Douglas Neuroinformatics Platform Documentation

Release 0.1

The Douglas Neuroinformatics Platform

May 17, 2022

THINGS WE KNOW ABOUT

Hello and welcome to our documentation page.

CHAPTER

ONE

ABOUT THE PLATFORM

The Douglas Neuroinformatics Platform is a single-sign-on Ubuntu Linux based computer system. Originally the computing platform of the Douglas Cerebral Imaging Centre (CIC), it has grown and generalized to meet the informatics needs of the Douglas Research Centre as a whole.

1.1 Hardware

The system itself consists (as of mid-2021) of Linux-based single-sign-on user management system, along with a 1 PB storage system, ~40 8-12 core workstations with 16-32 GB of RAM, and a 10-node compute cluster of 12-core 48 GB RAM compute nodes. Primary servers are replicated across two sites for disaster recovery. Unified storage across systems is provided via NFS from the primary storage server to all systems. A heterogenous cluster compute environment is provided via a SLURM batch system providing access to the compute cluster as well as opportunistic scheduling on workstations.

Entry-level GPU computation is available on workstations to provide CUDA and OpenCL acceleration suitable applications.

1.2 Software

The platform workstations run Ubuntu Linux variants, with full productivity software (LibreOffice, GIMP, Inkscape, etc), modern web browsers (Chrome, Firefox), and a large suite of scientific software. Servers run Ubuntu Linux server, as well as compute nodes. Limited Windows workstations are available to provide access to proprietary software such as E-Prime. Scientific software is available in the *software quarantine*.

CHAPTER

TWO

GETTING STARTED

Hello. Happy to get you started

2.1 Set up a user account

User account creation is handled by-request via email to contact@douglasneuroinformatics.ca.

Requests must be submitted by the **supervising principal investigator**.

Principal investigators may also request creation of a lab group if one does not already exist.

Once accounts have been created new users will be invited to meet in person (or via video chat) to set a password for their accounts.

2.2 Ask a support question

Support for the Douglas Neuroinformatics platform is provided on the forum located at https://discourse.douglasneuroinformatics.ca/

Invite-only access is provided during initial sign-up.

CHAPTER

THREE

USING THE DNP SYSTEM

Make sure you have received your access information

In order to access and use the Douglas Neuroinformatics Platform, first ensure that you have completed the *Getting Started* section.

3.1 Accessing the System

The platform can be accessed in a number of ways:

1. Physical access
2. Remote access within the Douglas.
3. Remote access outside the Douglas.

3.1.1 Physical Access

Physical access to a suite of workstations is available inside the Douglas CIC, contact the CIC Administrative Assistant (Louis Théroux) for keycard access.

3.1.2 Remote Access within the Douglas

All platform hardware is acccessible within the Douglas Research Centre network via `ssh`. The main userserver is available at `cicus03`, the storage server at `cicss03` and workstations are available in the range `cicws[01..41]`.

Playing nice

Neuroinformatics workstations all allow for multiple simultaneous users, please make a best effort to choose a workstation not already being used by others.

The command `who` can list currently logged-in users, while `htop` will show a graphical display of the current state of CPU and Memory utilization.

In general, fewer users is better, as well as low CPU and memory utilization (represented by the length of the horizonal coloured bars in `htop`)

3.1.3 Remote Access outside the Douglas

Remote access from outside the Douglas is provided by authorization only. Please contact the platform to request access.

3.2 Access to Data

3.2.1 Filesystem Layout

Each computer in the system has access to two system-wide network filesystems. User home directories `$HOME` are mounted from the user server (currently `cicus03`) on `/home/cic/<username>`. Home directories are suitable for storing the regular configuration files, as well as papers and similar files, it should not be used to store data.

The high performance filesystem (currently hosted on `cicss05`) is available under the `/data` path. All users have access to scratch storage at `/data/scratch2`, which is suitable for storing data during processing. Scratch does not keep any historical versions and is not backed up, so it should not be relied upon for long-term storage. In the future it is expected a age-based deletion policy will be implemented to maintain sufficient free space and encourage users to use storage properly.

Paid storage for individual lab groups is also available under the `/data/` path, the exact name depends upon what the group decided during creation.

Mounting on demand

Network filesystems are not automatically mounted on boot-up on machines, but rather mounted lazily on-demand. As such, `/data` will appear empty before attempts are made to access a specific filesystem. GUI file managers do not work particularly well in this scenario, as there will be no folders to interact with. It is recommended to perform file management using the command line tools.

3.2.2 Transferring Data

Access to filesystems is enabled via the `scp/sftp` functionality of `ssh`, as well as the `rsync` program over `ssh`. Linux and OSX users can find the `sftp` `scp` and `rsync` commands in their terminal. Windows users can use WinSCP or FileZilla for a graphical tool to access data or MobaXterm for a proper Linux-like terminal.

Here are few example commands for data transfer, note that these commands assume that you are within the Douglas:

```
# Copying a directory of files to cicss05 in /data/scratch/<username>
$ rsync -avz directory <username>@cicss05:/data/scratch/<username>
# Copying another directory from cicss05 to your local system
$ rsync -avz <username>@cicss05:/data/scratch/<username>/another_dir .
```

Resumable file transfers

`rsync` is strongly recommended for all data transfers, as it supports resuming interrupted transfers

---

Bulk data transfer

For bulk data transfers, please connect directly to `cicss05` to bypass any round-trip data would need to travel if performing transfers to workstations.

Real time data access

The sshfs project allows for filesystems to be mounted remotely via ssh. This allows you to access files without having to explicitly transfer them back-and-forth. See the site for details and for windows see here.

3.2.3 Accessing Human MRI Scanner DICOMS

The Siemens MAGNETOM Prisma Human MRI scanner sends collected data to `cicus03` which acts a pseudo-PACS system, collecting the files and storing them at `/home/cic/dicom/transfers` accessible for 14 calendar days before being moved to long-term cold storage. Users are expected to access their data during that time and copy it to appropriate long term storage. Recovery from cold storage is available with delayed access and a recovery fee associated with staff time.

3.2.4 Accessing Animal MRI Scanner Raw Data

The Bruker BioSpec 70/30 scanner produces data in the raw Bruker data format. The data sets are accessible on the Bruker controlling computer for 14 calendar days from the date of the acquisition at `/opt/<PV version>/data/<username>`, with `<PV version>` the ParaVision software version (PV5.1 or PV6.0.1) and `<username>` the login of the user performing the scans. Users are expected to access their data during that time and copy it to appropriate long term storage. Recovery is available with delayed access and a recovery fee associated with staff time. Data can be accessed by 1) accessing the *Neuroinformatics Platform* and then 2) accessing the bruker7t computer. The *tools for data transfer* will work, provided you take into account you need to connect to an additional computer (the bruker7t).

Conversion to other formats (DICOM and NIFTI) is available upon request.

3.3 Access to Software

3.3.1 Accessing Scientific Software (quarantine aka modules)

All computer systems have installed a standard suite of desktop productivity software (office, image maniupuation, web browser, etc.). If a readily available productivity software package is not installed and you wish to use it, open a support ticket to request installation.

Scientific sofware is deployed across all computers in the platform via shared network drive, ensuring the same version of software is run on all machines during any kind of cluster processing. To make multiple versions of software available, software is isolated in seperate installation directories and access is managed via the `module` system. The `module` system allows for multiple versions to live side-by-side, for dependencies between software to be specified, and for conflicting versions to be specified.

On which machines is the quarnantine available?

The software quarantine (aka modules) is available on workstation systems (`cicwsNN`) and the compute nodes (`ciccsNN`).

It is not available on the login or storage servers as scientific processing should not take place there.

The obtain a list of available software, run `module avail`, below is an example generated on 2021-06-09, the format of the naming below is <modulename>/<moduleversion>:

```
------------------------------------------------------------------------------------------
↪--- /opt/quarantine/modules ------------------------------------------------------------
↪---------------------------------
ACVD/20161025                                          jmrui/6.0beta                  ␣
↪                  pyminc/0.41
AFNI/2014.09.08.21.47EDT                               mango/3.8                      ␣
↪                  pyminc/0.42b
AFNI/2017.03.30.07.57EDT                               mango/4.0.1                    ␣
↪                  pyminc/0.46
AllenGeneMNI/dev                                       matlab/R2012a                  ␣
↪                  pyminc/0.47
anaconda/2.0.1                                         matlab/R2014a                  ␣
↪                  pyminc/0.48
anaconda/2019.03-python3                               matlab/R2015aSP1               ␣
↪                  pyminc/0.49
anaconda/2.1.0                                         matlab/R2016a                  ␣
↪                  pyminc/0.51
anaconda/2.3                                           matlab/R2018b                  ␣
↪                  PyQC/1.0
anaconda/2.5                                           mi-brain/2020.04.09            ␣
↪                  qbatch/2.1.3
anaconda/4.1.1                                         MIDER/v2                       ␣
↪                  qbatch/2.1.5
anaconda/4.2.0-python3                                 minclaplace/
↪a280379ff13d8265a9ca342cfa6f4510c24d26b8 qbatch/2.2
anaconda/4.3.0-python2.7(default)                      minc-stuffs/0.1.12^minc-toolkit-1.9.
↪10                quarantine
anaconda/5.0.1-python3                                 minc-stuffs/0.1.12^minc-toolkit-1.9.
↪11                R/3.1.1
anaconda/5.1.0-python3                                 minc-stuffs/0.1.15a^minc-toolkit-1.
↪9.11               R/3.2.1
anaconda/miniconda3                                    minc-stuffs/0.1.16^minc-toolkit-1.9.
↪11                R/3.2.4
ANTs/20190211                                          minc-stuffs/0.1.20^minc-toolkit-1.9.
↪11                R/3.3.3
ANTs/20191007                                          minc-stuffs/0.1.20^minc-toolkit-1.9.
↪15                R/3.4.0
ANTs/20191119                                          minc-stuffs/0.1.21^minc-toolkit-1.9.
↪15                R/3.5.0
ANTs/20200104                                          minc-stuffs/0.1.22^minc-toolkit-1.9.
↪16                R/3.5.1
ANTs/20200227                                          minc-stuffs/0.1.24^minc-toolkit-1.9.
↪16                remotemesa/1.0
ANTs/20200410                                          minc-stuffs/0.1.24^minc-toolkit-1.9.
↪17                R-extras/3.1.1
ANTs/2.1.0                                             minc-stuffs/0.1.4^minc-toolkit-1.0.
↪01                 R-extras/3.2.1
ANTs/2.1.0rc3                                          minc-stuffs/0.1.7^minc-toolkit-1.0.
↪01                 R-extras/3.2.3
ANTs/2.2                                               minc-stuffs/0.1.9^minc-toolkit-1.9.
↪10                 R-extras/3.2.4
```

(continues on next page)

(continued from previous page)

```
ANTs/2.3.1                                          minc-toolkit/1.0.01                ␣
↪                 R-extras/3.3.3
bpipe/0.9.8.6                                       minc-toolkit/1.0.04                ␣
↪                 R-extras/3.4.0
bpipe/0.9.8.7                                       minc-toolkit/1.0.07                ␣
↪                 RMINC/1.2.4.5^minc-toolkit-1.0.01^R-3.1.1
bpipe/0.9.9                                         minc-toolkit/1.9.10                ␣
↪                 RMINC/1.2.4.7^minc-toolkit-1.0.01^R-3.1.1
bpipe/0.9.9.2                                       minc-toolkit/1.9.10.1              ␣
↪                 RMINC/1.2.4.9^minc-toolkit-1.9.10^R-3.2.1
bpipe/0.9.9.3                                       minc-toolkit/1.9.11                ␣
↪                 RMINC/1.3.0.0^minc-toolkit-1.9.10^R-3.2.1
bpipe/0.9.9.4                                       minc-toolkit/1.9.15                ␣
↪                 RMINC/1.3.0.0^minc-toolkit-1.9.11^R-3.2.4
bpipe/0.9.9.5                                       minc-toolkit/1.9.16                ␣
↪                 RMINC/1.4.0.0^minc-toolkit-1.9.11^R-3.2.4
bpipe/0.9.9.6                                       minc-toolkit/1.9.17                ␣
↪                 RMINC/1.4.0.3^minc-toolkit-1.9.11^R-3.2.4
BrainExplorer/2                                     minc-toolkit-extras/1.0            ␣
↪                 RMINC/1.4.1.1^minc-toolkit-1.9.11^R-3.2.4
braingl/0.0-1                                       miniconda/2020-03                  ␣
↪                 RMINC/1.4.2.0^minc-toolkit-1.9.11^R-3.2.4
braingl/dev                                         MIPAV/6.0.1                        ␣
↪                 RMINC/1.4.3.0^minc-toolkit-1.9.11^R-3.2.4
brain-view2/1.0^minc-toolkit-1.0.01                 MIPAV/7.1.1                        ␣
↪                 RMINC/1.4.3.1^minc-toolkit-1.9.11^R-3.2.4
brain-view2/1.0^minc-toolkit-1.9.10                 MIPAV/7.2.0                        ␣
↪                 RMINC/1.4.3.2^minc-toolkit-1.9.11^R-3.2.4
Bru2Nii/1.0.20170707                                mni.cortical.statistics/0.9.4      ␣
↪                 RMINC/1.4.3.3^minc-toolkit-1.9.11^R-3.2.4
c3d/2015.06.22                                      mountainlab/20171005               ␣
↪                 RMINC/1.4.3.4^minc-toolkit-1.9.11^R-3.2.4
CIVET/1.1.10                                        mricrogl/12-February-2016          ␣
↪                 RMINC/1.4.4.0^minc-toolkit-1.9.15^R-3.3.3
CIVET/1.1.12                                        mricrogl/12-June-2015              ␣
↪                 RMINC/1.5.0.0^minc-toolkit-1.9.15^R-3.4.0
CIVET/2.1.0                                         mricron/1JUNE2015                  ␣
↪                 RMINC/1.5.1.0^minc-toolkit-1.9.16^R-3.4.0
CIVET-extras/1.0                                    mricron/22DEC2015                  ␣
↪                 RMINC/1.5.2.0^minc-toolkit-1.9.16^R-3.5.0
cmake/3.10.2                                        MRO/3.2.3                          ␣
↪                 RMINC/1.5.2.1^minc-toolkit-1.9.16^R-3.5.0
cmake/3.13.1                                        mrtrix3/3.0_rc3_30c24e3            ␣
↪                 RMINC/1.5.2.1^minc-toolkit-1.9.16^R-3.5.1
dcm2niix/1.0.20171215                               mrtrix3/d861bbe6                   ␣
↪                 RMINC/1.5.2.2^minc-toolkit-1.9.17^R-3.5.1
dcm2niix/1.0.20181125                               MVGC/1.0                           ␣
↪                 RMINC/1.5.2.3^minc-toolkit-1.9.17^R-3.5.1
dcm2niix/1.0.20190902                               niak/v0.13.5                       ␣
↪                 rstudio/0.98.1103
dcm2niix/1.0.20200331                               niftimatlib/1.2                    ␣
↪                 rstudio/0.99.491
```

(continues on next page)

(continued from previous page)

```
DKE/2015.10.28                                  NODDI_toolbox/0.9                ␣
↪                  rstudio/0.99.896
DKE-ft/2015.10.26                               octave/4.0.2                     ␣
↪                  rstudio/0.99.903
fmristat/2006.06.06                             paraview/5.0.0                   ␣
↪                  rstudio/1.0.136
freesurfer/5.3.0                                PLINK/1.0.7                      ␣
↪                  rstudio/1.0.143
freesurfer/6.0                                  PLINK/201502                     ␣
↪                  rstudio/1.1.383
FSL/5.0.6                                       PLS/6.1311050                    ␣
↪                  rstudio/1.1.453
FSL/5.0.7                                       PLS/6.15                         ␣
↪                  rstudio/1.1.463
FSL/5.0.8                                       pvconv/0.57                      ␣
↪                  SGE-extras/1.0
FSL/5.0.9                                       pycharm/2017.2.1                 ␣
↪                  shapeit/2r790
FSL/6.0.2                                       pydpiper/1.10                    ␣
↪                  singularity/2.6.1
gcc/4.1.2                                       pydpiper/1.11                    ␣
↪                  snptest/2.5
gift/4.0a                                       pydpiper/1.13.1                  ␣
↪                  SPAMS/2.5
git/2.1.0                                       pydpiper/1.14-beta1              ␣
↪                  SPM/12b_r6080
git/2.3.0                                       pydpiper/1.14-beta2              ␣
↪                  SPM/12b_r6225
gradunwarp/1.0.2                                pydpiper/1.15                    ␣
↪                  SPM/8_r5236
GreatApes/dev                                   pydpiper/1.18                    ␣
↪                  SPM12/r6080
GTOOL/0.7.5                                     pydpiper/2.0.1                   ␣
↪                  SPM12/r6225
HCP_ConnectomeWorkbench/1.3.2                   pydpiper/2.0.10                  ␣
↪                  SPM12/r6685
ilastik/1.3.3post3                              pydpiper/2.0.12                  ␣
↪                  SPM12/r7219
ILT/4bc2d846                                    pydpiper/2.0.13                  ␣
↪                  SPM8/r5236
impute/2.3.2                                    pydpiper/2.0.3                   ␣
↪                  SPM8/r6313
itksnap/3.2.0                                   pydpiper/2.0.5                   ␣
↪                  torch/d3b017d2aba6a865f44caf1b8e5f07996c26d4c0
itksnap/3.4.0-minc                              pydpiper/2.0.6                   ␣
↪                  webp/1.1.0
itksnap/3.6.0                                   pydpiper/2.0.8
java/8u171-32bit                                pyminc/0.4
```

To load a module, use the load command, `module load <modulename>` which will load the deafault version (D) if it is specified in the module system, otherwise it will load the latest version. To load a specific version of a module, specify the version during loading, `module load <modulename>/<moduleversion>`.

Module Dependencies

If you try to load a module which has dependencies you have not yet satisfied, you will receive an error that will specify the dependencie and modules available that would satisfy it, each missing depndency stops the module loading so this may occur several times as you load all dependencies:

```
$ module load pyminc
pyminc/0.51(7):ERROR:151: Module 'pyminc/0.51' depends on one of the module(s) 'anaconda/
↪miniconda3 anaconda/5.1.0-python3 anaconda/5.0.1-python3 anaconda/4.3.0-python2.7␣
↪anaconda/4.2.0-python3 anaconda/4.1.1 anaconda/2019.03-python3 anaconda/2.5 anaconda/2.
↪3 anaconda/2.1.0 anaconda/2.0.1'
pyminc/0.51(7):ERROR:102: Tcl command execution failed: prereq anaconda
```

CHAPTER

FOUR

RESEARCH METHODS

4.1 Sharing data

Publishing research data is an important step in ensuring that your research outputs are following the **F**indable **A**ccessible **I**nteroperable **R**eusable (FAIR) principles for scientific data management. Although making your work accessible to others involves costs (effort and time, mostly), it considerably increases the reach of your work and thus also benefits your academic career. You can make the process of sharing your work, data, and code substantially easier if you plan it right from the start of your project. A data management plan, good organization of your data, and good documentation of your actions will all help make this process much easier, and are also essential for you to be able to reproduce and understand your own work. In practice, your closest collaborator is often you from 6 months ago!

How to read this document

This document is organized around a series of questions that you can ask yourself when planning to share data. For each question, we provide you with good external resources that provide answers.

4.1.1 Check data sharing constraints

Your first step should be to ensure that you **can** in fact share the raw or derivative data you intend to share. Here are some reasons you may be constrained in your ability to share your data. Most of the time this **does not mean that you cannot share any data**! Often it just means that someone has already made a decision on **how** you should share data.

Here are some scenarios that could suggest that there are constraints on how you share data:

- there is already a **data management plan** that describes how work derived from a dataset or project are going to be shared. If a data management plan exists, you should follow it!
- the data you are working on involves **human participants**. You are responsible to ensure that the privacy of your participants is protected and identifiable information is not shared.
- you are reusing data that someone else has acquired. These data may have a **license or data usage agreement** that you have agreed to and that you may be bound by.

Note: Data management plans are increasingly becoming a requirement by funding agencies and if your group. If your group needs advice on this, you can reach out to the DNP or the Douglas Open Science group.

These are questions your principal investigator, lab manager, or someone responsible for data curation in your group may be able to help answer.

Additional resources for data sharing constraints

- Tri-Council policy on privacy and confidentiality
- Slides by McGill on data sharing constraint considerations

4.1.2 What is the purpose for sharing the data

All shared data should be well documented, well organized, and follow community accepted data standards and file formats. But depending on the main purpose for sharing data, you may make different decisions on how and what to share. Generally, there are two options:

1. You are sharing **a dataset** that you have acquired (e.g. raw data) or that you have processed (derived data). Your goal here is that someone else might want to find these data useful, will reuse them and give you credit for your work.
2. You are sharing **data as part of an empirical manuscript submission**. Your goal here is to support your findings and figures with the data that support your analyses - and possibly to fulfill the requirements of the journal.

The main difference between these two options is that option 2. is a special case of 1. That is, to share data in support of a publication, you should follow the same steps and rigour in sharing a standalone dataset but you will also want to consider providing your data in a way that facilitates reproduction of the analyses described in your publication.

Sharing a dataset for re-use

Consider turning your shared dataset into a publication

All shared data should have a DOI and be citable. But there are also dedicated data journals that will allow you to make an in depth description of your data and clearly show your contibution. Here is an overview of some data sharing journals.

This is a major scientific contribution that stands on its own. The main goal here is to provide your data in as standardized a form and with as detailed a description as possible.

To prepare your data for sharing, you should:

- provide a detailed explanation of the data in a README file.
  - Include information on acquisition, processing, inclusion, and exclusion criteria
  - Turing way documentation guidelines
  - How to make data accessible
- For derivative data, include a reference to the raw data detailed descriptions of any processing you have done
  - Provide a reference to the raw data that your dataset is derived from (publication, DOI, or URL)
  - you should include the exact sequence of processing commands that were performed
  - ideally this would be in the form of the actual script files you have run
- data files should follow community accepted standards for file formats and naming
  - e.g. BIDS for imaging data
  - where no standards exist or are applicable, make sure that file names are standardized, human readable, and explained in the README file
  - make sure that data files are organized in a hierarchical directory structure with human readable directory names. the directories should be referred to in the README with a brief explanation of their contents

- Share tabular data in a standardized and human readable way
  - Don't mix tabular and non-tabular data (i.e. don't include images or text annotations in an excel spreadsheet)
  - Use common data formats, ideally text based ones like `.csv` (comma separated files) or `.tsv` (tab separated files)
  - Provide the name of variables in the first row of a table (header row)
  - Use human readable variable names (and explain them in data dictionaries)
  - Turing way guidelines on tabular data
- Provide explanations for all variables used, e.g. with data dictionaries
  - data dictionaries are tables that map each variable name to a human readable definition and also provide additional information on things like allowable values, range of values, unit types.
  - data dictionaries are very helpful for re-users of your data!
  - OSF tutorial on how to create data dictionaries

Sharing data in support of a published manuscript

Let's assume you have just finished the manuscript for an exciting research project (congrats) and now you want to share the data that support your central findings and figures together with the manuscript. This is not only a great way to ensure that other can follow and re-use your research that will make it more likely your work will have an impact, many journals are starting to require the sharing of data with publication.

Consider sharing derivative data on their own

If your analyses are based on data that you have derived (e.g. through preprocessing with an image processing pipeline) from an existing dataset, then you should consider to share these derived data as a standalone dataset. You can then refer to the shared derived data in your publication and your publication specific data sharing.

Preparing data to be shared in support of your analyses should follow the same general principles as when you are *sharing a dataset for re-use*. In contrast to sharing a standalone dataset, the focus here is more on facilitating a reader in reproducing and understanding the specific analyses you have described in your publication.

Consider sharing code with data

Sharing data even with the best annotations and descriptions is not as easy to understand and reproduce as when you also share your analysis code. See this great overview of "The Turing Way"

Here are some questions to get you started

- What analyses should your readers be able to reproduce and have access to?
  - all major figures
  - the central claims of the publication
  - specific pre-processing, inclusion or exclusion steps
  - outlier analyses
  - etc
- What are the steps taken between your raw data and the major outcomes you want to show your readers
  - Here it can be helpful to look at your methods figure

  - Alternatively, draw a flow chart on a piece of paper or write it down as hierarchical text
  - If you do share code with your paper, each step should ideally map to a specific command or script
  - For each step, make a note of the input and output data required
- What intermediate data do you want to share?
  - Some analyses steps take too long, results are too large, or require too many resources for a reader to reasonably re-run them
  - Other times, raw data may not be possible to share due to privacy concerns, whereas derivative summary data may be uncomplicated
  - In these cases it may make sense to just share intermediate data
  - Here the flow-chart will come in very handy to inform you on what data are needed to reproduce a given analysis step
- Make a data sharing plan
  - Based on the previous decisions and information, you should now make a plan on what to share and how
  - Start a text document and write down each data file you want to share and what name you plan to give it and in what directory you plan to store it
  - This can become the foundation for your README file and your data dictionaries
  - This plan will also help you make sure that you are using file names consistently, e.g. across your README file, in your data availability statement, and in your scripts
- Can my readers understand the intermediate data files
  - Make sure that intermediate data also have human readable names
  - They should also be described in the documentation of your shared data
- Do the data really work with my analysis code or descriptions
  - Once you have all of this, you should review it
  - If you have code to be shared with the publication and data, make sure it runs and produces the expected oomes
  - Go over your data sharing plan with a colleague to check if it makes sense or things are missing
  - Ask someone to review the documents to see if they can follow your analyses given the data and documentation

At the end of this process, you should have a directory with description files (README, data dictionaries) and data that you want to share.

4.1.3 What repository should the data be shared on

Once you are clear on what data you want to share and for what purpose, you can make a decision on the data publishing repository you want to use. A data repository primarily serves two purposes:

- to provide the storage space for your data to be hosted (so you don't have to pay for or worry about it)
- mint a **D**igital**O**bject**I**dentifier for your dataset, a permanent record and link that will forever point to your dataset and that others can use to access your data. This is often also required by journals for data accessibility requirements.

The Nature Publishing Group maintains a repository of recommended subfield specific and domain general data repositories that you can take a look at.

Another great resource is the data publishing guideline of the F1000 publishing platform.

A good domain general data repository is Zenodo.org. Zenodo is funded by the CERN, and apart from points for coolness this also means a high level of confidence that it will remain accessible long into the future. We can recommend Zenodo for basic data sharing needs.

Another great platform for data sharing is the **O**pen**S**cience**F**ramework. This is maintained by the Center for Open Science

Often these data repositories allow you to generate a DOI before you publish your data. This can be very helpful if you want to include the DOI in your manuscript submission, but also still want to make changes to the data you intend to share.

4.1.4 What license to share data with

Generally you should share your data with as open and permissive a license as you can. A good starting point are the Creative Common Licenses, e.g. CC BY 4. You can make choices on permissions for commercial use, how data users should cite you and so on. Here is a Creative Commons License selector tool.

There may also be constraints on the type of license you can choose.

- your data may be based on data that already came with a license and you may be bound to re-share under the same terms
- your group, or the project you work on may have a data management plan that specifices a license
- your data repository may require that you use a certain type of license
- the journal you are publishing to may require a certain license

These are good things to check with your PI or lab manager.

Additional resources for selecting a license

- Turing Way data licenses
- Guide to Open Data Licensing
- OSF guide on licenses

4.1.5 Additional resources

- OSF guidelines
- The Turing Way
- Data sharing resources of the McGill library