

*Disclaimer: The procedures described in the following are meant as an example, how to install sybil and sybilSBML on Mac OS 11. Where appropriate, we added links to the original descriptions underlying specific steps. The necessary procedures on your specific platform or version of Mac OS operating system can be different. We give absolutely no warranty, that this procedure will work or that there are no negative consequences of the procedure on your system. We are not claiming, that this instruction is the only and best way to install the software. NEVER use any commands given in the following description, if you are not aware of what you are doing and what the commands exactly do and mean. Only download software from trustworthy locations. Be aware that turning off system protection can make your system unsafe and instable. Always have backups of your system.*

## **1. Installing GNU R for Mac on Mac OS 11 (Big Sur):**

### **Step 1: Installing XQuartz**

Download the latest XQuartz installer image from <https://www.xquartz.org/> and run the installer to install it. In this description, XQuartz version 2.8.1 was used.

### **Step 2: Installing R for Mac**

Download the R installer package from <https://cran.r-project.org/bin/macosx/>. In this description, R-4.1.0.pkg was used.

If GNU R is installed, the following command in the Terminal App should display the R version that had been installed:

```
R --version
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
https://www.gnu.org/licenses/.
```

## 2. Installing the binary package version of sybilSBML on Mac OS 11 (Big Sur):

A binary package version of sybilSBML, that was linked against the static version of the libSBML library can be downloaded [here](#). The source code of libSBML can be found [here](#). The source code of sybilSBML can be obtained from [here](#). Although this installation procedure is probably the easiest, keep in mind, that installing binary packages is always a security risk for your system. It is also possible to install the package from sources (see section 3).

### **Step 1: Installing sybil package**

Start the R environment and type the following command to install the sybil package from the [CRAN](#) repository:

```
install.packages("sybil")
```

### **Step 2: Installing the binary package of sybilSBML for Mac OS**

Download the latest binary package version of sybilSBML [here](#). In this description, sybilSBML\_3.1.3.tgz was used. In R for Mac, type the following command in the R environment (modify example file path shown here for the correct location of the .tgz file):

```
install.packages("/Users/mayo/sybilSBML_3.1.3.tgz", repos =  
NULL)
```

### **Step 3: Loading libraries**

You should now be able to load the libraries by starting R and typing the following commands:

```
library(sybil)  
library(sybilSBML)
```

### 3. Installing the source package version of sybilSBML:

The procedure of installing sybilSBML from source code on Mac OS is more comprehensive, since it is necessary to build and install libSBML from the source distribution, set environment variables and system settings and install sybilSBML from source code. In the following, we will go through the necessary steps to install sybil and sybilSBML on Mac OS 11 (Big Sur). Please make sure, that you adjust individually presented paths according to the situation on your system.

In order to be able to build or install source packages,  $R \geq 4.0.0$  for Mac OS uses Xcode and Fortran (<https://cran.r-project.org/bin/macosx/tools/>).

#### **Step 1: Installing latest Xcode version**

Make sure, you have installed or updated to the latest version of Xcode. In this description, Xcode version 12.5.1 was used.

#### **Step 2: Install GNU Fortran for Mac OS**

Download the disk image of GNU Fortran for Mac OS 11 from [here](#). In this description, gfortran 11.1 for BigSur (macOS 11) Intel ([gfortran-Intel-11.1-BigSur.dmg](#)) was used. Install the package `gfortran.pkg` from the downloaded disk image. You might receive a message, that the package cannot be opened, because it comes from an unidentified developer. In order to open it or to add an exception for the app, refer to the macOS [User Guide](#).

#### **Step 3: Install CMake $\geq 3.10$**

Building libSBML from the source distribution in the next step, requires the installation of the [CMake](#) application. In this description, [cmake.3.21.0-rc2-macos-universal.dmg](#) was used. Drag & Drop the `CMake.app` from the disk image into your Applications folder.

#### **Step 4: Install libSBML from the source distribution of libSBML.**

The following information is based on the procedure described [here](#). We recommend to build libSBML from the source distribution, to prevent problems based on different compilers and / or versions of standard libraries in the linking process.

Download the source distribution of the latest stable version of libSBML from [here](#) and decompress it. In this description [libSBML-5.19.0-core-plus-packages-src.tar.gz](#) was used. You should find a folder called something similar like `libSBML-5.19.0-Source` in the location, where the archive had been decompressed.

Next, open the Cmake application from your applications folder. In the application window, look for the field: “*Where is the source code*” and enter the location folder of your previously decompressed libSBML folder (e.g. `/Users/mayo/libSBML-5.19.0-Source`). Adjust the path according to the situation on your system. In the field “*Where to build the binaries*”, you can enter the location of a directory in which

the binaries will be built (e.g. `/Users/mayo/libSBML-5.19.0-Source/build/`). Adjust the path according to the situation on your system.

You can leave “*Unix Makefiles*” and “*Use default native compilers*” in the fields, which specify the generator for this project.

Click on the *Configure*-button. After configuration, search for the field “*ENABLE\_FBC*”. The checkbox is not checked by default. If you want to be able to use functions depending on the libSBML FBC plugin (recommended), you should check this box. Next, search for the field “*ENABLE\_GROUPS*”. The checkbox is not checked by default. If you want to be able to use functions depending on the libSBML Groups plugin (recommended), you should check this box.

Re-click on the “*Configure*”-button to activate the changes.

Click on the “*Generate*”-button to generate the Makefiles.

Open the *Terminal* application from the applications menu on your Mac. Change directory to the decompressed source folder (adjust the path according to the situation on your system) and enter the following commands, that will install libSBML in the default location `/usr/local`:

```
cd /Users/mayo/libSBML-5.19.0-Source/build
make
sudo make install
```

### **Step 5: Adjust environment variable and system settings to make the dynamic library loader able to find the libSBML dynamic libraries at runtime**

Because of security settings in Mac OS, we have to make some settings, so that the dynamic library loader is able to find the libSBML shared libraries when loading sybilSBML. Otherwise, the process of installing sybilSBML from source code will fail, even if the respective libraries could be found at building time. CRAN suggests to set the environment variable `R_LD_LIBRARY_PATH` (<https://cran.r-project.org/doc/manuals/r-release/R-admin.html#External-software>). Type the following command in your Terminal application:

```
export R_LD_LIBRARY_PATH="`R RHOME`/lib:/usr/local/lib"
```

Another issue comes with the system integrity protection of Mac OS preventing the execution of unauthorized code, and therefore the libSBML shared libraries from loading. Apple explains [here](#), how to disable and enable system integrity protection. A discussion dealing with the issue while installing unrelated software can be found here: <https://github.com/apache/incubator-mxnet/issues/575>.

In order to make it possible to install R packages from source code that link to third party shared libraries on our system, it was necessary to disable the system integrity protection. Otherwise, it was not possible to load the libSBML shared libraries when loading the sybilSBML library, which lead to a failing installation.

### **Step 6: Installing sybil package**

Start the R environment and type the following command:

```
install.packages("sybil")
```

### **Step 7: *Installing sybilSBML***

Download the source code package version [here](#). In the Terminal App, type the following command (adjust example file path given here to the correct location of the .tar.gz file on your system):

```
R CMD install /Users/mayo/sybilSBML_3.1.3.tar.gz
```

### **Step 8: *Loading libraries***

You should now be able to load the libraries by starting R and typing the following commands:

```
library(sybil)  
library(sybilSBML)
```