

Example PDF Report

Contents

R Markdown Basics	1
R Markdown	1
Insert pre-existing content	1
image	1
text	2
Rmarkdown and BioLockJ	3

R Markdown Basics

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Insert pre-existing content

Including pre-existing images and text is especially convenient for generating reports. The markdown itself is just an assembly tool for putting together results that were generated in upstream modules. The upstream modules focus on individual results; while this report weaves them together into a comprehensive story.

image

The `include_graphics()` method is very handy for including images. For example:

```
include_graphics("sheep_mocha.jpg")
```



text

Here is an example referencing a text file:

Photo credit Ivory Blakley 2020

Rmarkdown and BioLockJ

BioLockJ includes a module called `Rmarkdown`. You can add it to the module run order with this line:

```
#BioModule biolockj.module.diy.Rmarkdown
```

Use relative paths in the Rmd file to reference pipeline files. For example:

```
workingDir <- getwd()
moduleDir <- dirname(workingDir)
pipeRoot <- dirname(moduleDir)
```

For ALL BioLockJ modules, the working directory is a subfolder of the module directory, and all modules in a pipeline have a folder immediately under the pipeline root directory. So the three lines in the above chunk are always the same anytime R is used in a BioLockJ pipeline.

In this case, the “working directory” is the `resources` sub-directory of module folder for module: `rmarkdown` .

The module is in a pipeline located here:

```
/Users/ieclabau/git/sheepdog_testing_suite/test/module
```

To reference other modules in this pipeline, use the pipeline path and the modules name. Do not reference the module number, that part of the name may change in different iterations of the pipeline. But the module name is consistent. For example:

```
genModModule <- dir(pipeRoot, pattern="GenMod")
genModOutFile <- file.path(genModModule, "output", "time.txt")
```

This relative reference style assumes that this is running in the context of a BioLockJ pipeline, and that the pipeline includes a module “01_GenMod” that includes an output file called `time.txt`. The markdown report for a given pipeline is tailored to the pipeline.

The `time.txt` file says:

Sorry, I don't see a GenMod module in the current pipeline.

This report was generated at: Fri Nov 27 16:34:28 2020 .

It's a good practice to end scripts with reports with a call to `sessionInfo()` :

```
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] knitr_1.30
##
## loaded via a namespace (and not attached):
## [1] compiler_4.0.2  magrittr_2.0.1  tools_4.0.2     htmltools_0.5.0
## [5] yaml_2.2.1      stringi_1.5.3   rmarkdown_2.5   stringr_1.4.0
## [9] xfun_0.19       digest_0.6.27   rlang_0.4.8     evaluate_0.14
```