A basic workflow for using the GMACS repository for developpers

Matthieu VERON

Last compiled on 13 June, 2022

1 Introduction

This document is intended to give you "guidelines and generic steps" to follow when working on GMACS to release a new version. Its aims are to show you how to use this Github repository to modify GMACS version while tracking code changes. The idea here is to make this workflow easy to follow so we will work with Github with no command-line interface using the graphical interface for Git: Github Desktop.

We assume you already know how to use Rstudio with projects. If you have never used Github (and/or Githhub Desktop) with Rstudio, you can see the following nice online workshops: the "Happy Git and Github for the useR" from *Jenny Brian* and the R Workflow workshop from *Elizabeth Holmes*.

2 Set-up for using this workflow

As a member of the [*GMACS-project*] organization, you probably already have a Github account; if not, you will need one if you want to act as an active developer. You will also need to get installed R (or Rstudio) and Github Desktop on your computer. Below are the links to install these programs:

- Install R
- Install Rstudio
- Install Github Desktop

If you want to link your R/Rstudio to your Github account so that you can push your changes from Rstudio directly to Github, please refer po the two workshops listed above. I will not cover this topic in this document.

3 Let's get the GMACS_Assessment_code repository on your computer

I am going to show you a workflow to get the GMACS_Assessment_code repository on your computer. Here, you have two options:

1. First, you want to contribute to the development of GMACS. You will therefore **fork** the repository to be able to directly push up changes to GMACS to the organization's repository.
2. The second one, you only want to **clone** the repository of the organization and modify it to your liking without pushing up these modifications to the repository of the organization. In this case, you will not contribute directly to the development of GMACS.

3.1 Fork the GMACS_Assessment_code repository to your computer

By ***forking*** this repository, you will be able to contribute to the organization's repository. If you just want to copy it and then modify it for your own purpose on you computer, please follow the steps described in the Copy a GMACS-project repository section.

Warning: *the following workflow will provide you with "real" access to the GMACS repository. Therefore, any changes you may make to this repository can become "permanent". Please, never delete an important file or folder if you are not sure. Thank you.*

1. In Github Desktop, click *File > Clone Repository*
2. In the *"Clone a repository"* popup window, chose the URL tab and paste the GMACS_Assessment_code repository url.
3. Check the selected folder in the local path and click `Clone`.

You are now ready to make some changes to GMACS, commit them and push up them to the organization repository. You can also create a new RStudio project in the folder on holding the GMACS_Assessment_code repository on your computer and start to work with GMACS outputs.

3.2 Copy a GMACS-project repository

Here I show you how to copy the GMACS_Assessment_code repository on your computer. By follwoing these steps, you will not be able to contribute to the GMACS-project organisation but you will have the possibility to adapt this repository for your purpose without fear of impacting the organization's repository. Below are the steps to follow:

1. Get and copy the url of the GMACS_Assessment_code repository.
2. Go to you Github account. It should be github.com/your-name.
3. In Github, click the `+` in top right and select `import repository`.
4. Paste the url in ***the Your old repository's clone URL*** section and give a name for this new repo. You have now the GMACS_Assessment_code on your own Github (i.e., you have an url looking like *github.com/your-name/GMACS_Assessment_code* if you kept the same name as the one of the organization).
5. You can now clone this repository to your computer following the same steps as those used to clone the GMACS_Assessment_code repository from the organization website.

4 Let's work on a new version of GMACS

In the **GMACS** folder of the GMACS_Assessment_code repository, two subfolders should attract your attention when you want to develop a new version of GMACS. I'm talking about development when you make a change (no matter how small) to the GMACS code.

1. In the **Dvpt_Version** folder, you will find the version of GMACS currently in development. This means that this version is still being tested after some modifications and has not been released as a stable version.
2. The latest stable and tested version of GMACS is hold in the Latest_Version subfolder.

These two subfolders contain all the hardware you need to run GMACS either in a "development way" or to get a stable assessment.

If you want to make new changes to the GMACS code, please work only in the sub-folder dvpt_Version.

In the following, I give you some guidelines about how correctly update a GMACS version and push up those change on Github. Obviously, this assumes that you have previously forked (and not cloned) the organization's repo on your computer.

Let's take the example of the incorporation of time-varying natural mortality in GMACS.

Reminder: **You will be working in the Dvpt_Version sub-folder.**

4.1 Modify the gmacsbase.tpl

In the Dvpt_Version subfolder, open the gmacsbase.tpl file in an editor and incorporate new functions/variables to get a time-varying natural mortality configuration. This operation does not require any "new entries" or variable declarations in the .ctl, .dat or .prj files so that the change you make will remain (first) at the gmacsbase.tpl file level.

4.2 Check compilation and build the executable

Once you are done with the implementation of the new insigths in GMACS, you need to check that you are able to compile the model and build the executable.

The R script [Update_GMACS.Rmd] will guide you through all these steps. A number of functions have been developed to make the process of updating GMACS easier and more transparent.

If the compilation worked, then you can modify (if needed) the .dat, .ctl, .prj files.

4.3 Modify the .dat, .ctl, .prj files

4.4 Compare the results of this new version with the results from the last assessment

5 Spread the new GMACS version to the whole community

5.1 Push up changes to the organization's repository

5.2 Re-run the assessments with the latest version (optional)