running_ER_on_cluster

Vignette from the ER repository

https://github.com/Hanxi-002/EssReg/blob/main/EssRegVignette_pipeline.pdf

Install development version

```
library(devtools)

# Note: if you submit an array job that calls this function in quick succession,
# you'll get rate limited and error out.

# run this to install
devtools::install_github(repo = "TranscriptionFactory/JishnuLabTools", force = F,
                         dependencies = T,
                         auth_token = "github_pat_11ACCQ6NA0Y200JwWDcXeW_xdHZ5ZomD3HWE8g2mRfoFc

library(JishnuLabTools)
```

Data format should be saved as a csv or rds file as one of these:

- Separate X and Y
- Combined X and Y where Y is the first column

Get example yaml files

You can edit these files to have the paths to your X and Y data by using the list accessors (e.g. regression$x_path = 'path to x') or you can just save the yaml and edit it.

Note the output path should always end in "/"

Example regression yaml

```
regression = JishnuLabTools::regression_params
knitr::kable(data.frame(regression_parameters = unlist(regression)))
```

Example classification yaml

```
classification = JishnuLabTools::classification_params

# this is just for printing here
classification$y_levels = "[0, 1]"
knitr::kable(data.frame(classification_parameters = unlist(classification)))
```

	classification_parameters
x_path	
y_path	
out_path	/
k	5
y_factor	TRUE
y_levels	[0, 1]
eval_type	auc
rep_cv	20
nreps	20
alpha_level	0.05
thresh_fdr	0.2
permute	TRUE
std_cv	FALSE
std_y	FALSE
benchmark	FALSE
delta	0.1
lambda	1
lasso	TRUE
plsr	TRUE
pcr	TRUE

If you want to run a coarse grid search over specific (as opposed to the predfined) deltas and lambdas, the yaml file

should have a list of values for delta and lambda, like this:

```
classification = JishnuLabTools::classification_params

# this is just for printing here
classification$y_levels = "[0, 1]"
classification$delta = "[0.1, 0.05, 0.01]"
classification$lambda = "[1.0, 0.1]"
knitr::kable(data.frame(classification_parameters = unlist(classification)))
```

	classification_parameters
x_path	
y_path	
out_path	/
k	5
y_factor	TRUE
y_levels	[0, 1]
eval_type	auc
rep_cv	20
nreps	20
alpha_level	0.05
thresh_fdr	0.2
permute	TRUE
std_cv	FALSE
std_y	FALSE
benchmark	FALSE
delta	[0.1, 0.05, 0.01]
lambda	[1.0, 0.1]
lasso	TRUE
plsr	TRUE
pcr	TRUE

Save the proper yaml file somewhere

```
classification$x_path = 'x.csv'
classification$y_path = 'y.csv'
classification$out_path = '/'

yaml::write_yaml(classification, 'where_you_want_to_save_yaml_file')
```

Slurm script for single submission (put your email into –mail-user=)

```
#!/bin/bash
#SBATCH -t 3-00:00
#SBATCH --job-name= ER
#SBATCH --mail-user=
#SBATCH --mail-type=FAIL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --mem=150g
#SBATCH --cpus-per-task=16

module load gcc/10.2.0
module load r/4.2.0

Rscript runER.R --yaml_path 'path_to_yaml' --coarse_grid F
```

Save this as runER.R or whatever comes after Rscript above

```
#!/usr/bin/env Rscript
args = commandArgs(trailingOnly=TRUE)

library(devtools)
library(doParallel)
library(foreach)
library(tidyverse)

# if need to install
devtools::install_github(repo = "TranscriptionFactory/JishnuLabTools@master", force = F,
                         dependencies = T,
                         auth_token = "github_pat_11ACCQ6NA0Y200JwWDcXeW_xdHZ5ZomD3HWE8g2mRfoFc

library(JishnuLabTools)

cores <-  as.numeric(Sys.getenv("SLURM_CPUS_PER_TASK", unset=NA))
if(is.na(cores)) cores <- detectCores()
# if(!is.na(cores) & cores > 1) cores <- cores
registerDoParallel(cores)
cat('number of cores using', cores, '. . .\n')


# process arguments from command line
command_args = JishnuLabTools:::arg_loader(args, JishnuLabTools::runER_args)

yaml_path = command_args$yaml_path
coarseGrid = command_args$coarse_grid

# call ER function
JishnuLabTools::runER(yaml_path, coarseGrid)
```

Slurm batch submission

You want to point to a folder with yaml files or dataframes (combined X/Y) Note, you should comment out the `install_github()` in the runER.R file so that you don't get rate limited

```
#!/bin/bash
#SBATCH -t 3-00:00
#SBATCH --array= numbers
#SBATCH --job-name= ER
#SBATCH --mail-user=aar126@pitt.edu
#SBATCH --mail-type=FAIL
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --mem=150g
#SBATCH --cpus-per-task=16


echo "SLURM_JOBID: " $SLURM_JOBID
echo "SLURM_ARRAY_TASK_ID: " $SLURM_ARRAY_TASK_ID
echo "SLURM_ARRAY_JOB_ID:  " $SLURM_ARRAY_JOB_ID

cd 'path to where you have yaml files'

arrayfile=`ls | awk -v line=$SLURM_ARRAY_TASK_ID '{if (NR == line) print $0}'`

module load gcc/10.2.0
module load r/4.2.0
echo $arrayfile
# usage: Rscript -d datapath_from_working_dir_including_extension
Rscript runER.R --yaml_path $arrayfile --coarse_grid F
```