

---

# MODELING BIOLOGICAL SYSTEMS

---

## INSTRUCTIONS TP-TD1

### I - The R programming language

Download and install R.

You can find it here, for the OS you use (Windows, Linux or for Mac) : <https://cran.r-project.org/>

### II - Integrated Development Environment

Use an IDE (Integrated development environment = a pretty graphical interface) of your choice to get a more user-friendly programming environment. The R integrated IDE is *RGui*, once R is installed on your computer, you can launch *RGui* to see what it looks like. You can work with *RGui* if you like, however the vast majority of R users work with another IDE called *RStudio*, available to download here (the free version is more than enough for our needs): <https://rstudio.com/products/rstudio/download/>

If you are already familiar with programming, this cheat sheet should be enough for you to understand how to use *RStudio*: <https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf>

(From now on, will assume you are using *RStudio*. It will not change anything for most of them, but it might for some)

Once *RStudio* is installed, launch it, and start a new script. You are now ready to code

### III - Coding

Please note that in the bottom right window, you can see an “Help” tab. This will give you access to almost all the documentation available on R and all the libraries you will be using.

You can also access the help by typing “?” on your console, followed by the name of the library or the function you are seeking help with.

Try it by typing the following on your console:

```
> ?install.packages
```

The “install.packages” function is used to **install** R libraries Try it out with:

```
> install.packages("tidyverse")
```

“tidyverse” is a suite of libraries useful in R, it will install many libraries that you will use later on. When the install is complete, you will see the R prompt “>” on your console again

Now that the library is installed, you need to **call it**, in order to use it:

```
> library(tidyverse)
```

You only must install the libraries once, but you will need to call them again every time you start a new session

You should now see this in your console, which means the package was loaded correctly:

```
-- Attaching packages ----- tidyverse 1.3.0 --  
  
v ggplot2 3.3.2  
v tibble 3.0.3  
v tidyr 1.1.2  
v readr 1.3.1  
-- Conflicts ----- tidyverse_conflicts() -- x  
dplyr::filter() masks stats::filter()  
  
x dplyr::lag() masks stats::lag()  
  
v purrr 0.3.4  
v dplyr 1.0.2  
v stringr 1.4.0  
v forcats 0.5.0
```

It also informs you about conflict, a couple of functions from different libraries have the same name, which will cause problem if you try to use them. If you want to be sure to use the right function, you can use a specific syntax to precise the library:

```
> package_name::function_name()
```

## IV - Tutorials

Today will be all about learning how to code with R. For that, there is nothing better than using R directly.

Install and call the **"swirl"** package on your console. *Swirl* is an excellent tutorial built directly in R. It will help you get the basic concepts of the R language.

Start by adding a few courses you will use today:

```
> install_course("R Programming E")  
> install_course("Getting and Cleaning Data")  
> install_course("Exploratory Data Analysis")
```

Start "swirl" by calling the function:

```
> swirl()
```

You can now choose lessons. Start by doing the entire "R Programming E" lessons (recommendation : 20 minutes maximum), and then go directly to the two other courses "Exploratory Data Analysis" and "Getting and Cleaning Data". Select the sub-lessons that inspire you. You do not have to do them all. Be sure to spend enough time learning about plots, as you will use them a lot in the future.