Frederick National Laboratory
for Cancer Research

Installing CANDLE In Other Environments
CANDLE NIH Workshop February 2018

George Zaki [C]

BIDS, FNLCR

February 22nd 2018

.

DEPARTMENT OF HEALTH AND HUMAN SERVICES • National Institutes of Health • National Cancer Institute

Frederick National Laboratory is a federally funded research and development center operated by Leidos Biomedical Research, Inc., for the National Cancer Institute.

Agenda

- Introduction to containers
- Example of hyper-parameter optimization on HPC clusters
- Using CANDLE on Biowulf

Agenda

- **Introduction to containers**
- Example of hyper-parameter optimization on HPC clusters
- Using CANDLE on Biowulf

Why containers? Why singularity?

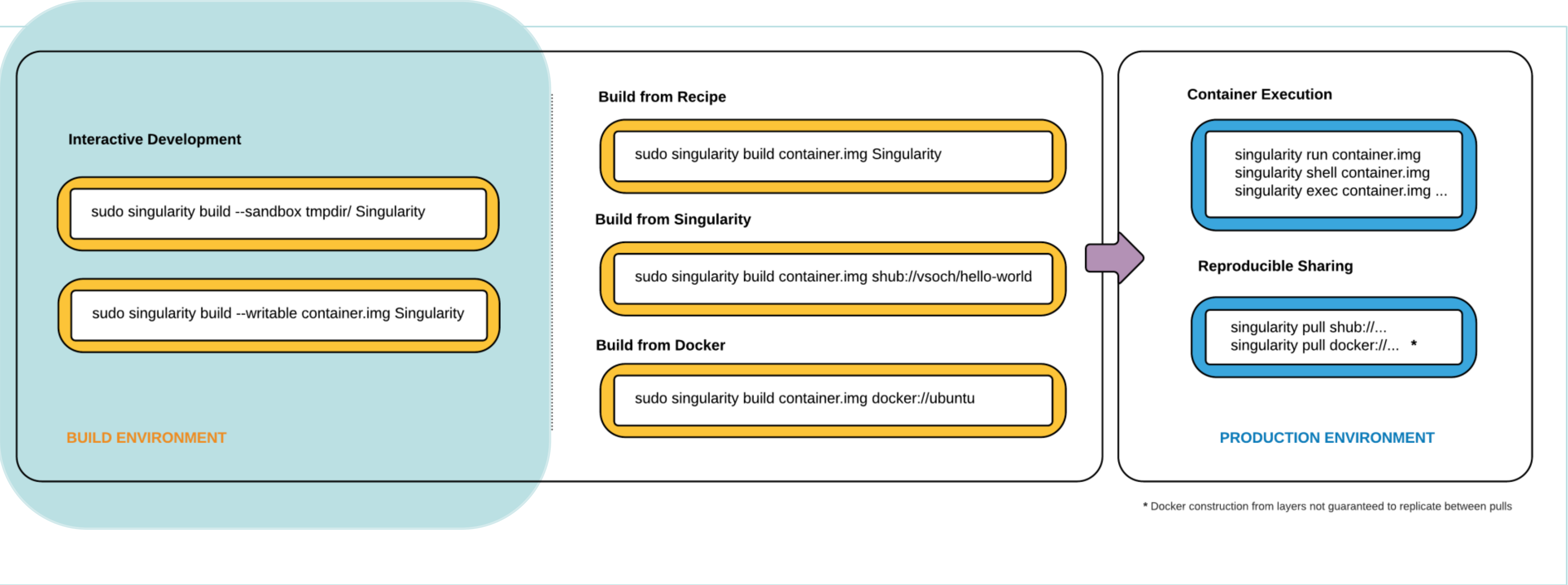
Containers

- Light weight OS virtualization (i.e., user your preferred OS)
- Package entire scientific workflow, and share your code
- No need to for system admin to install software
- Be in control of versioning and updates

Singularity

- Adopted by most HPC systems including Biowulf.
- Can run your MPI jobs
- Support from Biowulf system admins
- Very active user group

Singularity cycle



Interactive Development

```
sudo singularity build --sandbox tmpdir/ Singularity
```

```
sudo singularity build --writable container.img Singularity
```

BUILD ENVIRONMENT

Build from Recipe

```
sudo singularity build container.img Singularity
```

Build from Singularity

```
sudo singularity build container.img shub://vsoch/hello-world
```

Build from Docker

```
sudo singularity build container.img docker://ubuntu
```

Container Execution

```
singularity run container.img
singularity shell container.img
singularity exec container.img ...
```

Reproducible Sharing

```
singularity pull shub://...
singularity pull docker://...   *
```

PRODUCTION ENVIRONMENT

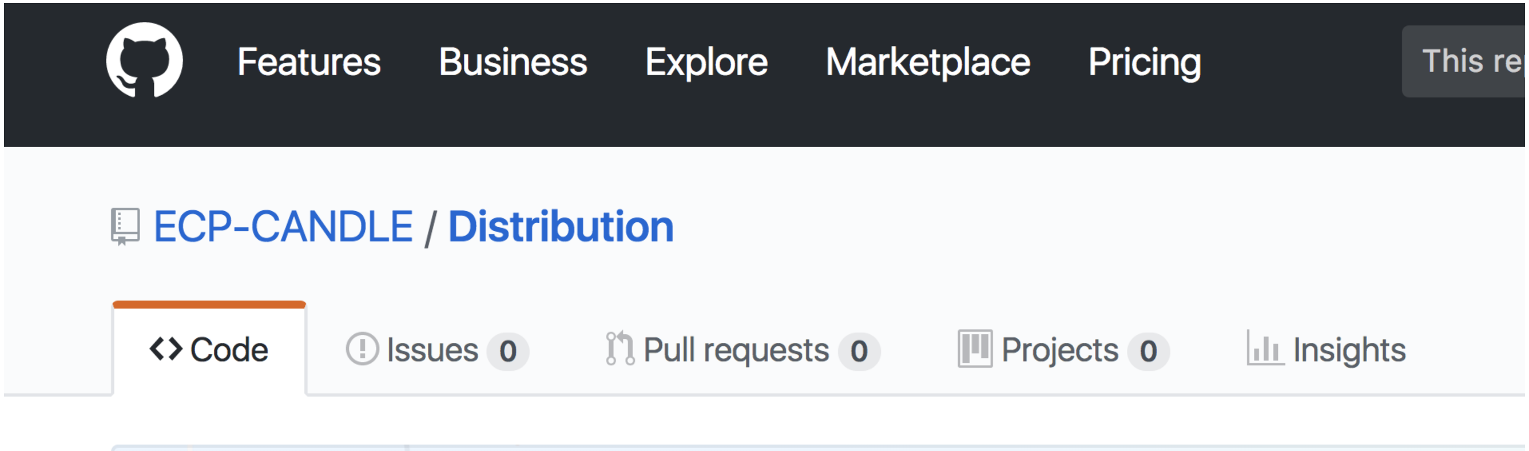
* Docker construction from layers not guaranteed to replicate between pulls

1- Build your singularity image (one file) from a recipe, or a Docker on your local sandbox.
2- Copy the singularity image to the HPC cluster / Cloud cluster.
3- Run on production environment.

singularity.lbl.gov

Building CANDLE Using Singularity

- A singularity recipe that encapsulates all CANDLE dependencies is shared on GitHub.
- Navigate to the Distribution and follow the README.



ECP-CANDLE / **Distribution**

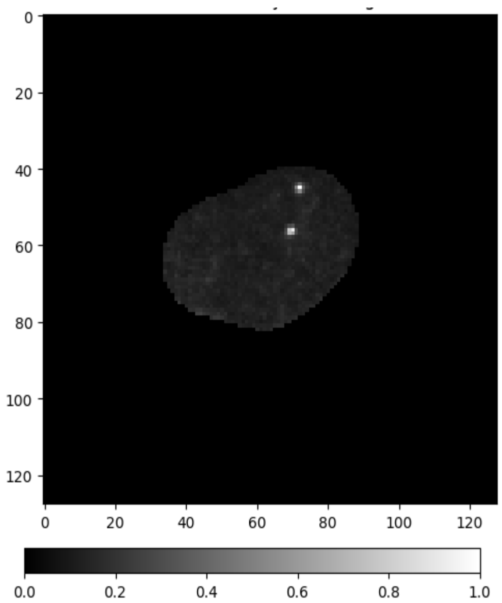
- All you need is to install singularity and mpi on your laptop, cluster, cloud, etc.

Agenda

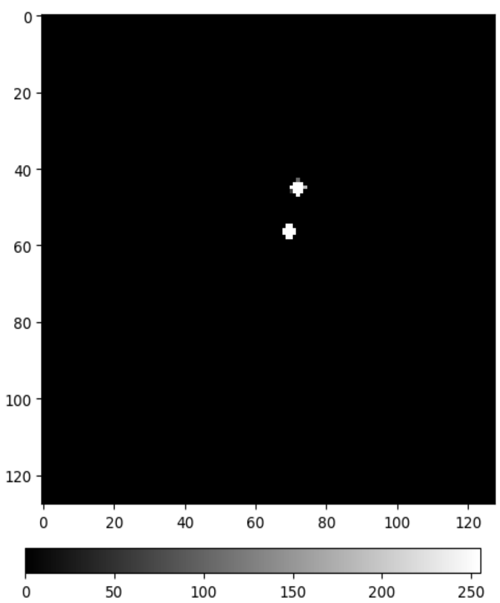
- Introduction to containers
- **Example of hyper-parameter optimization on HPC clusters**
- Using CANDLE on Biowulf

The Problem:
Pixel based segmentation for RNA activities

Input

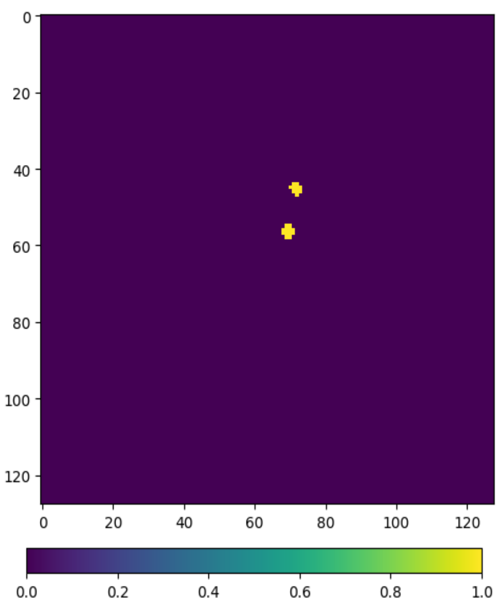


Test FISH Image



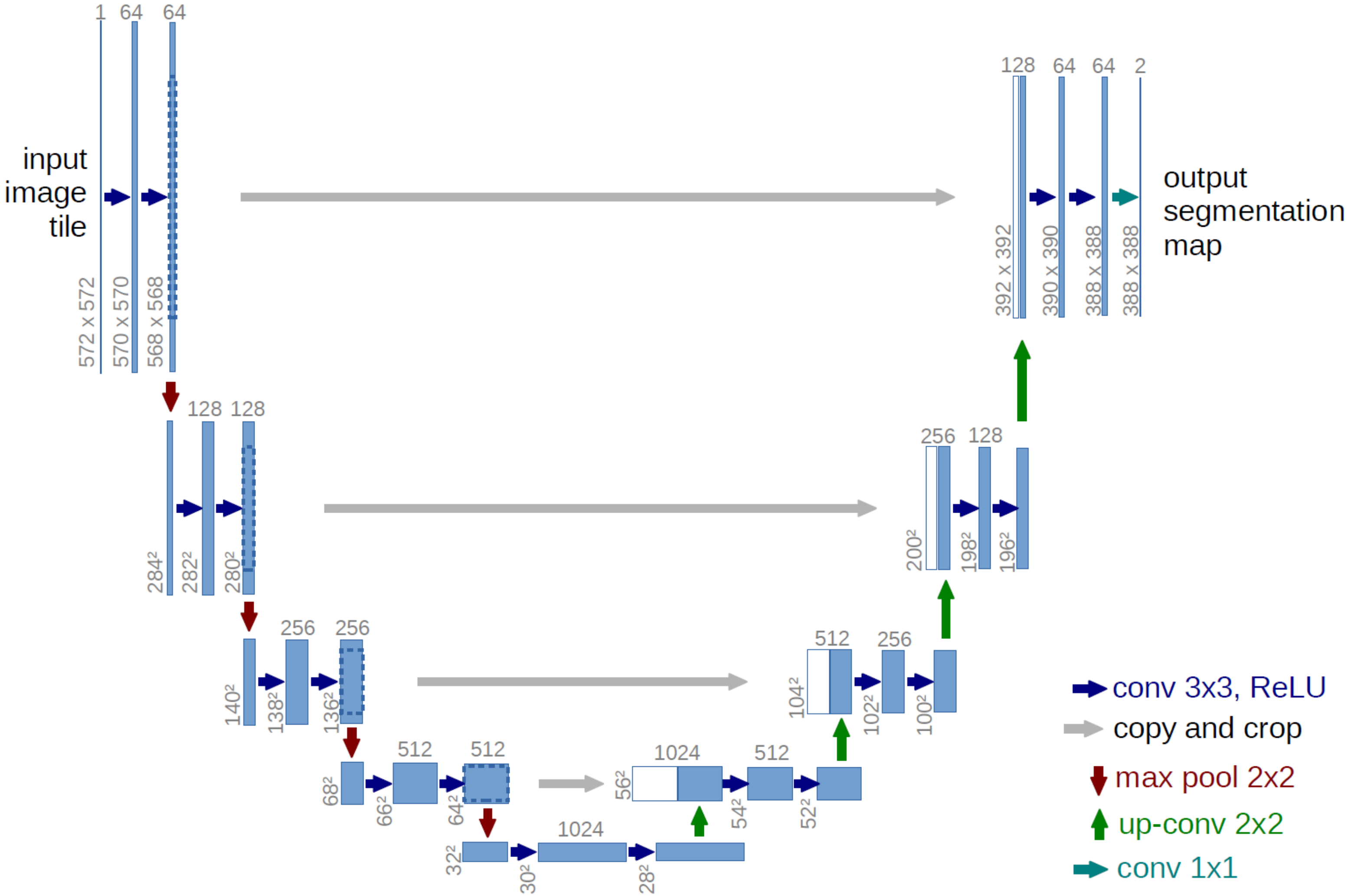
Ground Truth

Output



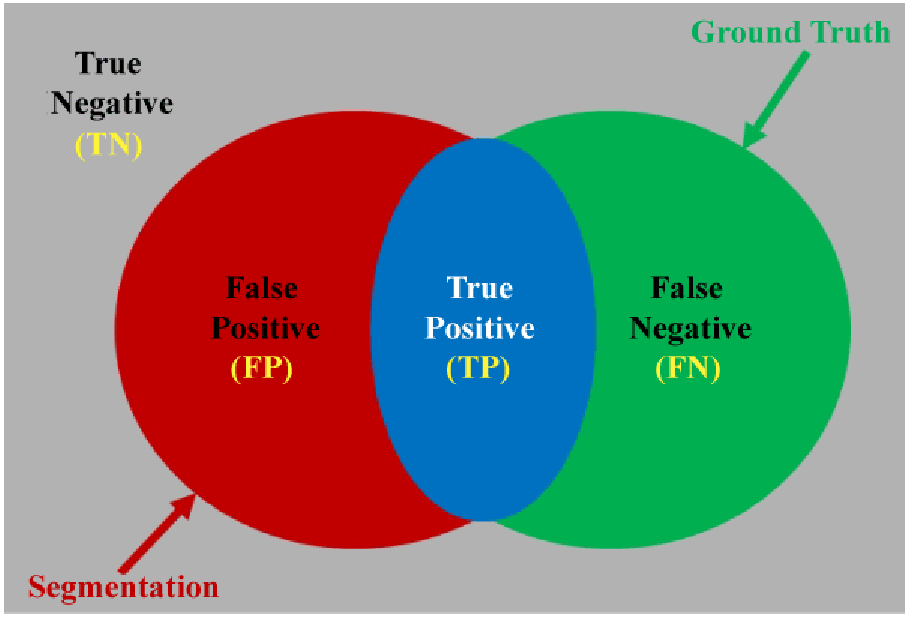
Prediction

UNet – A Fully Convolution Neural Network For Pixel-Based Segmentation



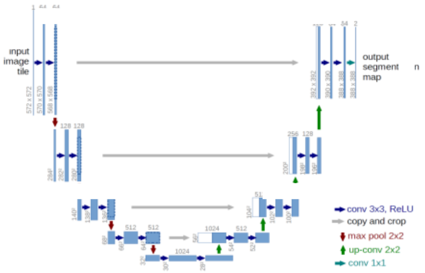
Source: Ronneberger et. al,., MICAAI, 2015

UNet objective function: Dice Coefficient



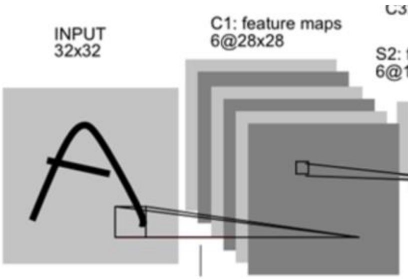
$$\text{Dice Coefficient} = \frac{(2*TP)+1.0}{(FP+TP+FN)+1.0} \in [0,1]$$

Unet Hyper parameters: (288 possible values)



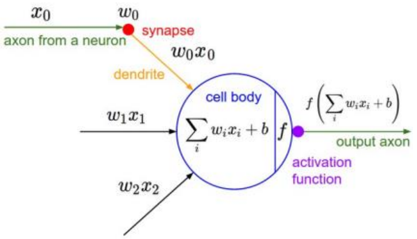
ONLY 2 Levels of Max-Pooling

N_layers = {2,3,4,5}



How many convolution filters?

Num_filters= {16,32,64}



What is the activation function?

Activation= {relu, softmax, tanh

0	0	0
1	1	1
0	0	0

Size of conv filter?

Filter_size = {3x3, 5x5}

Drop out some results to avoid overfitting?

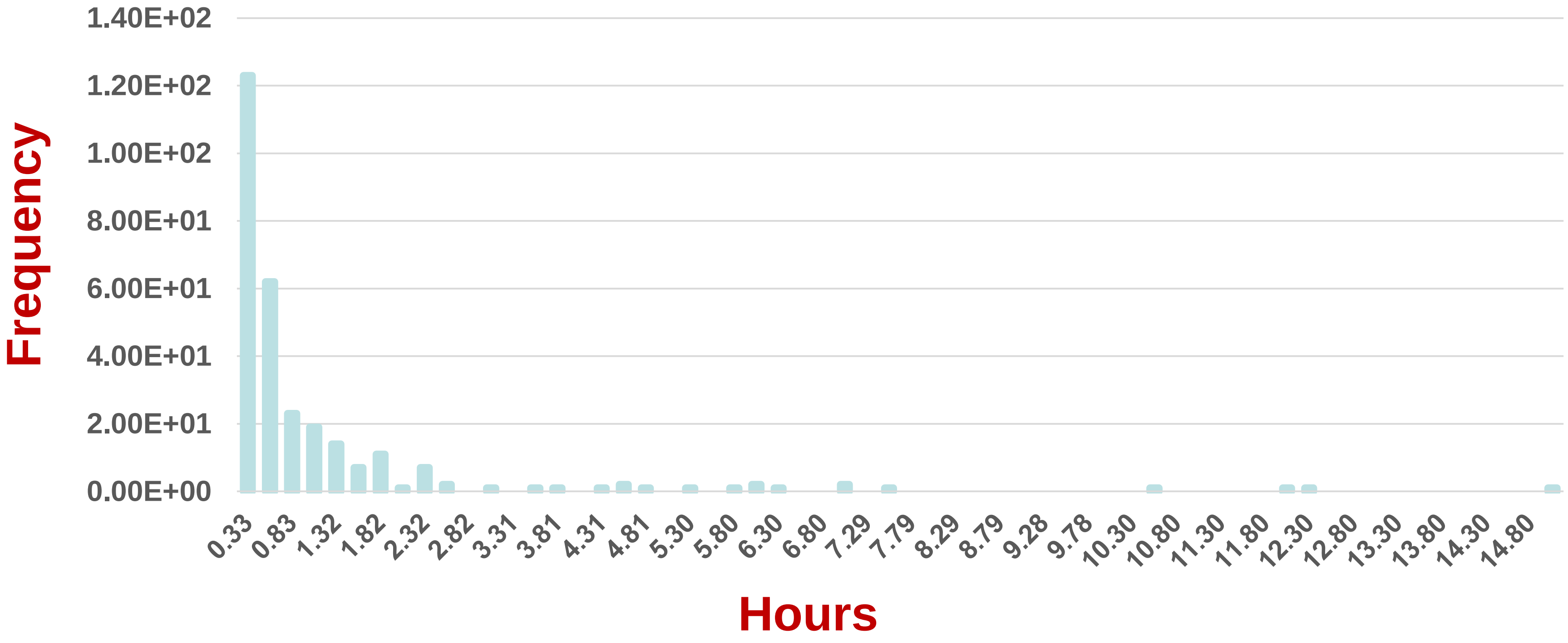
Drop_out = {0, 0.2, 0.4, 0.6, 0.8}

Experiment on Biowulf:

- local swit-t installation
- 16 **K80 GPU** nodes
- 30 hours
- 10GB memory per task

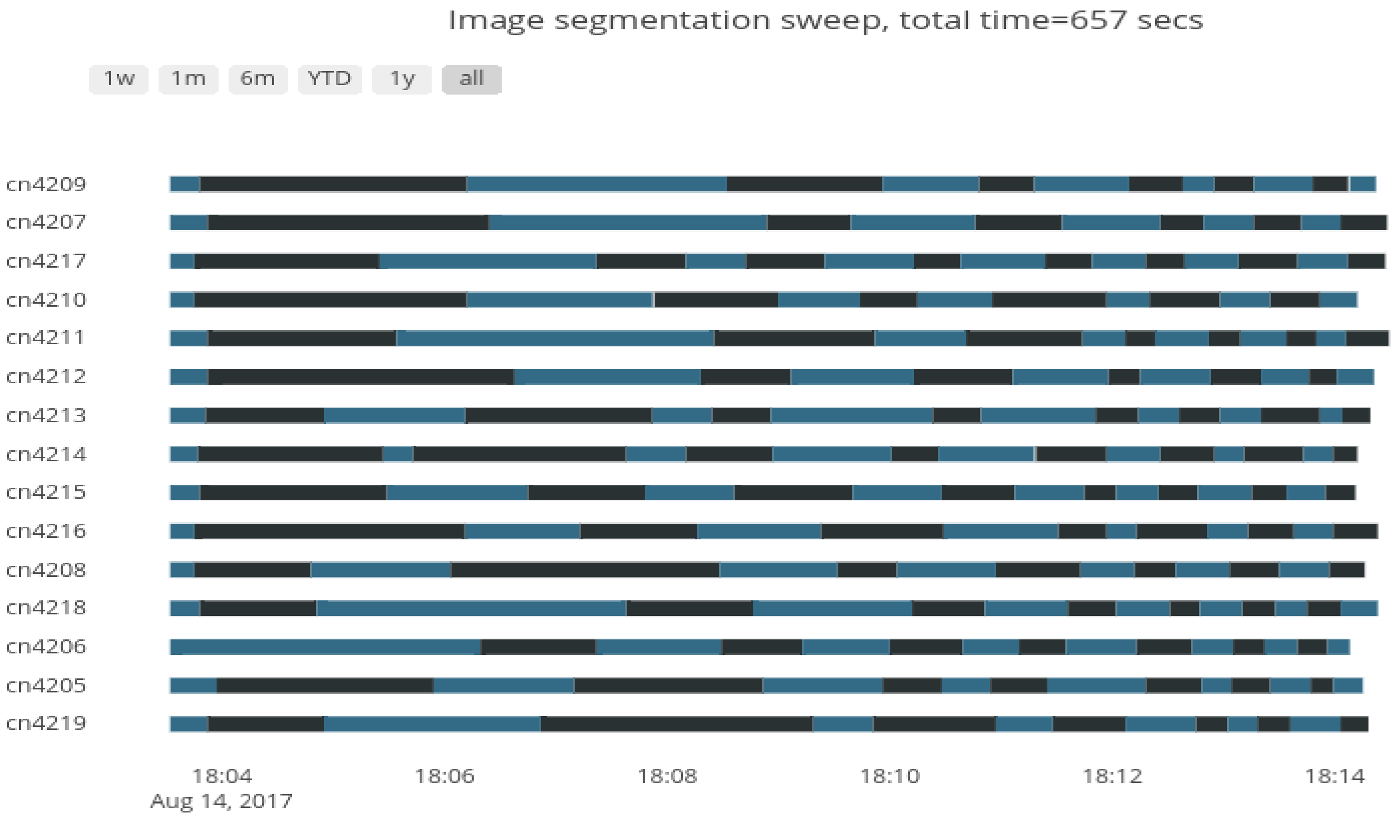
The need for ADLB

Runtime Histogram

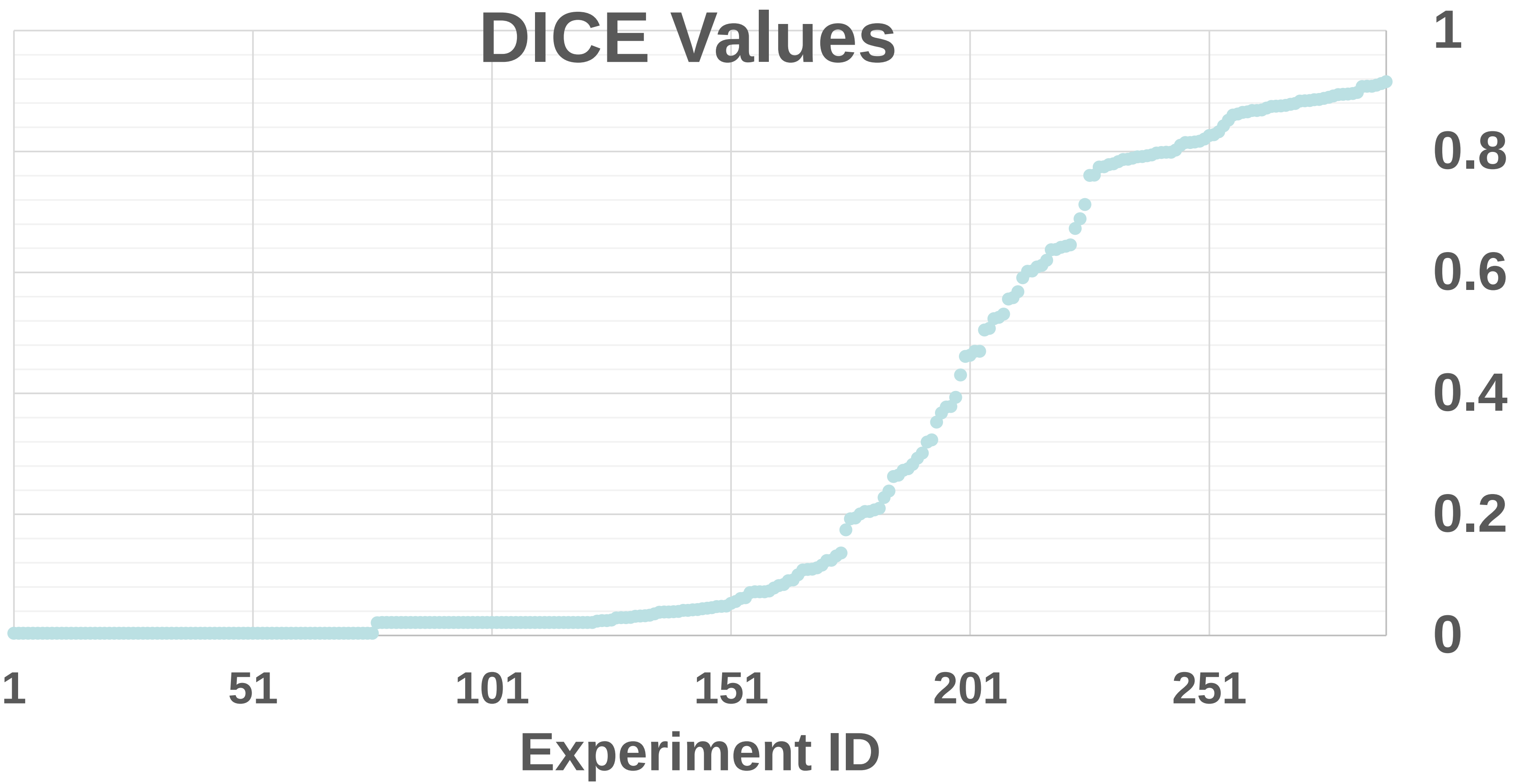


Total runtime = 17.5 hours, experiment runtime = 30 hours.
Can be multiple threads are using the same GPU. Need further investigation.

Smoke test on 14 GPUs.



Hyper parameters sweep



Can we do better than sweeping all parameters?

Agenda

- Introduction to containers
- Example of hyper-parameter optimization on HPC clusters
- **Using CANDLE on Biowulf**

Hyper-parameter optimization using mlrMBO

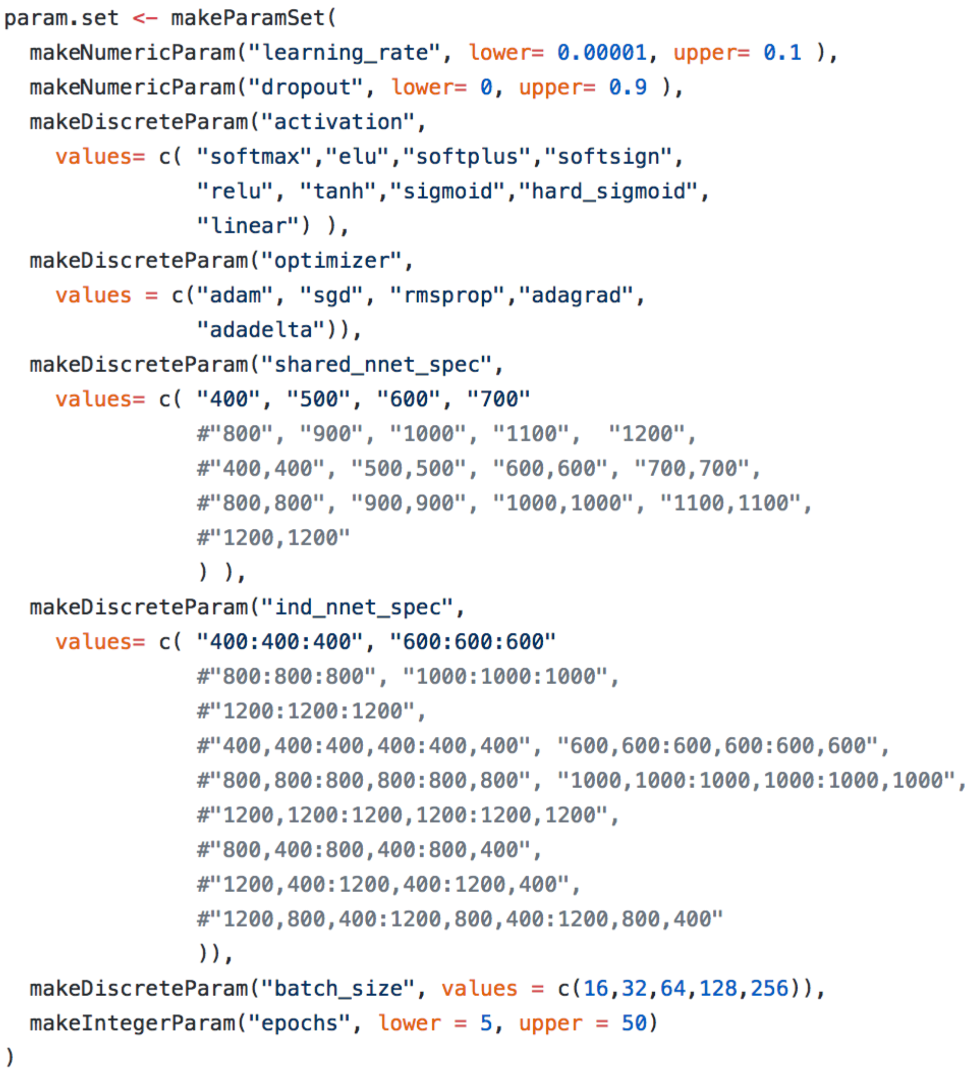
- Objective: Optimize a black box function using Bayesian optimization

  P(score/configuration) using a History of (score,configuration) pairs.

- Arguments:
  - Budget (number of evaluations)
  - Initial design, to create the history.
  - Number of iterations

- Swift/T workflow is integrated with the mlrMBO R package.
- Users have to define the parameters space and the simulation arguments. mlrMBO will suggest multiple points in every iterations

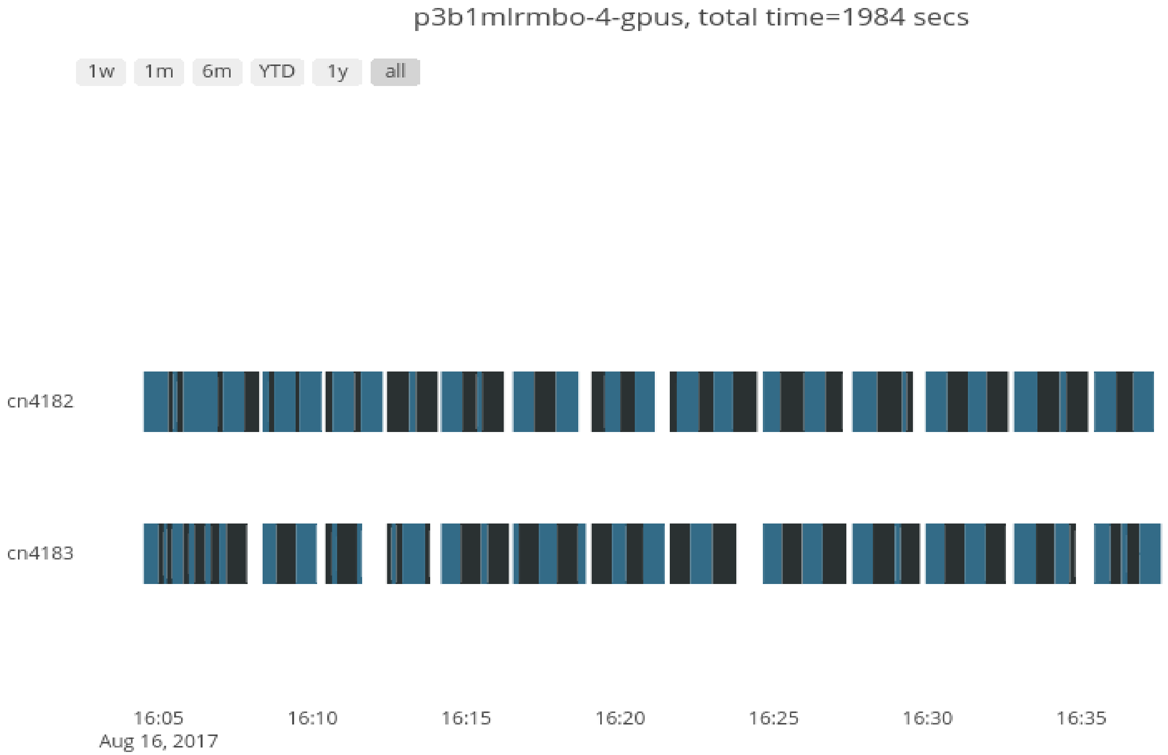
Parameter space example



```
param.set <- makeParamSet(
  makeNumericParam("learning_rate", lower= 0.00001, upper= 0.1 ),
  makeNumericParam("dropout", lower= 0, upper= 0.9 ),
  makeDiscreteParam("activation",
    values= c( "softmax","elu","softplus","softsign",
               "relu", "tanh","sigmoid","hard_sigmoid",
               "linear") ),
  makeDiscreteParam("optimizer",
    values = c("adam", "sgd", "rmsprop","adagrad",
               "adadelta")),
  makeDiscreteParam("shared_nnet_spec",
    values= c( "400", "500", "600", "700"
               #"800", "900", "1000", "1100",  "1200",
               #"400,400", "500,500", "600,600", "700,700",
               #"800,800", "900,900", "1000,1000", "1100,1100",
               #"1200,1200"
               ) ),
  makeDiscreteParam("ind_nnet_spec",
    values= c( "400:400:400", "600:600:600"
               #"800:800:800", "1000:1000:1000",
               #"1200:1200:1200",
               #"400,400:400,400:400,400", "600,600:600,600:600,600",
               #"800,800:800,800:800,800", "1000,1000:1000,1000:1000,1000",
               #"1200,1200:1200,1200:1200,1200",
               #"800,400:800,400:800,400",
               #"1200,400:1200,400:1200,400",
               #"1200,800,400:1200,800,400:1200,800,400"
               )),
  makeDiscreteParam("batch_size", values = c(16,32,64,128,256)),
  makeIntegerParam("epochs", lower = 5, upper = 50)
)
```

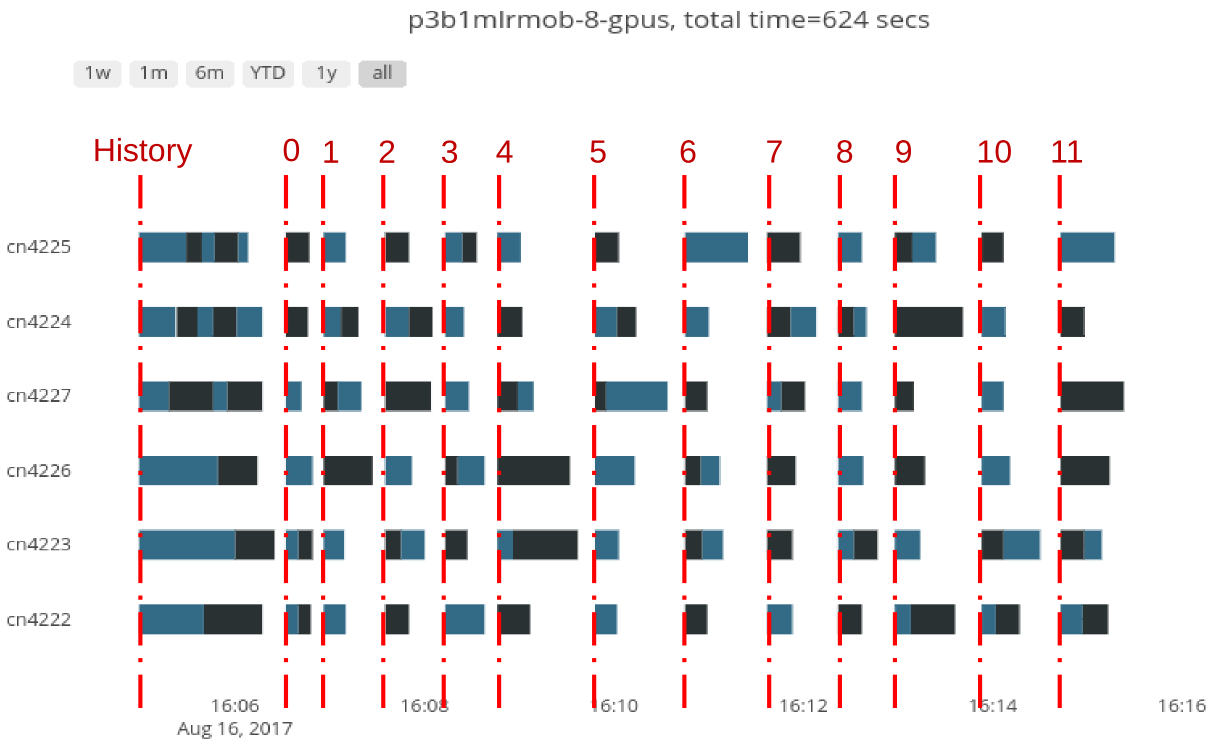
Biowulf run: Budget = 110, initial design = 20

2 working GPUs



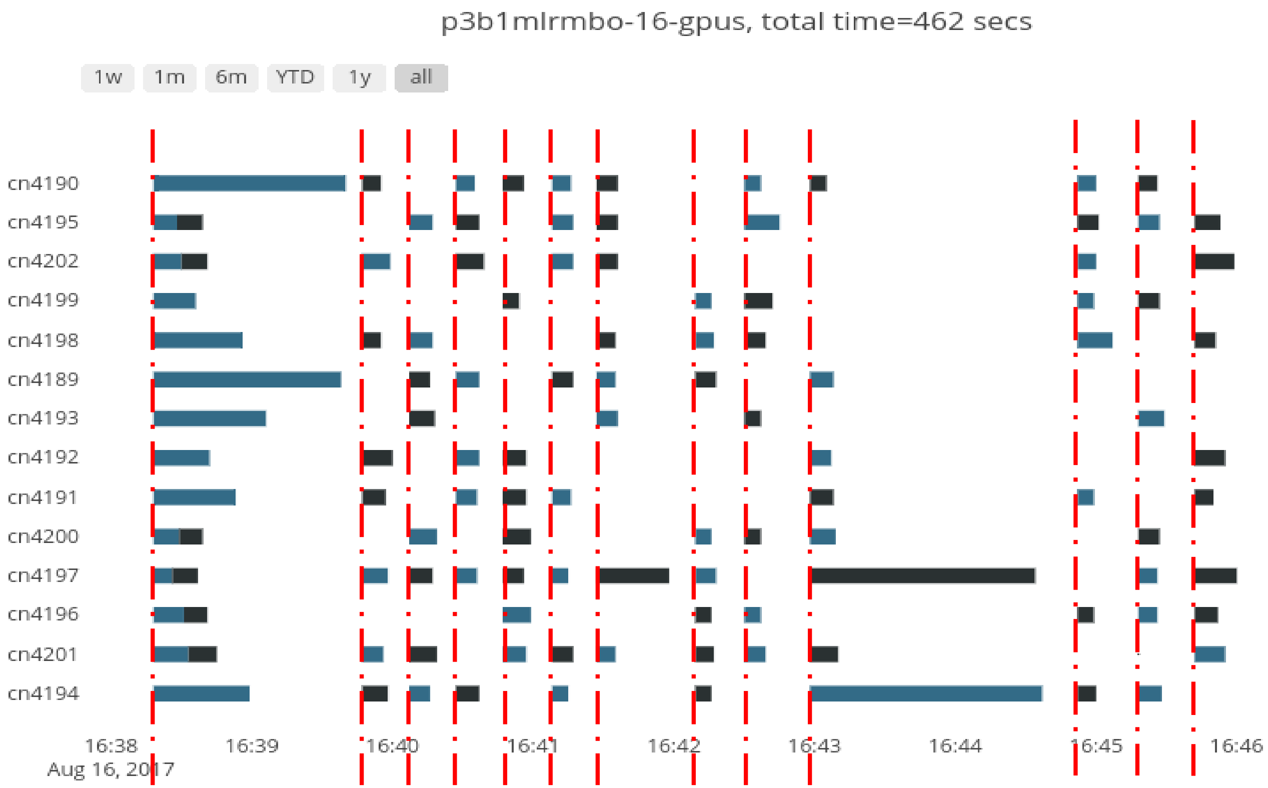
Biowulf run: Budget = 110, initial design = 20

6 working GPUs



Biowulf run: Budget = 110, initial design = 20

14 working GPUs



Can I use mlrMBO with Swift/T today?

- Yes!
  - A Swift/T project template exits on Biowulf under **/data/classes/candle/mlrmob-template**
  - Follow the readme files
- Arguments:
  - Define the function
  - Define the Arguments
  - Set the Budget (number of iterations)
  - Allocate resources (up to 32 K80 GPUs on Biowulf)
- Dependencies:
  - Singularity to run the container. Already installed on Biowulf!

Quick Demo

- singularity shell <singularity-image>
- singularity exec <singularity-image> <your-command>

- Running CANDLE using singularity on Biowulf.

CANDLE status on Biowulf

- Hyper parameter optimization is integrated with mlrMBO and hyperopt packages.
- Configuration of the GPU ID per training instance is available. Allocate 4 GPUs per node and let every GPU train a different model.
- Data parallelism is supported on single node by assigning every MPI job on a node.
- Next release should support data parallelism on multiple nodes.

Conclusions / Take Aways

- The blue print (definition file) for the singularity image is available on GitHub under **ECP-CANDLE/Distribution**
- The art of hyper-parameter optimization can be automated using Bayesian models.
- A template is available on **Biowulf** to make use of mlrMBO optimization package with Swift/T load balancer.
- Data parallelism can be controlled on 1 node in Biowulf today!