

NETISCE Manual and Tutorials

Lauren Marazzi

2021-11-29

Contents

1	About	5
2	Installation and Usage	7
2.1	Download NETISCE	7
2.2	Install Nextflow	7
2.3	Parameters and Configuration	7
2.4	Running NETISCE	8
3	NETSICE output	11
3.1	Chapters and sub-chapters	11
3.2	Captioned figures and tables	11
4	Toy Network Examples	15
5	Cell Fate Specification Example	17
6	Pluripotent Stem Cell Example	19
7	Adaptive Resistance in Colorectal Cancer Example	21
8	Footnotes and citations	23
8.1	Footnotes	23
8.2	Citations	23

9	Blocks	25
9.1	Equations	25
9.2	Theorems and proofs	25
9.3	Callout blocks	25
10	Sharing your book	27
10.1	Publishing	27
10.2	404 pages	27
10.3	Metadata for sharing	27

Chapter 1

About

Welcome to the NETISCE manual and tutorials. NETISCE is a workflow that identified control node perturbations for cellular reprogramming. This manual contains directions for initial setup, as well as tutorials for reproducing NETISCE cell reprogramming results in developmental, stem cell, and pluripotent biology.

Chapter 2

Installation and Usage

2.1 Download NETISCE

NETISCE pipelines can be downloaded from our github repository: <https://github.com/veraliconaresearchgroup/netisce>

We recommend that you run NETISCE on a high-performance cluster, as you may potentially generate files that are quite large, or run computations that may take a long time. However, we provide two Nextflow pipelines, one designed for hpcs (NETISCE_hpc), and another for running NETISCE on a local machine (NETISCE_local)

2.2 Install Nextflow

Nextflow is required to run the NETISCE pipeline. Please follow the instructions from <https://www.nextflow.io/> (see 'Getting Started' steps 1 & 2) to install Nextflow in the appropriate NETSICE folder (__local or __hpc)

2.3 Parameters and Configuration

Whether on your local machine or hpc, to run NETISCE you must specify the files and parameters within the .nf file

- params.expressions: csv file containing normalized expression data for network nodes in different samples
- params.network: network file (sif format)

- `params.samples`: text file specifying the phenotype for each sample in `params.expressions` file (tab delimited)
- `params.internal_control`: text file containing a list of nodes to be used as internal marker nodes
- `params.alpha`: alpha parameter for signal flow analysis (default =0.9)
- `params.undesired`: string of the undesired phenotype (as labeled in the `params.samples` file)
- `params.desired`: string of the desired phenotype (as labeled in the `params.samples` file)
- `params.filter`: filtering parameter for criterion 2 (“strict” or “relaxed”)
- `params.kmeans_min_val`: minimum k-means value for clustering (default=2)
- `params.kmeans_max_val`: maximum k-means value for clustering (default=10)
- `params.num_nodes`: number of nodes in network for which normalized expression data exists (within the `params.expressions` file)
- `params.num_states`: number of randomly generated initial states (default=100000, or 3^n where n is the number of network nodes and 3^n is less than 100000)

please see the `input_data` folder for examples of files to match the formatting.

2.3.1 NETISCE__mutations.nf

If you are interested in including mutational information, please use the `NETISCE__mutations.nf` pipeline. You must additionally specify `params.mutations`: csv file containing mutational configuration for network nodes (0 for loss of function, 1 for gain of function). Please see example in `input_data`

2.3.2 nextflow.config

If you are running nextflow on an hpc, please specify your executor, and `clusterOptions` within the `nextflow.config` file. Please see <https://www.nextflow.io/docs/latest/config.html> for more information

2.4 Running NETISCE

Once you have specified the parameters, run NETISCE using the following command

```
./nextflow run NETISCE.nf -resume ##or NETISCE__mutations.nf if including mutational data
```


We recommend using the `-resume` flag in the case that you change a file or parameter within your pipeline. This way, nextflow caches results that remain unchanged, preventing pipeline steps from being re-run.

Chapter 3

NETSICE output

Cross-references make it easier for your readers to find and link to elements in your book.

3.1 Chapters and sub-chapters

There are two steps to cross-reference any heading:

1. Label the heading: `# Hello world {#nice-label}`.
 - Leave the label off if you like the automated heading generated based on your heading title: for example, `# Hello world = # Hello world {#hello-world}`.
 - To label an un-numbered heading, use: `# Hello world {-#nice-label}` or `{# Hello world .unnumbered}`.
2. Next, reference the labeled heading anywhere in the text using `\@ref(nice-label)`; for example, please see Chapter `??`.
 - If you prefer text as the link instead of a numbered reference use: any text you want can go here.

3.2 Captioned figures and tables

Figures and tables *with captions* can also be cross-referenced from elsewhere in your book using `\@ref(fig:chunk-label)` and `\@ref(tab:chunk-label)`, respectively.

See Figure 3.1.

```
par(mar = c(4, 4, .1, .1))  
plot(pressure, type = 'b', pch = 19)
```

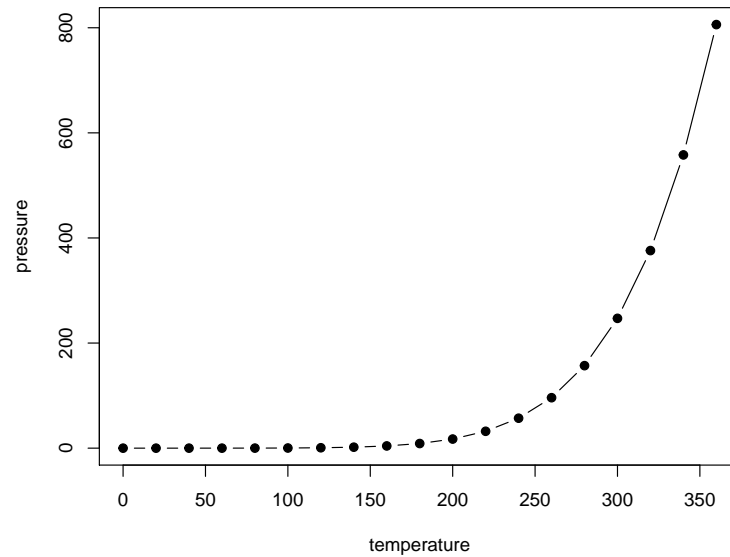


Figure 3.1: Here is a nice figure!

Don't miss Table 3.1.

```
knitr::kable(  
  head(pressure, 10), caption = 'Here is a nice table!',  
  booktabs = TRUE  
)
```

Table 3.1: Here is a nice table!

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000

Chapter 4

Toy Network Examples

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 5

Cell Fate Specification Example

Chapter 6

Pluripotent Stem Cell Example

Chapter 7

Adaptive Resistance in Colorectal Cancer Example

Chapter 8

Footnotes and citations

8.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one ¹.

8.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2021] (check out the last code chunk in `index.Rmd` to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file `book.bib`). Note that the `.bib` files need to be listed in the `index.Rmd` with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: <https://rstudio.github.io/visual-markdown-editing/#/citations>

¹This is a footnote.

Chapter 9

Blocks

9.1 Equations

Here is an equation.

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (9.1)$$

You may refer to using `\@ref{eq:binom}`, like see Equation (9.1).

9.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref{thm:tri}`, for example, check out this smart theorem 9.1.

Theorem 9.1. *For a right triangle, if c denotes the length of the hypotenuse and a and b denote the lengths of the **other** two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here <https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html>.

9.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>

Chapter 10

Sharing your book

10.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

10.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

10.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book—all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL <http://yihui.org/knitr/>. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2021. URL <https://CRAN.R-project.org/package=bookdown>. R package version 0.24.