



Marine Ecological Modelling Global Climate Change

Dissemination of results under the Open Science framework

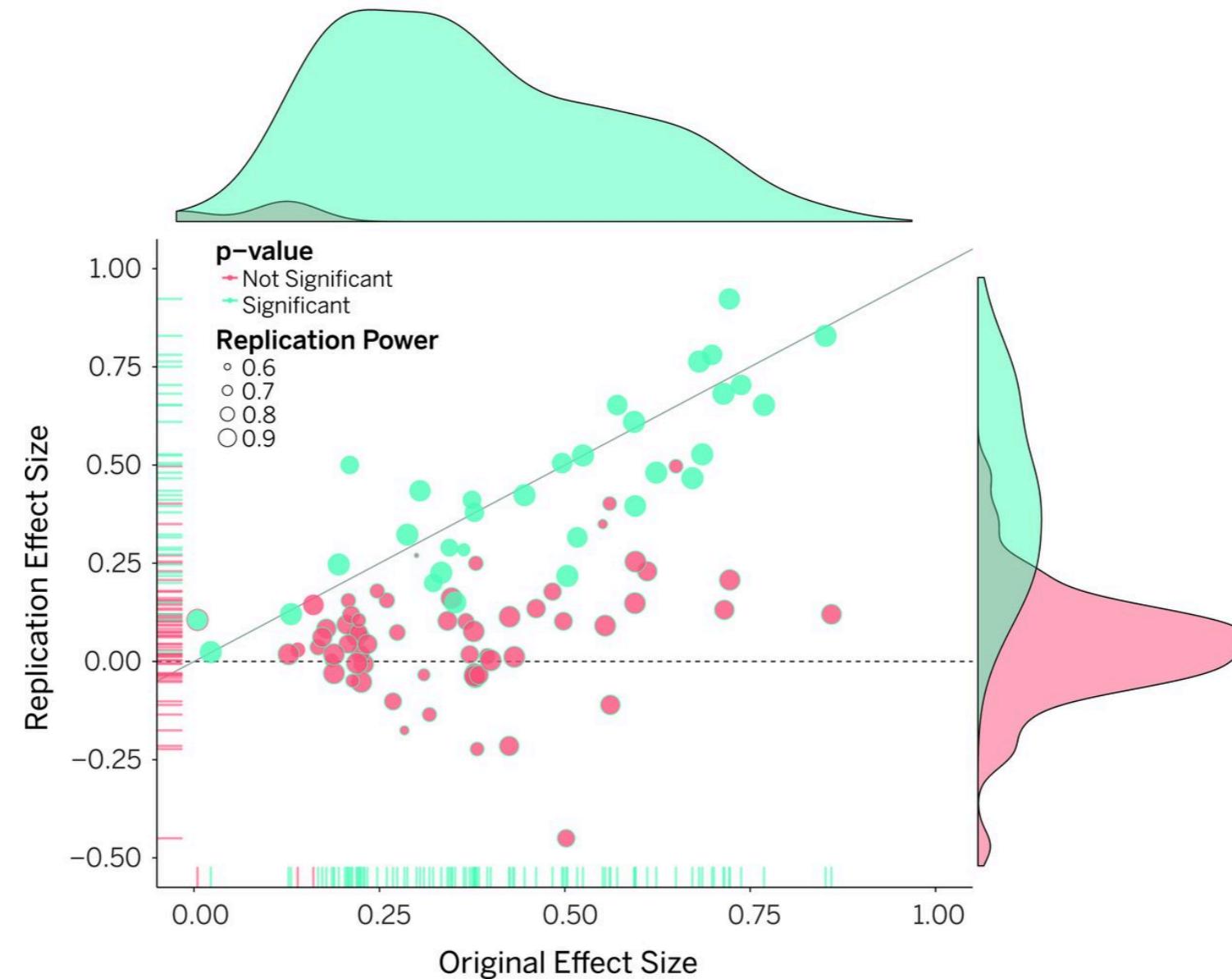
Jorge Assis, PhD // jmassis@ualg.pt // jorgemfa.medium.com
2020, Centre of Marine Sciences, University of Algarve



Open Science

Most scientists spend much time thinking about the types of data they need for their studies. Relatively **little effort is spent** considering **how to store, analyze and share their data.**

It is increasingly important to store and document scientific data in ways that facilitate: (i) Open Science; and (ii) their effective retrieval and interpretation in the future.



A replication os 100 studies show “97% of original studies had significant results ... 36% of replications had significant results”. (abstract). If no bias is assumed in the original results, there is little potential for replication (lack of open science).

** Diagonal line represents replication effect size equal to original effect size.



Open Science

Sharing information is ideal in science, but the reality looks like this:

A scientist **collects data and stores it on a machine.**

He writes or modifies a few small programs (which also reside on his machine) to analyze that data.

With results, he writes and submits a paper. He **might include data** (a growing number of journals require it) **but not the code.**

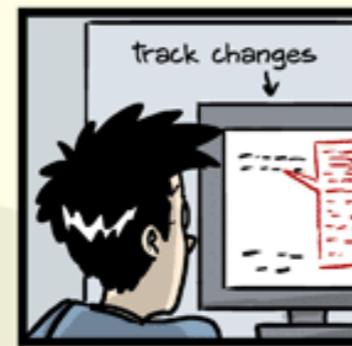
The paper is published and might include a link to a copy of data, but the **paper itself will be behind a paywall.**



"FINAL".doc



JORGE CHAM © 2012



FINAL_rev.18.comments7.
corrections9.MORE.30.doc

FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSSCHOOL????.doc



Open Science

For a growing number of scientists the process looks like this:

The data is stored in an open access repository like **figshare** or **Zenodo** and given its own **Digital Object Identifier (DOI)**.

The scientist creates a new **repository on GitHub** to hold the work.

As he does the analyses, changes are pushed to that repository.

When he is happy with the state of the paper, he posts a version to **preprint server** to invite feedback from peers.

The **published paper includes links to the preprint, to the code and data repositories**, which makes it easier for other scientists to replicate or use the work as starting point for additional research.



Open Science

Associating open data can increase the citation rate of scientific papers by as much as **69%** regardless the journal impact factor, date of publication, and the author's country of origin. But more than 80% of data never makes it to a repository.

Research articles that have been made Open Access are up to 600% more cited than those that have not.



Open Science

Simple Rules for Reproducible Computational Research:

For every result, keep track of how it was produced;

Avoid manual data manipulation steps (unique script does it all);

Archive the exact versions of all external programs used;

Version control all scripts (Git);

Record intermediate results, when possible in standardized formats;

For analyses that include randomness, note underlying random seeds.

e.g., in R use `set.seed(42)`;

Provide public access to scripts (Git) and data (Open repository).



Open access repositories

Online open access repositories allow researchers to **preserve and share their research outputs, including figures, datasets, images, and videos.**

E.g., Figshare and Zenodo are free to upload content and free to access, in adherence to the principle of open data.



Version control systems

Keep track of changes, creating different versions of our files -allows us to decide which changes will be made to the next version (**changes are called commits**).

Complete history of commits make up a repository, which can be kept in sync across different computers, facilitating collaboration.

Version control systems **start with a base version** of the document and then **record changes** you make each step of the way. You can rewind to any state of the document.

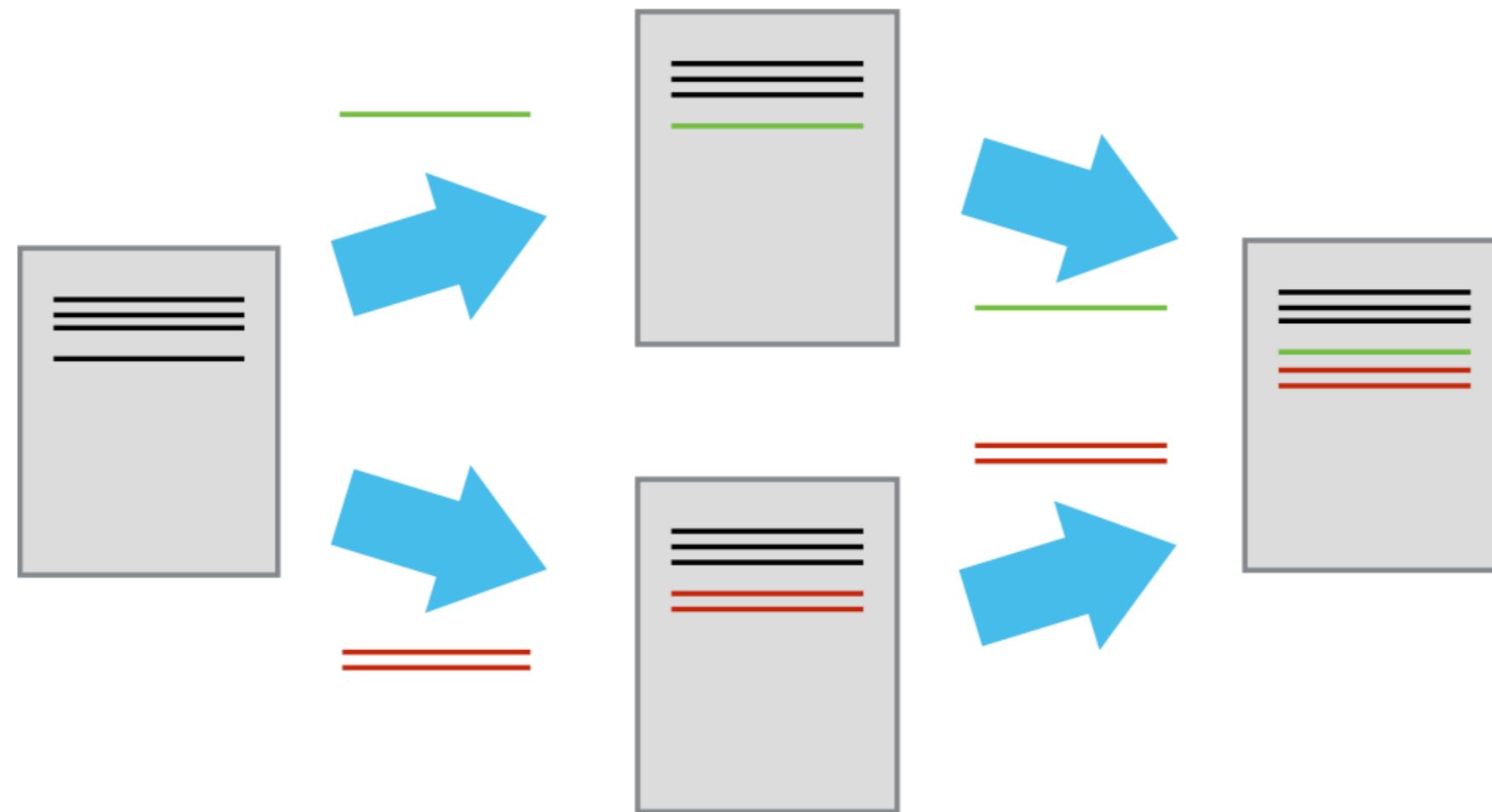




Version control systems

Users can make **independent changes** on the same document.

Unless multiple users make changes to the same section of the document - a conflict - you can **incorporate multiple sets of changes** into the same base document.





Using Git from RStudio

Version control are useful when developing data analysis scripts.

RStudio has built-in integration with Git with a nice interface for many common operations.

RStudio allows to create a **project associated with a Git repository** to track the development and changes of the project over time, revert to previous versions and collaborate with others.



Screenshot of the GitHub 'Create a new repository' page.

The page includes the following elements:

- Owner:** jorgeassis
- Repository name ***: modelSeagrass
- Description (optional):** A repository of R scripts to model the distribution of a seagrass species
- Visibility:** Public (selected) / Private
- Initialization:** Initialize this repository with a README (checked)
- Additional Options:** Add .gitignore: None, Add a license: None
- Create repository button:** A green button at the bottom.

Start by creating a new repository (e.g., GitHub).



Screenshot of a GitHub repository page for `jorgeassis / modelSeagrass`. The page shows basic repository statistics: 1 unwatched star, 0 forks, and 0 pull requests. Below the stats, there are navigation links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. A large central area is labeled "modelSeagrass /" and contains a placeholder for file uploads with the text "Drag files here to add them to your repository" and a "choose your files" link. At the bottom, a modal window titled "Commit changes" is open, showing fields for "Add files via upload" and "Add an optional extended description...". It includes two radio button options: "Commit directly to the master branch." (selected) and "Create a new branch for this commit and start a pull request." Below the modal are "Commit changes" and "Cancel" buttons.

The simplest way of sharing code is by uploading files to the repository.



Screenshot of a GitHub repository page for `jorgeassis / modelSeagrass`.

The repository description is: "A repository of R scripts to model the distribution of a seagrass species".

Key statistics shown:

- 3 commits
- 1 branch
- 0 packages
- 0 releases
- 1 contributor

Recent activity:

- jorgeassis** Rename myscript.R to myscript.R (Latest commit 690f2a0 now)
- README.md** Initial commit (2 minutes ago)
- myscript.R** Rename myscript.R to myscript.R (now)

The `README.md` file content is:

modelSeagrass

A repository of R scripts to model the distribution of a seagrass species

Permanently available at <https://github.com/jorgeassis/modelSeagrass>



README.md

Level of information to include in the readme file:

Repository name;

Project summary;

Funding information;

Primary contact(s) information;

Location of data and supporting information.



Proper integration of RStudio with a repository (e.g., GitHub).

The screenshot shows the RStudio interface with the following details:

- Console:** Displays the R startup message for version 3.6.2 (2019-12-12) "Dark and Stormy Night".
- Project:** Shows "(None)" in the top right.
- File Explorer:** Shows the contents of the user's home directory, including ".R", ".Rhistory", and various folders like Applications, Desktop, Documents, Downloads, Dropbox, Dump, imageFiles, Library, Movies, Music, Parallels, Pictures, Public, and renv.
- Create Project Dialog:** A modal window titled "Create Project" is open, listing three options:
 - New Directory:** Start a project in a brand new working directory.
 - Existing Directory:** Associate a project with an existing working directory.
 - Version Control:** Checkout a project from a version control repository.



Proper integration of RStudio with a repository (e.g., GitHub).

The screenshot shows the RStudio interface with a dark theme. The top bar displays 'RStudio' and the status 'Project: (None)'. The left pane is the 'Console' showing the standard R startup message. A modal dialog titled 'Clone Git Repository' is open in the center. It contains fields for 'Repository URL' (set to <https://github.com/jorgeassis/modelSeagrass>), 'Project directory name' (set to 'modelSeagrass'), and 'Create project as subdirectory of' (set to '/Desktop'). There are also 'Back' and 'New Project' buttons. The right pane shows a file browser with a list of files and folders in the user's home directory.

R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

New Project

Clone Git Repository

Repository URL:
<https://github.com/jorgeassis/modelSeagrass>

Project directory name:
modelSeagrass

Create project as subdirectory of:
~/Desktop

Open in new session

Environment History Connections

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home

Name Size Mod

.R .Rhistory Applications Desktop Documents Downloads Dropbox Dump imageFiles Library Movies Music Parallels Pictures Public renv



Saved edited files can be submitted by clicking on “Commit...”

The screenshot shows the RStudio interface with a GitHub integration. The top right corner features a yellow circular icon with a black dog silhouette. The main window displays an R script named 'myscript.R' with the following code:

```
1 rm(list=ls()[ls()!="v"])
2 gc(reset=TRUE)
3
4 source("Dependencies/mainFunctions.R")
5 source("0. config.R")
6
7
8 # -----
9
10 # I can edit the data in R and submit to github
```

The GitHub commit interface on the right shows a list of files: '.gitignore', 'modelSeagrass.Rproj', and 'myscript.R'. The 'myscript.R' file is marked with a blue 'M' icon, indicating it has been modified. Below this is a file browser showing the project structure and files:

| Name | Size | Last Modified |
|---------------------|-------|---------------|
| .. | | |
| .gitignore | 40 B | Apr |
| modelSeagrass.Rproj | 205 B | Apr |
| myscript.R | 175 B | Apr |
| README.md | 90 B | Apr |

The bottom section shows the R console output:

```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

The status bar at the bottom indicates the time as 10:48.



A dialogue opens where we can select which files to commit, and enter a commit message. The icons in the “Status” column indicate the current status of each file (e.g., M for modified). Clicking on a file shows information about changes in the lower panel. Once everything is the way we want it, we click “Commit”. Changes can be pushed by selecting “Push Branch” from the Git menu.

RStudio: Review Changes

Changes History master Stage Revert Ignore Pull Push

Staged Status Path

- .gitignore
- modelSeagrass.Rproj
- myscript.R

Commit message
Up.

Show Staged Unstaged Context 5 line Ignore Whitespace Unstage All

```
@@ -5,9 +5,6 @@ gc(reset=TRUE)
5 5 source("Dependencies/mainFunctions.R")
6 6 source("0. config.R")
7 7
8 8 # -----
9 9
10 # load(paste0(dataRecordsDirectory, "envEnsembleModels3.RData"))
11
12 occurrenceRecords <- read.table(gsub(".csv", " Processed.csv", dataRecordsFile), sep = ";", header = T)
13 speciesData <- read.table( paste0(substring( dataRecordsFile , 1 ,
as.numeric(gregexpr("/", dataRecordsFile)[[1]])[length(as.numeric(gregexpr("/", dataRecordsFile)[[1]])) ] ) ,
"speciesData.csv"), sep = ";", header = T)
10 # I can edit the data in R and submit to github
No newline at end of file
```



Screenshot of GitHub showing a repository named `jorgeassis / modelSeagrass`. The repository description is "A repository of R scripts to model the distribution of a seagrass species". The main page includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings.

The repository has 4 commits. A file named `myscript.R` is displayed, showing its content:

```
1 rm(list=ls()[ls()!="v"])
2 gc(reset=TRUE)
3
4 source("Dependencies/mainFunctions.R")
5 source("0.config.R")
6
7 # -----
8
9
10 # I can edit the data in R and submit to github
```

A sidebar on the left shows other files in the repository: `README.md`, `myscript.R`, and another `README.md`.

Allows working in RStudio with a version control and backup system.