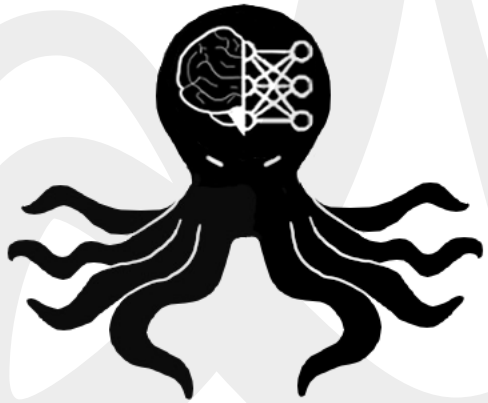
ARAMIS LAB
BRAIN DATA SCIENCE

7 décembre 2022

ClinicaDL

An open-source software for deep learning processing on neuro-imaging data



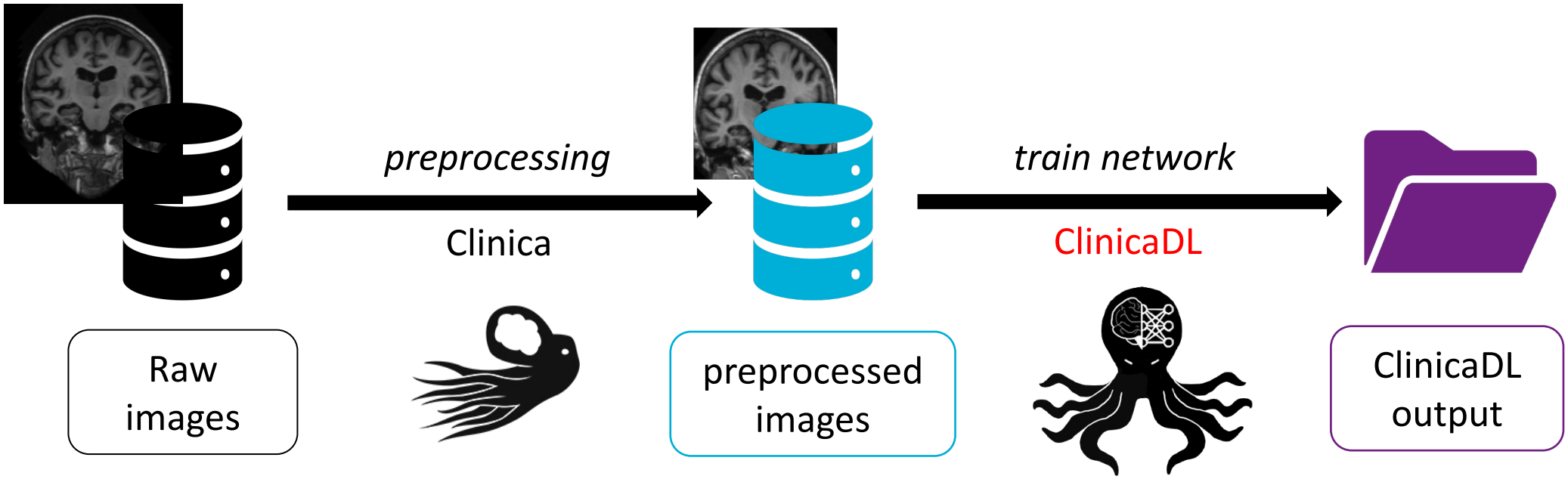
Inserm SORBONNE UNIVERSITÉ Inria ICM Paris Brain Institute Search, find, cure, for you, with you PR[AI]RIE PaRis Artificial Intelligence Research InstitutE cnrs

Software overview

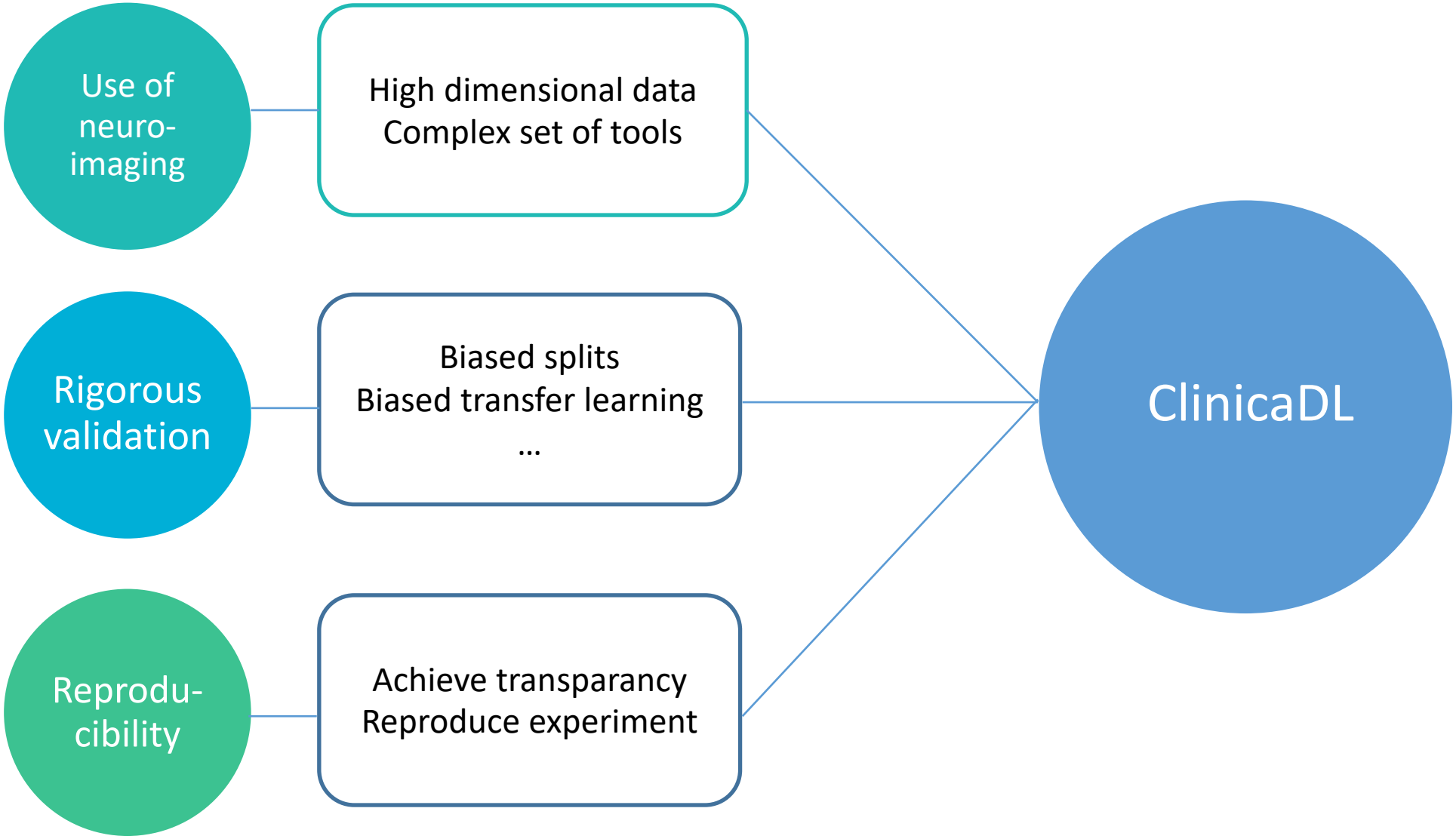
- What is ClinicaDL ?
- The answer to 3 main issues
- What does ClinicaDL look like ?
- An end to end deep learning framework

What is ClinicaDL ?

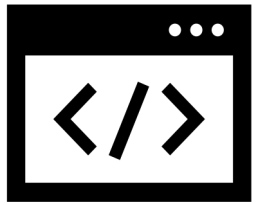
The software ecosystem



The answer to 3 main issues



What does ClinicaDL looks like ?



Command line

or

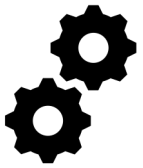
API mode

```
Entrée [ ]: from clinicadl import MapsManager

            maps = MapsManager("/path/to/MAPS")
```

```
clinicadl -h
Usage: clinicadl [OPTIONS] COMMAND [ARGS]...

  ClinicaDL command line.

  For more information please read the doc: https://clinicadl.readthedocs.io/en/latest/ . Source code is available on GitHub: https://github.com/aramis-
  lab/clinicaDL .

  Do not hesitate to create an issue to report a bug or suggest an improvement.

Options:
  --version      Show the version and exit.
  -v, --verbose  Increase logging verbosity.
  -h, --help     Show this message and exit.

Commands:
  extract        Extract Pytorch tensors from nifti images.
  generate       Generation of synthetic dataset.
  interpret      Interpretation of trained models using saliency map method.
  predict        Infer the outputs of a trained model on a test set.
  quality-check  Performs quality check procedure for t1-linear or t1-volume pipelines.
  random-search  Hyperparameter exploration using random search.
  resume         Resume training job in specified maps.
  save-tensor    Save the output tensors of a trained model on a test set.
  train          Train a deep learning model on your neuroimaging dataset.
  tsvtool        Manipulation of TSV files to prepare and manage input data.
```

End to end deep learning framework

CAPS = preprocessed database



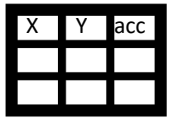
Train model

```
train
resume
random-search
from json
```

PyTorch

MAPS = ClinicaDL output

```
predict
save-tensor
```



Infer results and make prediction

```
interpret
```

Explore and Interpret results

Prepare data and metadata

```
extract
tsvtool
Quality-check
Generate
```

MAPS: Model Analysis and Processing Structure

```
MAPS
├── environment.txt ──────────────────────────────► Libraries versions (pip freeze)
├── fold-1
│   ├── best-loss
│   │   ├── interpretation
│   │   ├── model.pth.tar ────────────────────────► Saved model for fold corresponding to a metric
│   │   ├── validation
│   │   │   ├── prediction.log
│   │   │   ├── validation_metrics.tsv ──────────► Prediction and evaluation metrics
│   │   │   ├── validation_results.tsv
│   │   │   └── tensors ──────────────────────────► Output tensor for reconstruction (AE & VAE)
│   │   ├── test_a
│   │   └── train
│   ├── best-MSE
│   └── training_log ─────────────────────────────► Training logs including tensorbord logs
├── fold-2
├── fold-3
├── groups
│   ├── validation
│   ├── test_a ───────────────────────────────────► Data groups
│   └── train
├── informations.log ─────────────────────────────► Human-readable file with main specifications
└── maps_parameters.json ─────────────────────────► Computer-readable file with all specifications
```

Reproducibility (Store all hyperparameters)


```toml
[Model]
architecture = "default"
multi_network = false

[Architecture]
latent_space_size = 128
feature_size = 1024
n_conv = 4
```

```toml
[Classification]
selection_metrics = ["loss"]

[Regression]
selection_metrics = ["loss"]
label = "age"
loss = "MSELoss"
```

```toml
[Computational]
gpu = true
n_proc = 2
batch_size = 8
evaluation_steps = 0

[Reproducibility]
seed = 0
deterministic = false
```

Toml & Json files


```
preprocessing_dict:
    preprocessing:          "t1-linear"
    mode:                   "image"
    use_uncropped_image:    false
  file_type:
    pattern:                "*space-MNI152NLin2009cSym_desc-Crop_res-1x1x1_T1w.pt"
    description:            "T1W Image registered using t1-linear and cropped (matrix size 169×208×179, 1 mm isotropic voxels)"
    needed_pipeline:        "t1-linear"
  mode:                     "image"
  network_task:             "classification"
caps_directory:             "/gpfswork/rech/bgc/commun/data/labels_list/session_3/ADNI_NIFD_caps_linear.tsv"
tsv_path:                   "/gpfswork/rech/bgc/commun/data/labels_list/session_3/ADNI_NIFD_tsv_linear.tsv"
```

Practical aspect

- Dependencies
- Current state of the project
- Test on the ClinicaDL software package

Data constraints & dependencies

Data constraints

ADNI Alzheimer's Disease Neuroimaging Initiative

OASIS Open Access Series of Imaging Studies

biobank uk

+ TSV JSON Tensor

CAPS ClinicaDL input

Dependency management

```
141  [[package]]
142  name = "clinica"
143  version = "0.7.2"
144  description = "Software platform for clinical neuroimaging studies"
145  category = "main"
146  optional = false
147  python-versions = ">=3.8,<3.11"
148
149  [package.dependencies]
150  argcomplete = ">=1.9.4,<2.0.0"
151  attrs = ">=20.1.0"
152  cattrs = ">=1.9.0,<2.0.0"
153  click = ">=8,<9"
154  click-option-group = ">=0.5,<0.6"
155  colorlog = ">=5,<6"
```

Poetry

Dependencies

scikit learn

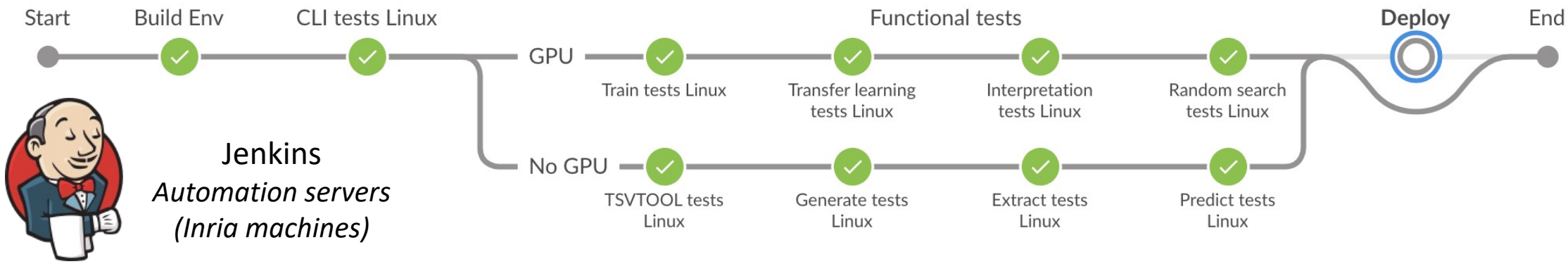
NumPy

PyTorch

ni learn

Test on the ClinicaDL software package

Continuous Integration



Jenkins
Automation servers (Inria machines)

2 machines

GPU - physical

CPU - virtual

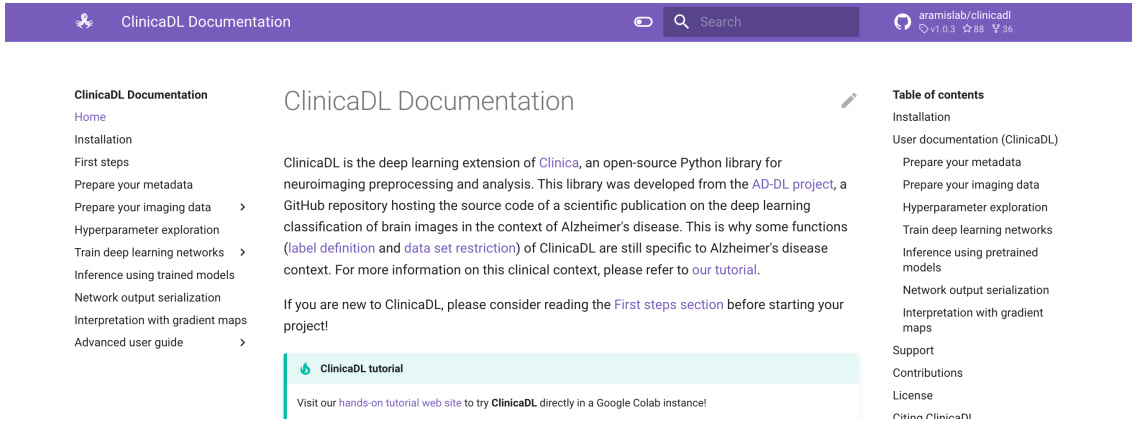
Stock testing data

aws

DVC

Current state of the project

Documentation: https://clinicadl.readthedocs.io/en/latest/



Development:

- Team of 2 engineers and 1 Phd student (ARAMIS)
- External contributions are rare

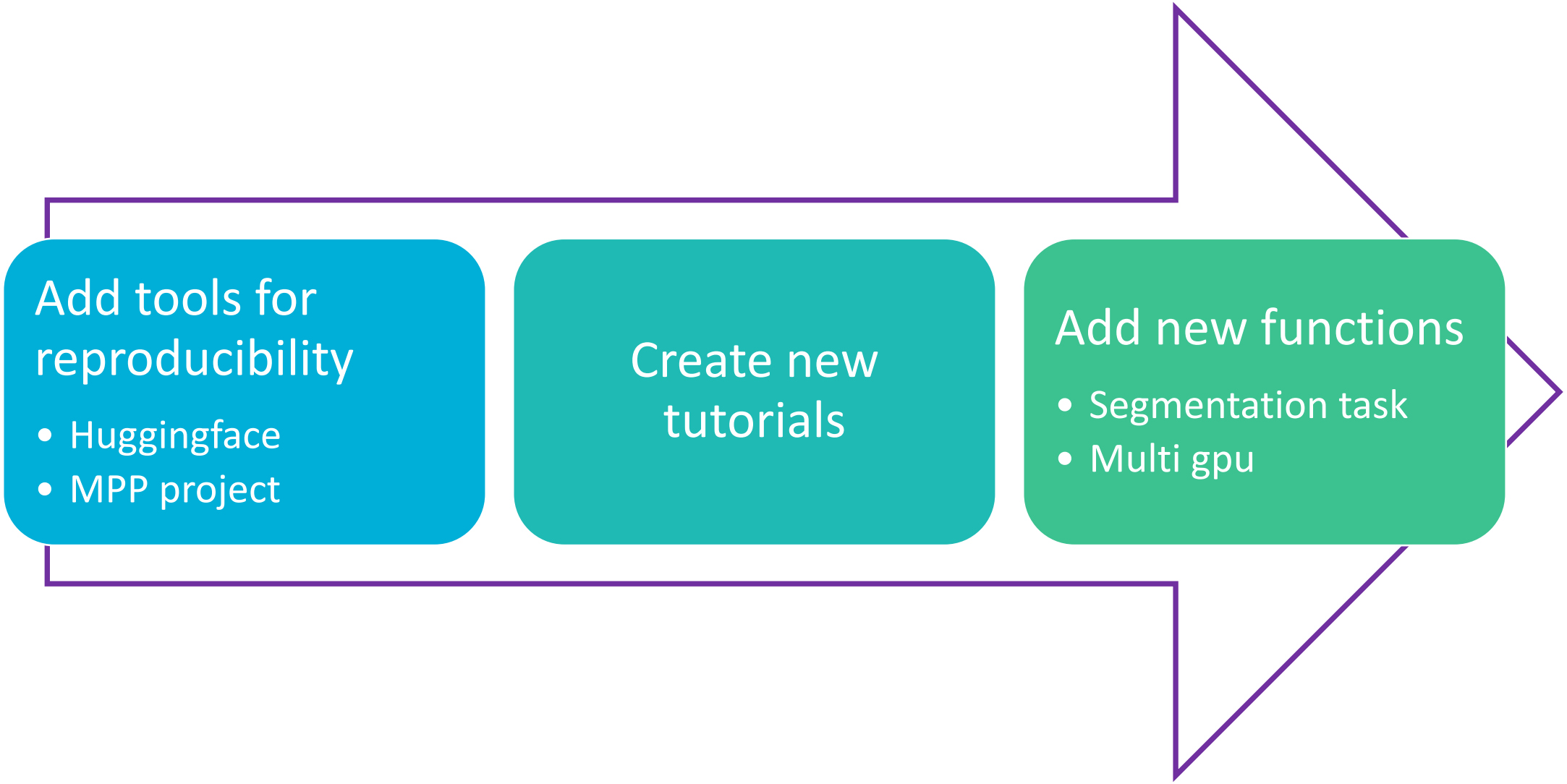
Versionning

15 releases
since august **2020**


```
pip install clinicadl==1.1.1
```

Github: https://github.com/aramis-lab/clinicadl

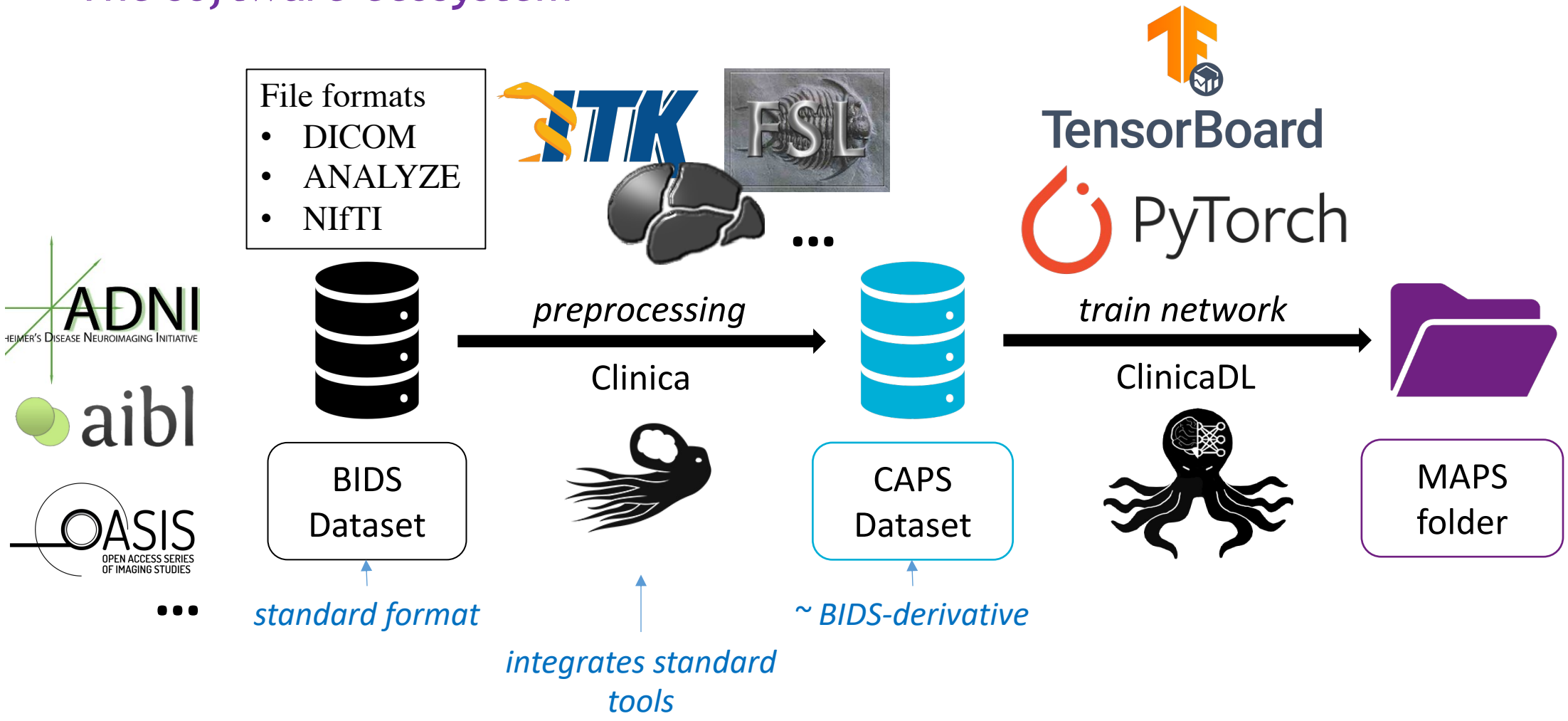
What's next ?


Add tools for reproducibility
• Huggingface
• MPP project
Create new tutorials
Add new functions
• Segmentation task
• Multi gpu

Thanks !
Any questions ?

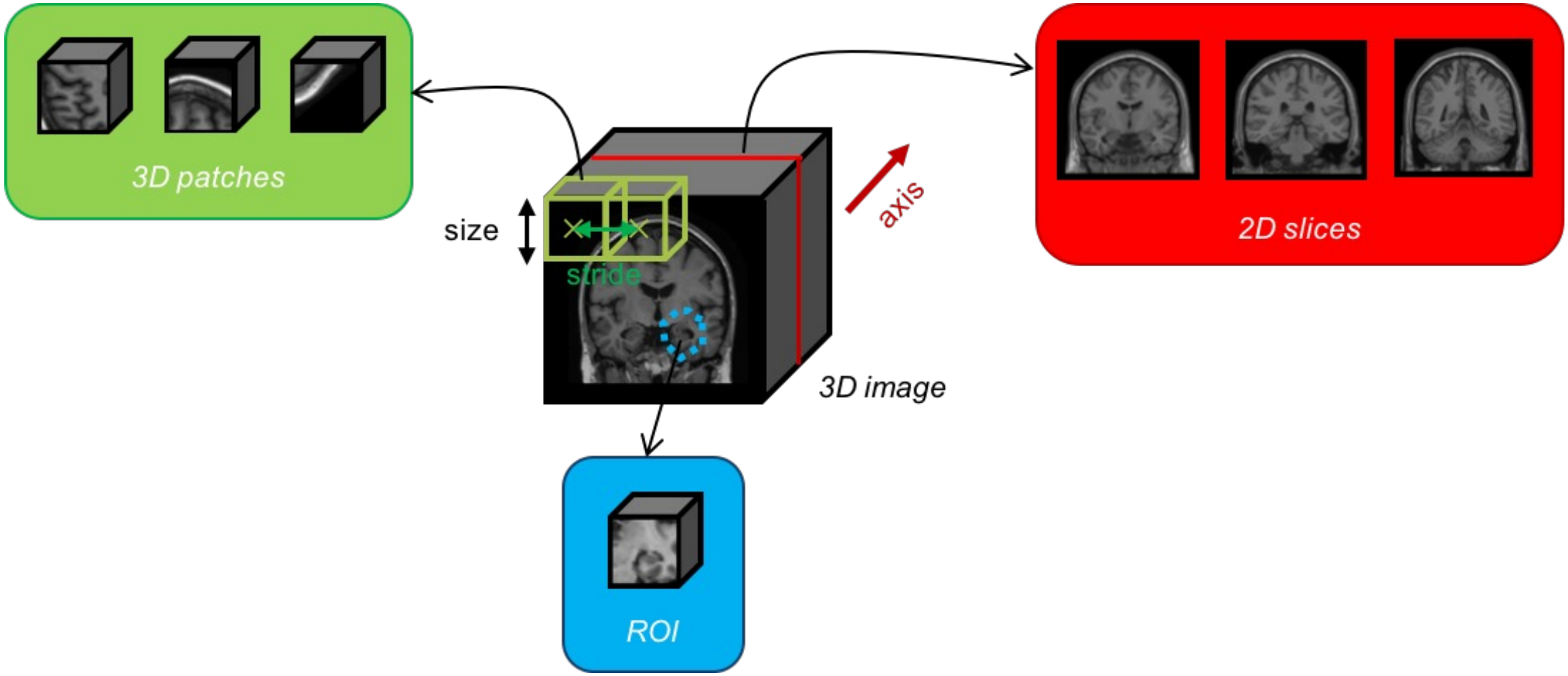
Issue #1: Easy use of neuroimaging data sets

The software ecosystem



Issue #1: Easy use of neuroimaging data sets

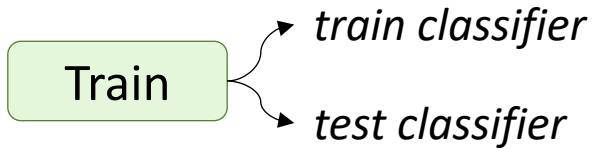
3D data decomposition



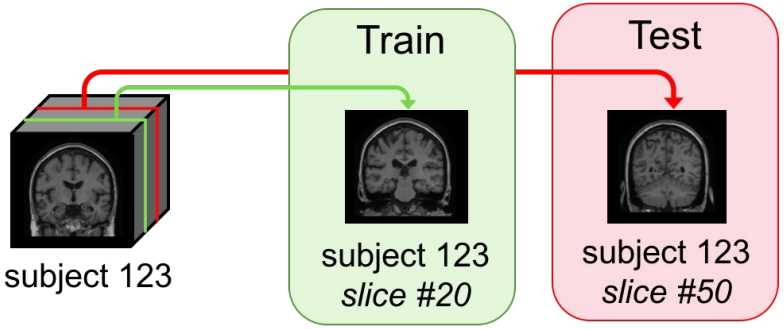
Issue #2: Avoid methodological biases

Scenarios of data leakage

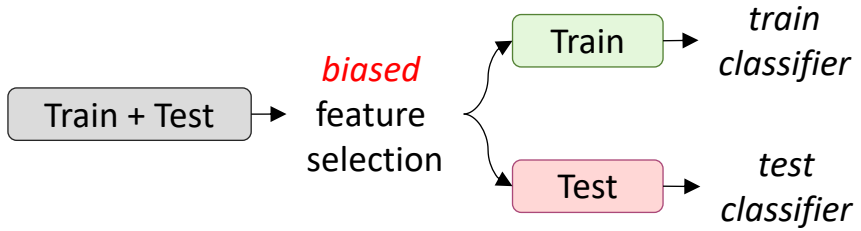
1. Absence of an independent test set



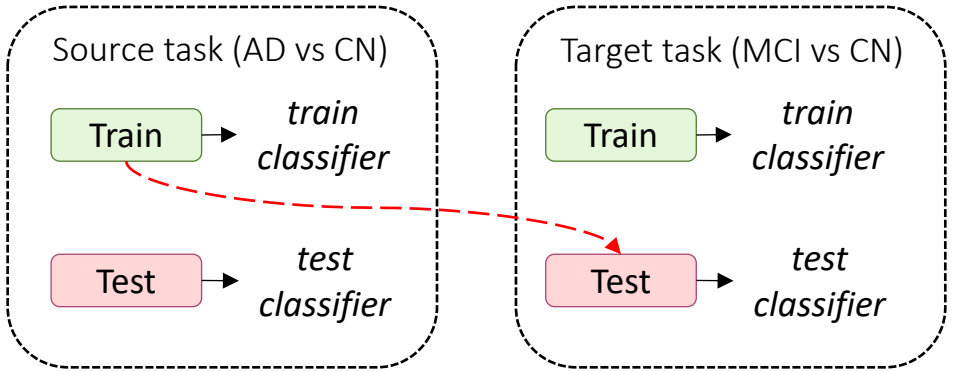
2. Biased split



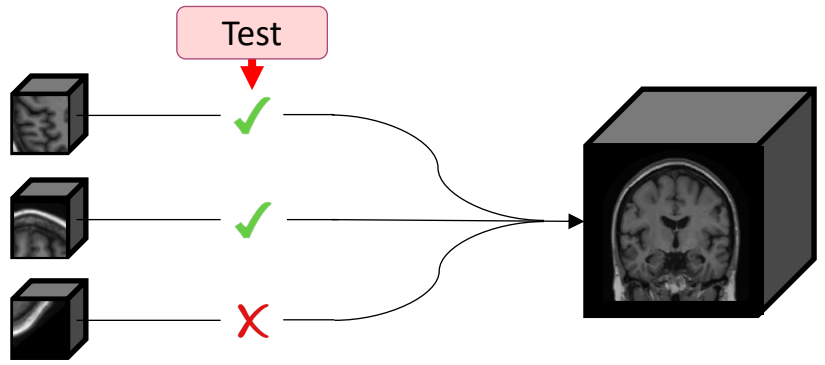
3. Late split



4. Biased transfer learning

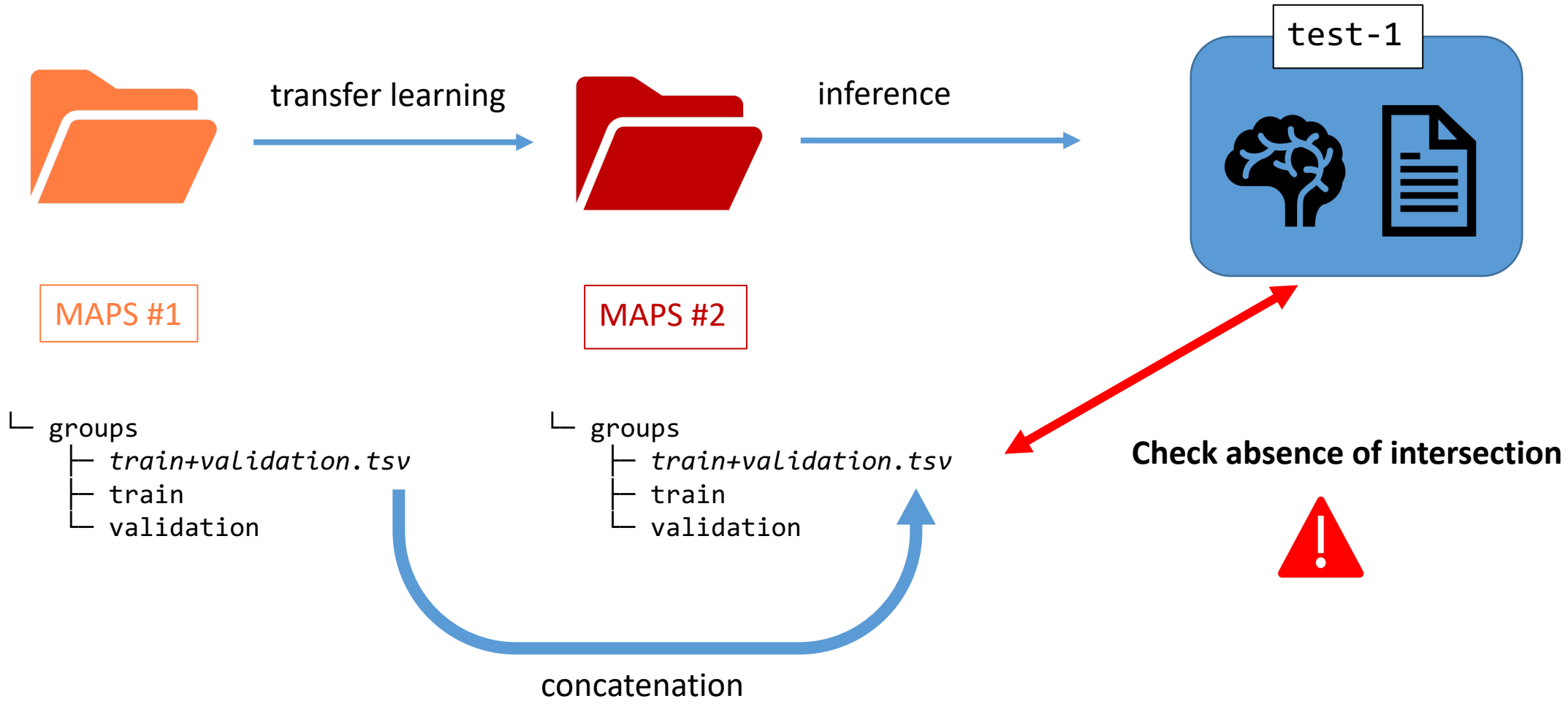


5. Biased ensemble learning



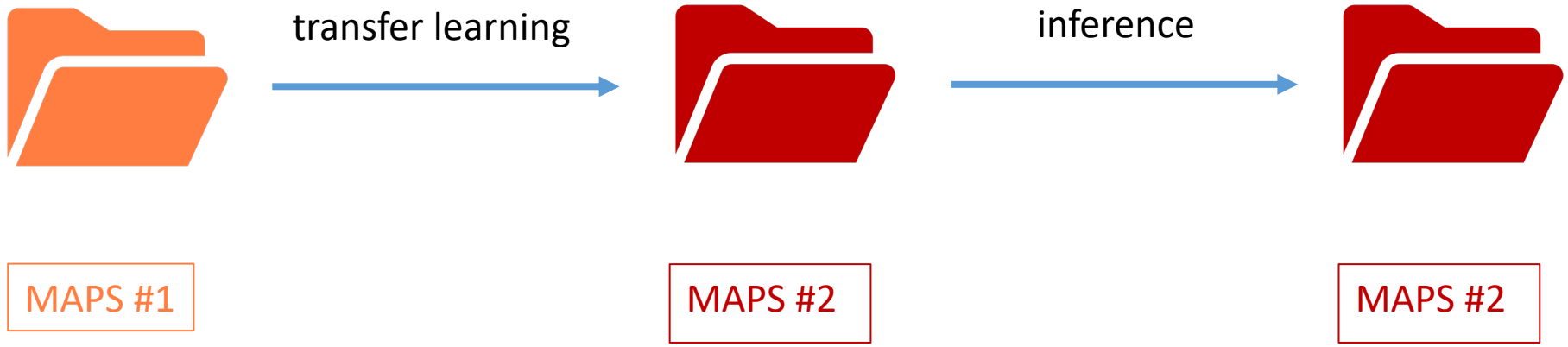
Issue #2: Avoid methodological biases

Example of solution to avoid data leakage



Issue #2: Avoid methodological biases

Example of solution to avoid data leakage




```
└─ groups
   ├─ train+validation.tsv
   ├─ train
   └─ validation
```

```
└─ groups
   ├─ train+validation.tsv
   ├─ train
   └─ validation
```

```
└─ groups
   ├─ train+validation.tsv
   ├─ train
   ├─ validation
   └─ test-1
```

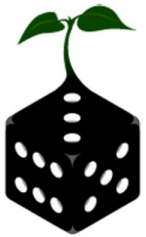
Issue #3: Reproducible deep learning

Reproduce experiments

Store all hyperparameters

```
preprocessing_dict:
    preprocessing:              "t1-linear"
    mode:                       "image"
    use_uncropped_image:        false
  file_type:
    pattern:                    "*space-MNI152NLin2009cSym_desc-Crop_res-1x1x1_T1w.pt"
    description:                "T1W Image registered using t1-linear and cropped (matrix size 169×208×179, 1 mm isotropic voxels)"
    needed_pipeline:            "t1-linear"
  mode:                         "image"
  network_task:                 "classification"
caps_directory:                 "/gpfswork/rech/bgc/commun/data/labels_list/session_3/ADNI_NIFD_caps_linear.tsv"
tsv_path:                       "/gpfswork/rech/bgc/commun/data/labels_list/session_3/ADNI_NIFD_tsv_linear.tsv"
```

Fix random seed



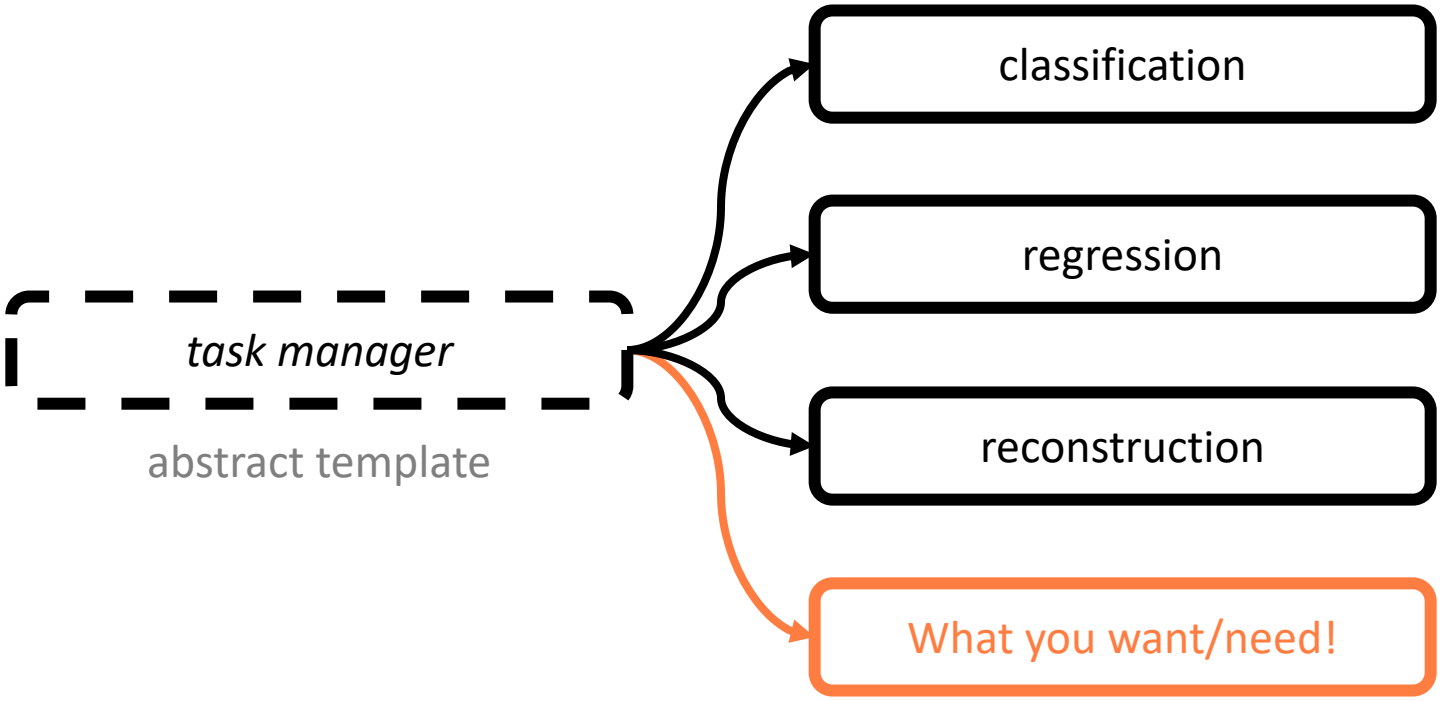
Fix initial state

Data loader ordering and transforms

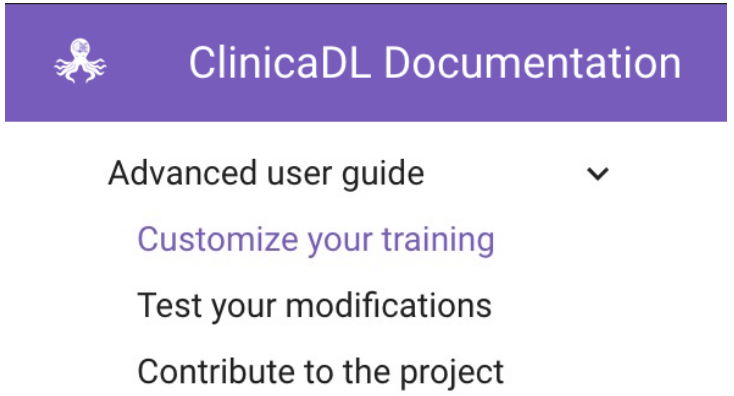
Enforce determinism on GPU

Bonus

Flexibility



More info online!

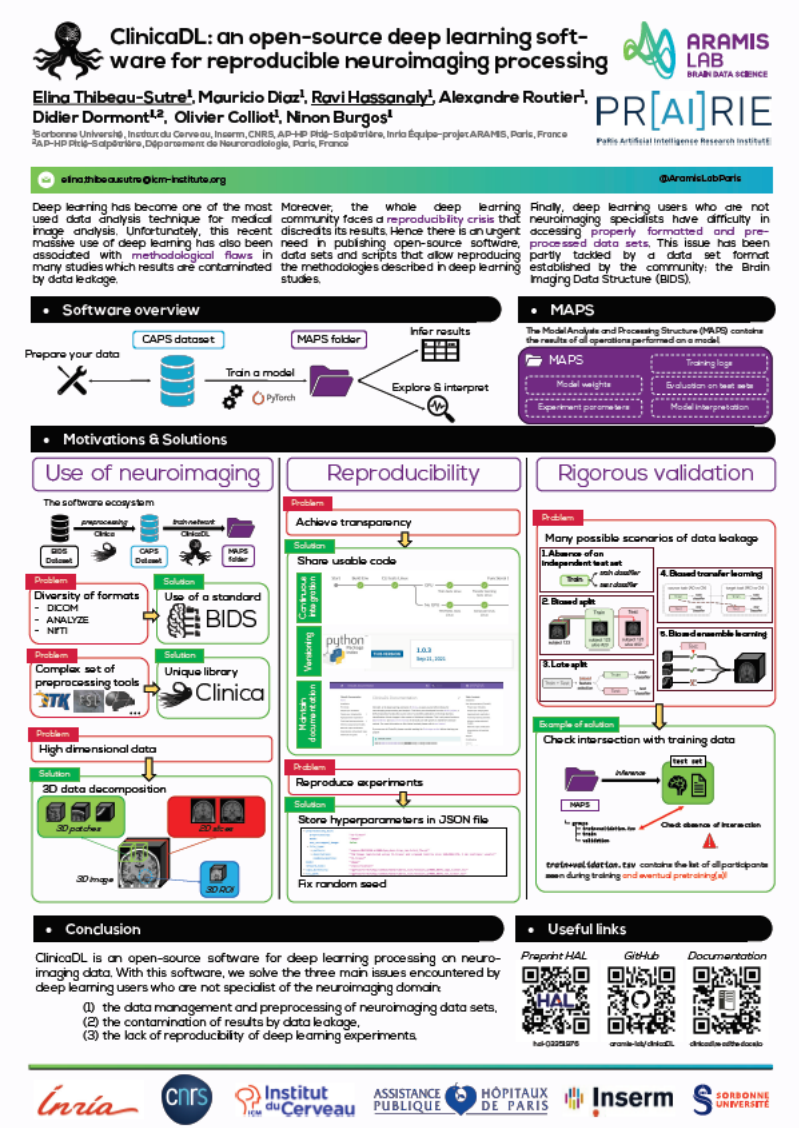


Available for **tasks**, **validation procedure**, **metrics**, **modes** and **architectures**!

Thank you!

ClinicaDL: an open-source deep learning software for reproducible neuroimaging processing

HAL archives-ouvertes.fr hal-03351976



Come and discuss with us at the poster session!