Spectre development

Overview

The Spectre R package is an open source project, which means out source code is available online (on Github), and users can contribute to creating new functions or improvements for Spectre. If you would like to contribute to Spectre, please adhere to the following guidelines.

Sub-pages

What to work on?

Firstly, look at any outstanding issues in the issue page and see if there's anything there you want to work on. If there isn't any, you can of course create a new issue and describe what is it you want to improve or add to the package.
It's **mandatory** to clearly describe in the issue ticket what is it you want to add/improve. The page is monitored by the core maintainers of the package. Thus, we might put up a comment to start up some discussions on the issue from time to time. Please respond when we do so.

Development protocols

Few guidelines when you decided to work on an issue:

1. Make all the changes on a new branch created from **development branch. DO NOT** apply your changes directly to master or development branch as your changes might impact other parts/features of the package that are otherwise functioning fine.
2. Make sure your changes/additions are properly tested. We recommend creating some unit tests.
3. Make sure you write a clear documentation to any new functions. We use roxygen to generate our documentation. If you are not sure how to write a documentation in a format that is parseable by roxygen, Google is your best friend. There are other existing functions in the package which documentation have been written. You can also refer to them.
4. Make sure you include a vignette in form of sample run script if you are implementing a new feature. This sample run script should lay out the workflow generally applicable to your function. Note only do this if you are adding new feature, generally not required if you are just fixing bugs.
5. When you are ready to merge your changes to our main branch (development branch), create a new pull request in Github and assign either @Thomas Ashhurst @Givanna Haryono Putri (USYD) @Felix Marsh-Wakefield (USYD) as reviewer. **Wait until we've approved your changes before you merge.** If you don't wait, we will revert your merge ourselves.
6. **Do not** run roxygen on your branch ever. This will cause conflict on the documents modified by roxygen (the one we're not supposed to manually edit). We'll run roxygen once when we're about to merge the development branch to the master branch for a new release.
7. **Do not** import any libraries within your function script unless it's a sample workflow script. Library imports must be done by the user in their workflow script.

Install the 'development' version of Spectre

Install and load 'devtools'

Input

```
# Install (if not already installed) and then load devtools
if(!require('devtools')) {install.packages('devtools')}
library('devtools')
```

Load development branch

Input

```
# To install the 'development' branch, for testing new functions, use the following line to install
Spectre
install_github("immunedynamics/spectre", ref = 'development')
```

Install from local copy

If you have made changes locally but decided not to push to Github, you can install spectre from your local copy.

Note before you do this, remove any existing Spectre installation.

Input

```
# Load devtools
library('devtools')

# Set working directory to the local copy of Spectre
setwd('/Users/dev/Spectre')

# Install Spectre
install_local(force = TRUE)
```

Load Spectre library

Input

```
# Once successfully installed, load Spectre
library("Spectre")
```