
```

You can also get to the same place as your PC Properties using the Control Panel app:

```
Control Panel -> System and Security -> System
```

Then add the R and ROpenCVLite locations on the end so that the list looks like this (your "%System" lines may differ as will your username and R version perhaps)

Rather than type in the directory, you can use the Browse... button in the Edit window to navigate to its location and choose it that way if you like.

```
c:\Rtools\bin
c:\Rtools\mingw_64\bin
c:\fftw
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\CMake\bin
C:\Program Files\R\R-3.5.1\bin\x64
C:\Users\bdsin\Documents\R\win-library\3.5\ROpenCVLite\opencv\x64\mingw\bin
```

Install Wholebrain Part II: Wholebrain proper

Ensure that you quit RStudio, since the system PATH has changed. Then start it up again. No need to save the workspace each time it asks, btw.

Then hit up-arrow in the console on the left, so that the last thing you typed appears:

```
devtools::install_github("tractatus/wholebrain", args="--no-multiarch")
```

and this time, when it gets to the end and tries to load the wholebrain.dll, it should succeed since the system Path will contain the location of the R and ROpenCVLite libraries it could not find before.

Test Wholebrain

Quit RStudio and then launch it again, so that you are testing a usual future post-install session with Wholebrain. We can test using the same code the author provides in the original instructions:

```
library(wholebrain)
filename<-system.file('sample_tiles/rabiesEGFP.tif', package='wholebrain')
output<-segment(filename)
```

The most important is the first line: can you load the Wholebrain R library without any errors? If so it should be working!

The next two lines will load a file and plot results. Hit the "q" key to close the figures and get output, which should be a two element list. The output as of version 0.1.35 is:

```
OUTPUT SEGMENTED CELLS: 248
```

Differences from the Official Wholebrain Windows Install page

There is a page last updated in 2015 with the install process at Installing Wholebrain On Windows (http://www.wholebrainsoftware.org/cms/installing-wholebrain-on-windows/). These instructions are different in the following ways (as of Nov 14 2018):

- OpenCV is now installed via ROpenCVLite
  - This is per the author. The github DESCRIPTION file that R's install_github command uses to get dependencies includes ROpenCVLite as of October 2018.
  - This makes installing OpenCV much easier than before in original Step 5, the most lengthy and error-prone step by far

    - It also means no need to set LIB_OPENCV environment variable in original Step 7.
- RStan might be useful but is not required for installation. It can be installed later via "install.packages" or RStudio's "Tools->Install Packages..." when the need arises, like any other R package.
    - In other words There is no Step 3! (https://www.youtube.com/watch?v=6uXJlX50Lj8) (installing RStan was original Step 3)
- Newer versions of R, Rtools, and CMake work. I tried latest Rtools (3.5) but encountered an error. The author advised that 3.5 has had issues and recommended version 3.4 which works (original instructions use R 3.3.3 & Rtools 3.3 whereas here I use R 3.5.1 & R 3.4)
- RStudio (free, opensource version) installation as a step. Optional but makes dealing with R better.
- Since the Wholebrain dependencies vary, they are not installed explicitly beforehand
    - Original Step 8 includes some install.packages commands prior to the install_github command but if needed, they will be taken care of during the main install. The actual set of dependencies currently is a different, longer list.

Contacting the author of Wholebrain

The author, Daniel Fürth, is very responsive on gitter and helped with questions related to these instructions. Gitter is an in-browser chat for github projects like Wholebrain and can be accessed here: Wholebrain Gitter Lobby (https://gitter.im/tractatus/Lobby)

[TODO: Adding screenshots!]

Retrieved from "https://npcdocs.princeton.edu/index.php?title=Wholebrain&oldid=147"

- This page was last edited on 14 November 2018, at 17:10.