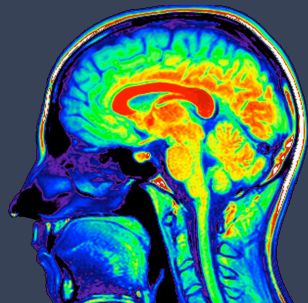


AlstroSight

Viewing neuron-astrocyte  
pharmacology through  
digital sciences



**AIstroSight**

*Inria*

# Summary

- 01 AlstroSight
- 02 HyperFlow

# 01

## The AlstroSight Project



**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:





**Objective:** propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:

?

PyTorch

Lightning AI

scikit learn

HyperFlow

HYDRA

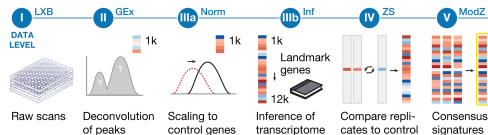
OPTUNA

mlflow



**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:
  - > intra/sub-cellular (molecular)
    - gene expression profiling



LINCS/CMAP gene expression perturbation profiles





**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:
  - > intra/sub-cellular (molecular)
    - gene expression profiling
    - cellular pathways and metabolism



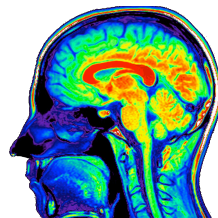


# AIstroSight



**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:
  - > intra/sub-cellular (molecular)
    - gene expression profiling
    - cellular pathways and metabolism
  - > whole brain medical imaging



- primary data
- confidential
- BIDS format

**cermap**  
imagerie du vivant





**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:
  - > intra/sub-cellular (molecular)
    - gene expression profiling
    - cellular pathways and metabolism
  - > whole brain medical imaging
  - > clinical/population data



- primary data (confidential)
- other partners planned
- data mining of online data





**Objective: propose 1 or 2 drug candidates within 4 years for neurological pathologies of neuronal or astrocytic origin**

- Combine biomedical expertise with numerical simulations and AI approaches over a range of spatio-temporal scales:
  - > intra/sub-cellular (molecular)
    - gene expression profiling
    - cellular pathways and metabolism
  - > whole brain medical imaging
  - > clinical/population data
- The project is still in an initial exploratory phase but some tools are already in development.

# 02

## HyperFlow

# HyperFlow

**A small Python framework for machine- and deep-learning**

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.



# HyperFlow

## A small Python framework for machine- and deep-learning

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

- Hydra & Optuna (<https://hydra.cc/>, <https://optuna.org/>)
  - > Unified hyperparameter management via YAML configuration files
  - > Hyperparameter optimization via configurable strategies
  - > Process parallelization



# HyperFlow

## A small Python framework for machine- and deep-learning

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

- Hydra & Optuna (<https://hydra.cc/>, <https://optuna.org/>)
  - > Unified hyperparameter management via YAML configuration files
  - > Hyperparameter optimization via configurable strategies
  - > Process parallelization
- MLflow (<https://mlflow.org/>)
  - > Full experiment tracking with parameter, metric, artifact and model logging for reproducible experiments.
  - > Web and command-line interfaces for comparing and exploiting results.
  - > Deploy models to registries and distribute them.



# HyperFlow

## A small Python framework for machine- and deep-learning

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

- Hydra & Optuna (<https://hydra.cc/>, <https://optuna.org/>)
  - > Unified hyperparameter management via YAML configuration files
  - > Hyperparameter optimization via configurable strategies
  - > Process parallelization
- MLflow (<https://mlflow.org/>)
  - > Full experiment tracking with parameter, metric, artifact and model logging for reproducible experiments.
  - > Web and command-line interfaces for comparing and exploiting results.
  - > Deploy models to registries and distribute them.
- (Pytorch) Lightning support (<https://www.pytorchlightning.ai/>)
  - > Hyperflow makes it easy to use Lightning via a dedicated class in the package.





# HyperFlow

**A small Python framework for machine- and deep-learning**

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

config.yaml

```
experiment:
  name: My Experiment
  description: Placeholder experiment.
  exp_class: experiment:MyExperiment
  params:
    model:
      layer_1_size: 50
      layer_2_size: 20
    training:
      epochs: 50
      batch_size: 100
      learning_rate: 0.05
```



# HyperFlow

## A small Python framework for machine- and deep-learning

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

### config.yaml

```
experiment:  
  name: My Experiment  
  description: Placeholder experiment.  
  exp_class: experiment:MyExperiment  
  params:  
    model:  
      layer_1_size: 50  
      layer_2_size: 20  
    training:  
      epochs: 50  
      batch_size: 100  
      learning_rate: 0.05
```

### Experiment subclass

```
from hyperflow.experiment import Experiment  
  
class MyExperiment(Experiment):  
    ...  
    My example experiment.  
    ...  
  
    # The only necessary method.  
    def __call__(self):  
        # The parameters set in the experiment.params section.  
        params = self.config.experiment.params  
  
        layer_1_size = params.model.layer_1_size  
        # ...  
        epochs = params.training.epochs  
  
        # ...  
        return score
```



# HyperFlow

## A small Python framework for machine- and deep-learning

Hyperflow seamlessly combines several industry-standard tools in a common framework. Users only need a config file and a simple subclass to use it.

**Used by:** members of AlstroSight (recommended for all ML & DL projects in the team)

**Maturity:** beta

- TODO:**
- Improve documentation.
  - Add more examples (sklearn, Pytorch, simpler Lightning example, TensorFlow, etc.).
  - Add unit tests.
  - Release (make the Git repo public, publish to PyPI).

Thank you for your attention!

Questions?