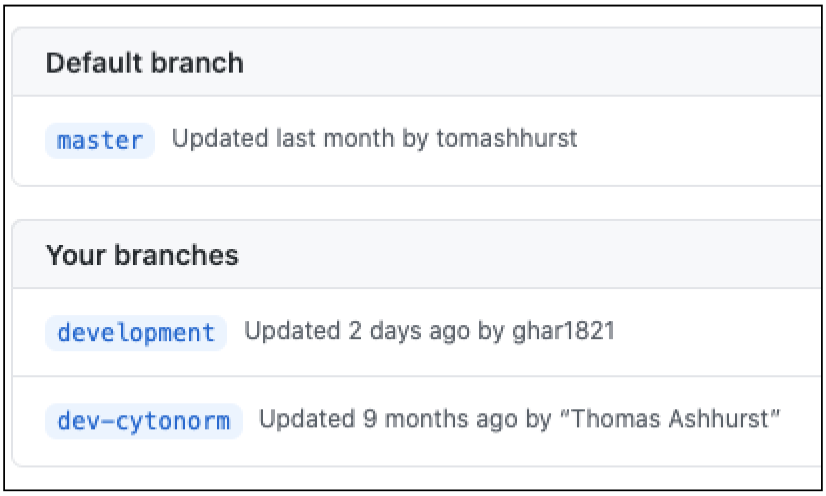
Development versions of Spectre

If you wish to use new or advanced features or Spectre, or access functions from separate development branches (e.g. such as spatial analysis tools), then you can use the instructions below. By default, Spectre's development version can be installed using the 'development' branch (described below). However, there are other developmental branches available. To see the available branches, please visit https://github.com/sydneycytometry/Spectre/branches.



Default branch

master Updated last month by tomashhurst

Your branches

development Updated 2 days ago by ghar1821

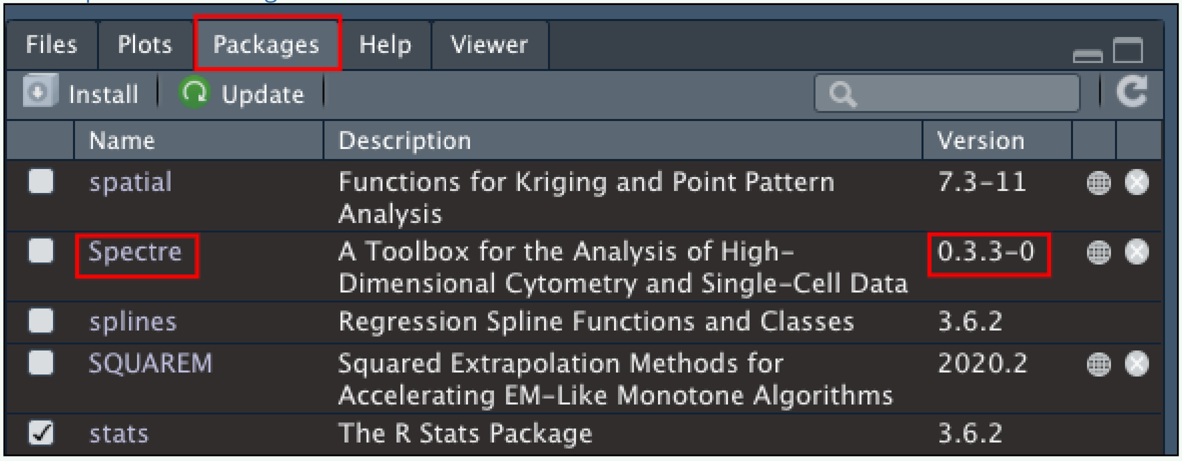
dev-cytonorm Updated 9 months ago by "Thomas Ashhurst"

We recommend only doing this if you are familiar with Spectre's base functionality, as developmental functions may be in various states of testing.

Installing a developmental branch

Installing new, or updating?

To see if you have Spectre installed, you can open **RStudio** and look under the '**packages**' tab. The version will be listed in the right hand column. Hit the ⟳ button to refresh the package list if you need. If you have the latest version installed, R won't install it again. If you have an older version, the R will install the latest version. If you wish, you can check the Spectre Home Page for the latest version.



	Name	Description	Version
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
☐	Spectre	A Toolbox for the Analysis of High-Dimensional Cytometry and Single-Cell Data	0.3.3-0
☐	splines	Regression Spline Functions and Classes	3.6.2
☐	SQUAREM	Squared Extrapolation Methods for Accelerating EM-Like Monotone Algorithms	2020.2
☑	stats	The R Stats Package	3.6.2

If you are updating Spectre, we advise deleting the previous installation to ensure a clean update.

```
remove.packages('Spectre')
```

Install Devtools

In R, install and load the 'devtools' package. The devtools package has a number of dependencies (packages that it needs to use) that it will automatically install when you install 'devtools' – this might take some time. You can then install a specific branch of Spectre – in this case, the 'development' branch.

✓ **Branch**

In the example below, we are installing the 'development' branch. To install another branch, such as the name of that branch. For example, to install the 'spatial' branch, use ref = 'spatial' instead.

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

## Install Spectre
library('devtools')
install_github("immunedynamics/spectre", ref = 'development')
```

Confirm or decline updates

During installation, you may see a message like the following. You can update all the packages if you wish, but if you are unsure, or using multiple packages in other contexts, it might be wise to wait until you have time to update everything and test your scripts afterwards.

```
These packages have more recent versions available.
Which would you like to update?

1: All
2: CRAN packages only
3: None
4: rlang    (0.4.1  -> 0.4.2 ) [CRAN]
5: digest   (0.6.22 -> 0.6.23) [CRAN]
6: roxygen2 (7.0.0  -> 7.0.1 ) [CRAN]

Enter one or more numbers, or an empty line to skip updates:
```

Install additional required packages

When you install Spectre, most of the dependencies will be installed automatically, excluding some packages from BioConductor. You an install these separately by running the following:

```
Spectre::package.install()
```

When you install Spectre, most of the dependencies will be installed automatically, excluding some packages from BioConductor. You an install these separately by running the following:

If required, you can install the Bioconductor packages manually (click to expand)

```
## Install BiocManager to download packages from Bioconductor
    if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager")

## Download additional BioConductor packages
    if(!require('flowCore')) {BiocManager::install('flowCore')}
    if(!require('Biobase')) {BiocManager::install('Biobase')}
    if(!require('flowViz')) {BiocManager::install('flowViz')}
    if(!require('FlowSOM')) {BiocManager::install('FlowSOM')}
```

Load libraries (to check installation)

Once the packages have been installed, you can load the libraries by running the following. Please ensure that each library successfully loads. If any do not load, or are listed as unavailable, please see the troubleshooting section below.

The Spectre package is designed to helps various computational tools work nicely with each other, in a single environment, design for high-dimensional cytometry/single-cell analysis. To do this, many of Spectre's functions are 'wrappers' for other packages (e.g. Spectre has functions to run FlowSOM and UMAP, but these functions use the code from the FlowSOM and UMAP R packages respectively) – ensuring the input and output parameters work in a consistent fashion, and adding modifications to tailor these packages to cytometry analysis. Because of this, Spectre requires the installation of a number of other packages. By default, many of these will be installed ***automatically*** when you install Spectre. However, this can sometimes be a problem if a necessary package attempts to be installed, but for some reason it can't be. In these cases, the error messages informing you that the package was not installed are easily missed.

```
## Check if all required packages have been installed
Spectre::package.check()
```

If installation of each package was successful, then you can proceed to load the other key packages.

✓ **Output: if installation was successful**

Nothing will be returned in R, or message relating to the loading of these packages will be returned.

If installation of any specific package was **not** successful, you will see something like the following. If this occurs, you will need to attempt to install the package directly. See our troubleshooting page for help.

Output: if installation was unsuccessful

Error in library('plyr') : there is no package called ‘plyr’

You can then load the required packages using the 'package.load()' function.

```
## Load all required packages
Spectre::package.load()
```

Installing additional packages for specialised workflows

For some specialised workflows, some additional packages are required. This is especially true for the spatial analysis workflows. To install these, you can utilise the 'package.' functions from Spectre. Currently the specialist areas can be specified as type = 'ml' for machine learning packages, or type = 'spatial' for spatial analysis packages.

For the spatial analysis workflows.

```
Spectre::package.check(type = 'spatial')    # Checks to see if the spatial analysis packages are
installed
Spectre::package.install(type = 'spatial')  # Installs the spatial packages
Spectre::package.load(type = 'spatial')      # Loads the spatial packages
```