**Conda is insufficient**

The current conda environment isn’t working properly. Prior experience showed conda’s r channel and conda-forge channel don’t work nice together. However, we do need to specify a version of R with the conda environment yaml, and the r channel one seemed to work better with Rstudio.

Christa has been having major issues with getting her packages to work, but managed to compile this list for functional genomics:

```

dependencies:

- r::rstudio

- r::r>=3.6.0 # any r must be specified to install rstudio

- r::r-devtools=2.0.2

- conda-forge::r-learnr=0.10.1 # this should install shiny and Rmarkdown

- conda-forge::r-upsetr # for combinatorial bar plots with dottable below

- conda-forge::r-pheatmap # for nice heatmaps

- bioconda::bioconductor-iranges # for working with interval data

- bioconda::bioconductor-rtracklayer # for importing bed files

- bioconda::bioconductor-genomicfeatures # for manipulating txdb objects

- bioconda::bioconductor-genomicranges # for working with genomic interval data

- bioconda::bioconductor-txdb.hsapiens.ucsc.hg38.knowngene # annotatations

- bioconda::bioconductor-chipseeker # tools for analysis of chipseq data

- bioconda::bioconductor-chippeakanno # another set of tools for chipseq data

- r::r-tidyverse=1.2.1

- conda-forge::r-dplyr=0.8.5 # had to update to this version for chipseeker

- conda-forge::r-tidyr=1.1.2 # had to update to this version for chipseeker

- bioconda::bioconductor-org.hs.eg.db

- bioconda::bioconductor-genomation

- bioconda::bioconductor-plyranges=1.6.0

```

Note: Rstudio server and Shiny Server both require a system installed R. It was added here to make Rstudio work. The System install of R is at version 3.6.3.

These packages seem to work fine, but it stops working when adding the remaining packages.

**Problem**

At least one package must come from github: gradethis. The latest gradethis comes with several nice features, but requires the latest github version of learnr as well, which causes a chain reaction with several packages (gradethis, learnr and htmltools from github, and strinig from CRAN).

Once downloaded we have compatibility issues.

**Solution**

Rstudio has recently released the second version of their own package manager: renv.

Once an environment is set up, renv can catalogue every used package and its source, including Bioconductor and github packages.

Using remotes::install\_github() I can specify which commit to download, which is also saved(!). If no commit is specified, it will catalogue HEAD (the latest commit).

Combined this means I can make the environment work on my machine, and renv promises it will work for others.

**Workflow**

The tutorial is developed in an Rstudio project. This means it has its own directory (which is connected to git), with startup script and all. This directory is indexed by renv with `renv::init()`:

```

renv::init(

project = NULL, # current working dir

settings = list(

r.version = "3.6.3", # specify this R version

vcs.ignore.library = T # don't add libraries to git

),

restart = T # restart the rsession (so it's truly empty)

)

```

This logged the `library()` calls in every file, and logged the versions in my general library.

Once set up, renv creates a libpath specific to this project, very similar to how I did it previously! This prevents conflicts with the global env, because it cannot be accessed. Making this library accessible as discussed in 4, we can use this to run the tutorial for anyone.

After initializing, there are only 2 commands you need regularly.

**Daily workflow**

To save changes in packages, use `renv::snapshot()`.

This updates the renv.lock file, which is under version control.

To apply these changes, use `renv::restore()`.

Note: Some other interesting renv commands are hydrate(), status() and dependency().

**Notes**

Renv has 2 options that need to be checked:

* use.cache: renv will store packages in a cache and symlink them to the renv lib. So the cache must be accessible for other users if this option is set. As home folders are hidden when the user is offline on the server, I’ve turned this off by setting `use.cache: FALSE` in `projectfolder/renv/setting.dcf`.
* sandbox: renv symlinks the minimal set of system packages in a single library in /tmp. This folder only exists while the project is loaded (e.g. I am logged in). So users must be able to use the regular R system libraries if this folder cannot be found. The setting can be turned off entirely by creating file `.Renviron` in the project base directory with `RENV\_CONFIG\_SANDBOX\_ENABLED = FALSE`

Although both features are very useful in general, I’ve turned them off. The end result is that there are now **up to four** libraries that must be accessible to all users: the renv library, and the (3) default system libraries. (Using conda there are no system libraries, but instead the conda library).

