

Reproducible pipelines

using **targets** and **renv**





A. Ginolhac | rworkshop | 2021-09-10

targets



targets and companion package tarchetypes



A workflow manager for R

- Saving you time and stress
- Understand how it is implemented in `targets`
 - Define your `targets`
 - Connect `targets` to create the `dependencies` 
 - Check `dependencies` with `visnetwork`
 - Embrace `dynamic` branching 
 - Run `only` what needs to be executed
 - Bundle `dependencies` in a Rmarkdown document with `tar_render()`
 - Increase reproducibility with the package manager `renv`
- Example with RNA-seq data from **Wendkouni Nadège MINOUNGOU**



Folder structure

```
.git/  
run.R  
_targets.R  
_targets/  
Repro.Rproj  
R  
  functions.R  
  utils.R  
run.R*  
renv/  
renv.lock  
report.Rmd
```

- With [renv](#). Snapshot your package environment (and restore! 📦)
- [_targets.R](#) is the only mandatory file
- Use a [R](#) sub-folder for functions, gets closer to a  package
- [Rmarkdown](#) file allows to gather results in a report
- In a RStudio project
- Version tracked with 

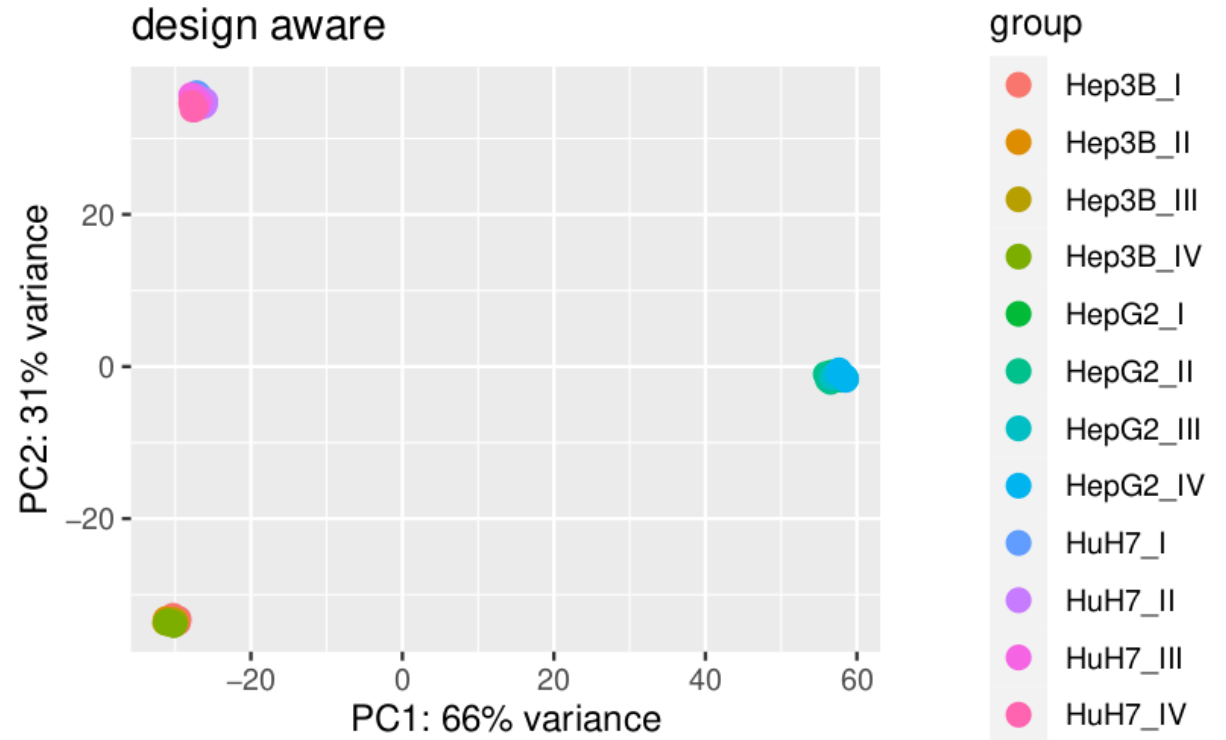


Example with RNA-seq data across 3 cell lines

PCA shows that differences between cells >> biological effect (roman numbers)

Solution: Split counts and metadata for each cell

Do we copy code 3 times?



Define targets = explicit dependencies



_targets.R, define 4 targets

Last **target** depends on the **3** upstreams

Dynamic branching makes dependencies easier to read.

Of course, someone has to write for loops, it doesn't have to be you

— Jenny Bryan

```
library(targets)
source("R/functions.R")
source("R/plotting.R")

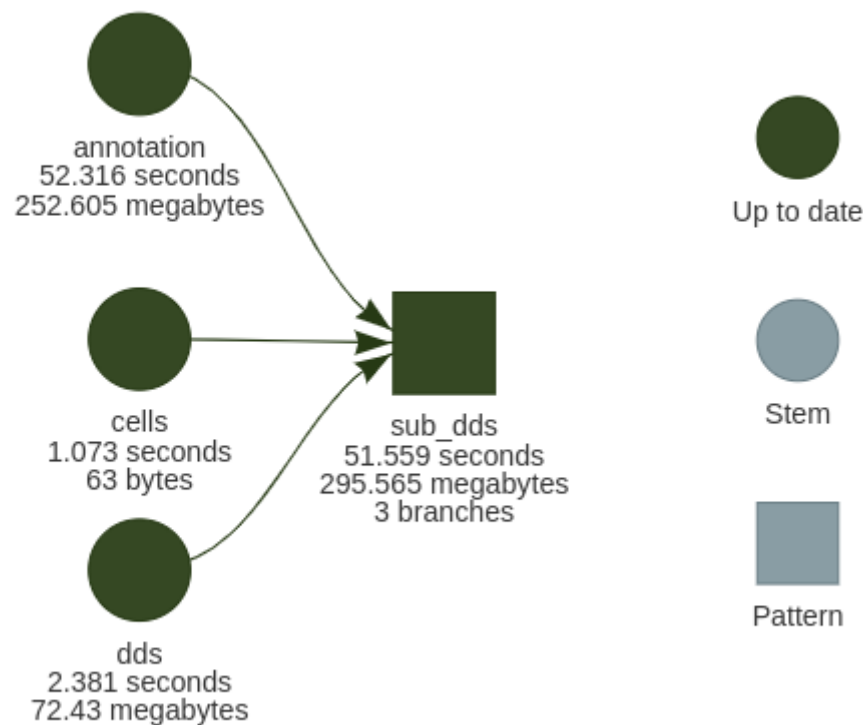
list(
  tar_target(cells, c("HepG2", "HuH7", "Hep3B")),
  tar_qs(dds, read_rds(here::here("data", "all.rds")),
    packages = "DESeq2"),
  tar_fst_tbl(annotation, gtf_to_tbl(here::here("data",
    "gtf", "gtf.v36.annotation"),
    packages = c("tibble", "rtracklayer")),
  tar_qs(sub_dds, subset_dds(dds,
    filter(annotation, type == "gene")
    .cell = cells),
    pattern = map(cells), # dynamic branching
    packages = c("DESeq2", "tidyverse"))
  [...])
```

Figure from `tar_visnetwork()`

Running targets

```
• run target annotation
• run target cells
• run target dds
• run branch sub_dds_3078b1e0
    condition time_h
HepG2_I1    control    0
HepG2_I2    HIL6       2
using pre-existing size factors
estimating dispersions
gene-wise dispersion estimates: 2 workers
mean-dispersion relationship
final dispersion estimates, fitting model and testing: 2 worker
• run branch sub_dds_d05c5da7
    condition time_h
HuH7_I1    control    0
HuH7_I2    HIL6       2
using pre-existing size factors
estimating dispersions
gene-wise dispersion estimates: 2 workers
mean-dispersion relationship
final dispersion estimates, fitting model and testing: 2 worker
• run branch sub_dds_c60d7096
    condition time_h
Hep3B_I1    control    0
Hep3B_I2    HIL6       2
using pre-existing size factors
estimating dispersions
gene-wise dispersion estimates: 2 workers
mean-dispersion relationship
final dispersion estimates, fitting model and testing: 2 worker
• end pipeline
```

Options to display time and object sizes



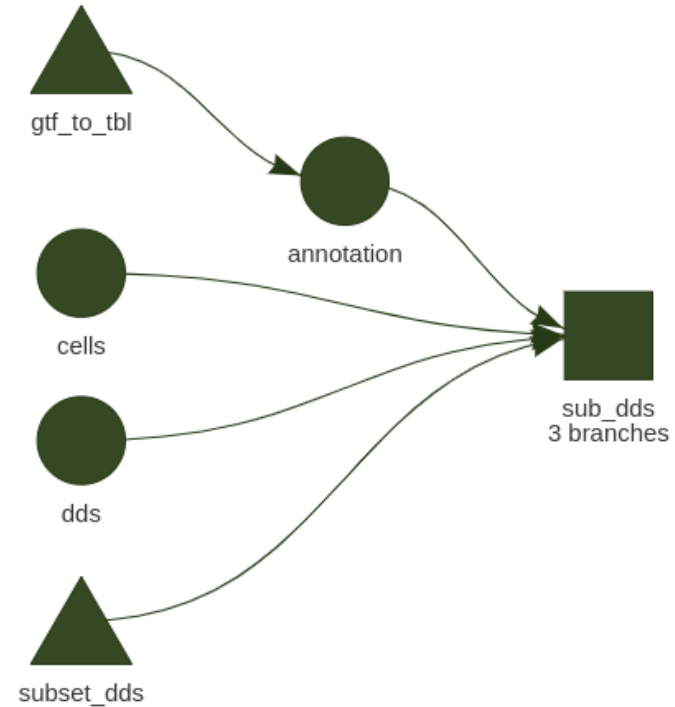
Re-running

- ✓ skip target annotation
- ✓ skip target cells
- ✓ skip target dds
- ✓ skip branch sub_dds_3078b1e0
- ✓ skip branch sub_dds_d05c5da7
- ✓ skip branch sub_dds_c60d7096
- ✓ skip pipeline

All good, nothing to be done ☐.

Actually **targets** tracks all objects and so functions

A more complete dependency graph shows **functions**



Up to date

Stem

Pattern

Function

Let's add the PCA per cell type now

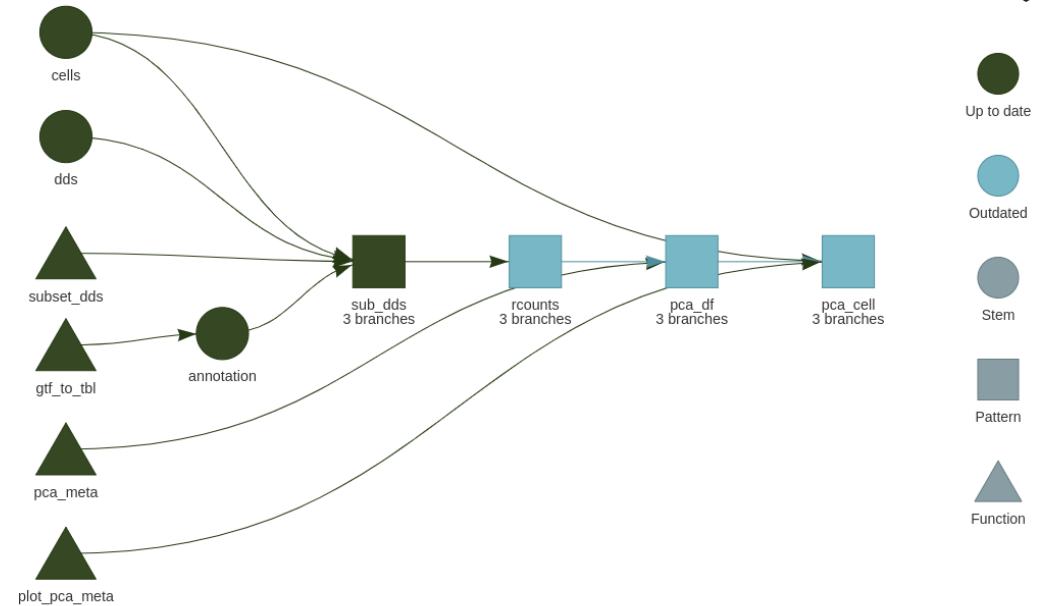
PCA, add 4 targets

Smaller targets avoid unnecessary re-running steps

```
[...]
tar_target(rcounts, vst(sub_dds, blind = TRUE),
  pattern = map(sub_dds),
  packages = c("DESeq2")),
tar_target(pca_df, pca_meta(rcounts),
  pattern = map(rcounts),
  packages = c("DESeq2", "tidyr", "dplyr")),
tar_target(pca_cell, tibble(cell = cells,
  pca = list(plot_pca_meta(pca_df))),
  pattern = map(cells, pca_df),
  packages = c("ggplot2", "tibble"))
[...]
```

Translate into:

- For every cell data, compute regularized counts (**vst**: variance stabilization)
- For every regularized counts, compute PCA (**df**: data.frame, *i. e* a table)
- For every cell names / PCA tables, plot PCA in a table for easier labeling



PCA results

Running

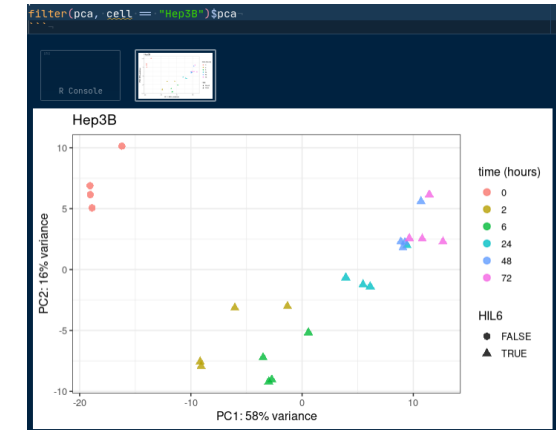


Awesome feature: load results IN a Rmarkdown document

Separate code from content

```
## Split per cell types  
{r: paged.print=FALSE}  
pca ← tar_read("pca_cell")  
pca  
# A tibble: 3 x 2  
  cell  pca  
  <chr> <list>  
1 HepG2 <gg>  
2 HuH7  <gg>  
3 Hep3B <gg>
```

How to display a plot



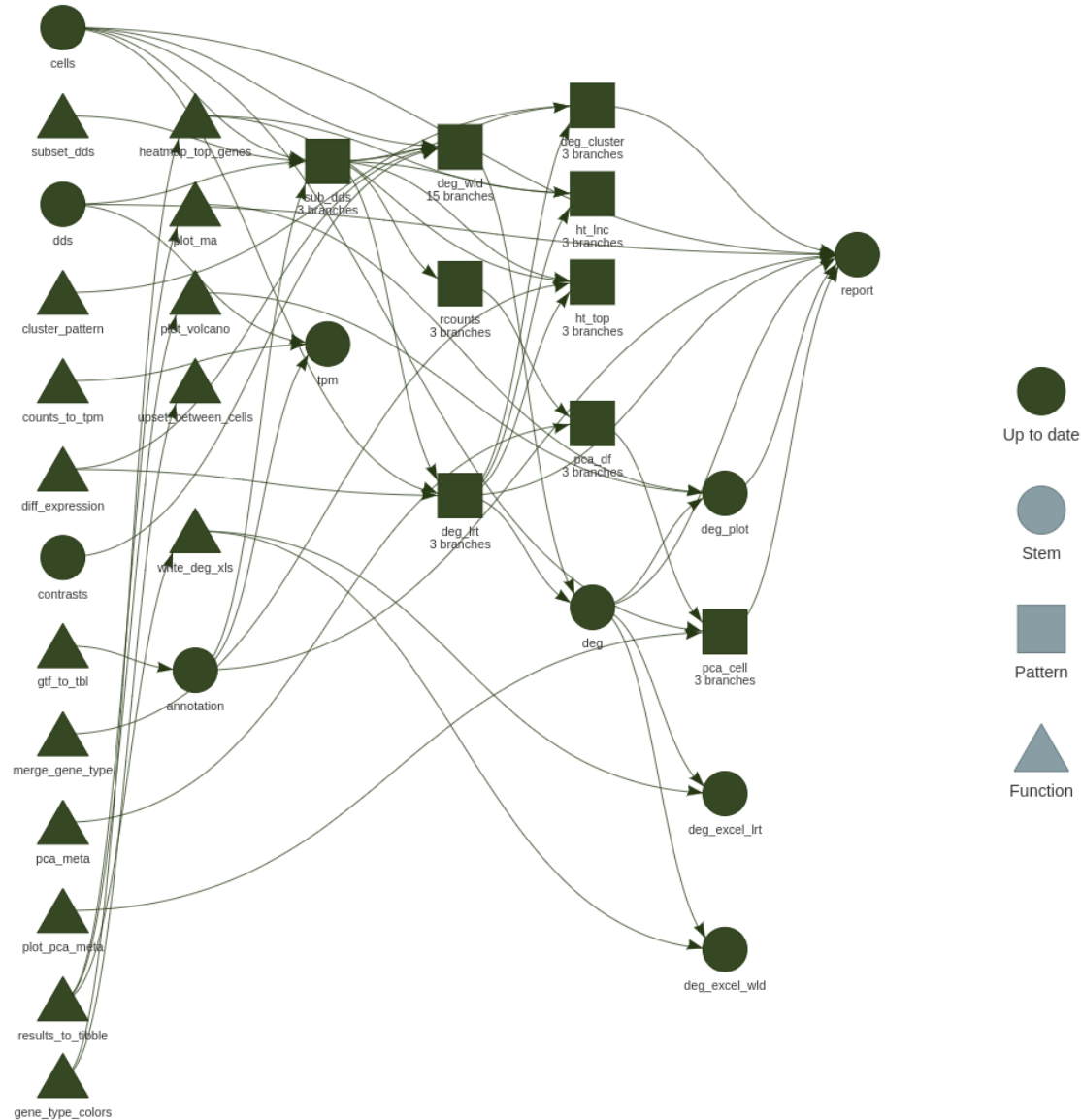
The full picture

Adding step by step desired analyses

Whole analysis takes 24 minutes and 4.54 seconds

Of course, someone has to remember the dependencies, it doesn't have to be you

— could be William Landau via Jenny Bryan



Is it worth the effort?

Yes

For you

- independence
- autonomy
- skills
- *free* time
- confidence over results
- reproducibility
- fun ☐

Reproducibility

`targets` via `git`

```
> renv::history()
  commit      author_date      committer_date      subject
1e8dd2278 2021-02-23 15:29:57 2021-02-23 15:29:57 reformat creating config files
24c1222db 2021-02-15 17:07:01 2021-02-15 17:07:01 highlight gene type in the DEG patterns
326c8a726 2021-02-04 16:16:38 2021-02-04 16:16:38 cluster LRT genes by they dynamic patterns
4c6791796 2021-01-26 13:08:15 2021-01-26 13:08:15 gene types in upset plots for lengths
5865ee70b 2021-01-21 16:36:48 2021-01-21 16:37:08 add upset plots
6c06b496e 2021-01-20 17:03:29 2021-01-20 17:03:29 add upset protoype
[...]
```

renv features

- **renv** parses your code and finds library calls
- **install** a package including its dependencies
- **snapshot** registers changes, hashes and origin

```
> renv::snapshot()
The following package(s) will be updated in the lockfile:
```

```
# CRAN =====
- RcppParallel [5.0.2 -> 5.0.3]
- cli [2.3.0 -> 2.3.1]
- pkgload [1.1.0 -> 1.2.0]
- tint [0.1.3 -> *]

# GitHub =====
- targets [ropensci/targets@main: 598d7a23 -> bdc1b29c]

Do you want to proceed? [y/N]:
```

restore to a certain point in time

renv.lock file after a **snapshot**

```
"R": {
  "Version": "4.0.3",
  "Repositories": [
    {
      "Name": "CRAN",
      "URL": "https://cloud.r-project.org"
    }
  ]
},
"Bioconductor": {
  "Version": "3.12"
},
"Packages": {
  "AnnotationDbi": {
    "Package": "AnnotationDbi",
    "Version": "1.52.0",
    "Source": "Bioconductor",
    "Hash": "ca5106b296b3aa6af713ce197be547c1"
  },
  "BH": {
    "Package": "BH",
    "Version": "1.75.0-0",
    "Source": "Repository",
    "Repository": "CRAN",
    "Hash": "e4c04affc2cac20c8fec18385cd14691"
  },
  "targets": {
    "Package": "targets",
    "Version": "0.1.0.9000",
    "Source": "GitHub",
    "RemoteType": "github",
    "RemoteUsername": "ropensci",
```



Reports as Rmarkdown documents



`targets`, written by [William Landau](#) (pictured), is flexible, robust and still allows for a customized report.

All computing is done only when needed, and code is away from writing content.

Once `knitted` the report can be sent to the inquirer.

Targets Markdown

New in `targets` > **0.6**. Instructions at [William bookdown](#)

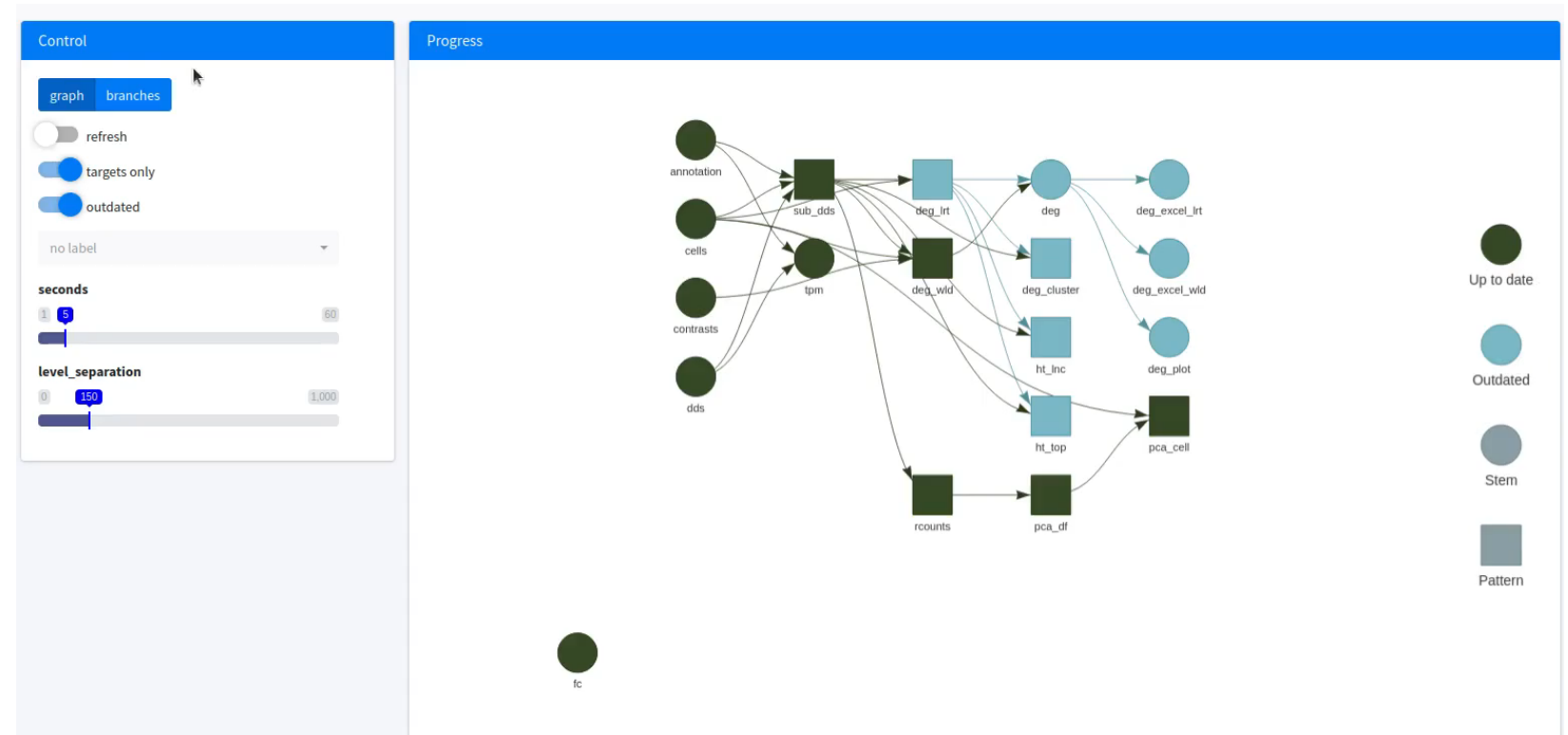
Test it as the Rmd template (and excellent [video](#) from R Lille meetup by [Landau](#)):

Bonus: watch the pipeline running live



- **targets** events watched live
- Here, after changing a threshold in the LRT step
- **branches** can be monitored too
- 2 videos joined as I fixed an **error** at 1'42"
- Option to display functions (unset here)

tar_watch() shiny app from targets



Before we stop (for good)

Highlights

- [targets](#), dependencies manager, re-run what's needed
- [renv](#), bundled your packages and versions inside a project

Acknowledgments ☐ ☐

- [Eric Koncina](#) early adopter of [targets](#)
- [Wendkouni N. Minoungou](#) for the RNA-seq data
- [William Landau](#) main developer of [targets](#)

Further reading 📖

- [Main website](#)
- [Targetopia](#) [Landau](#) universe of targets-derived
- [Video](#) from R Lille meetup by [William Landau](#). June 2021 45"
- [Documentation](#) as bookdown by [Landau](#)

Thank you for your attention!
Hope it was useful!