IMPLEMENTED METHOD

CODE AVAILABLE ON GITHUB

https://github.com/theMIDAgroup/conditional_neurofgm/tree/main

Implemented method

Input: Projection scores for all observed functions on the fixed basis.
Grouping factor
Number of penalty factor λ_j^F for the penalized regression to try
Set of neighboor recognition threshold t_j (define significant regression coefficent) to try. *NOTE:* $\epsilon_j = t_j * \lambda_j^F$
The number of cross validation folds.
Some parameters needed for the computation of the regression coefficient
ESTIMATION:
for $j \in V$ **do**
for $\lambda_j^F \in expseq([0, \lambda_j^{max}], L)$ **do**
Solve a multi-outcome regression for the scores of each function on all the other function scores and the interaction of the other function scores with the covariates of interest with a grouped vector on vector lasso regression.

$$\widehat{\mathbf{B}} \in \arg\min_{\mathbf{B}} \left\{ \frac{1}{2n} \left\| A^Y - A^X \mathbf{B}^T \right\|_F^2 + \lambda_j^F \sum_{l=0}^{2(p-1)} \|\mathbf{B_l}\|_F \right\}$$

for $\epsilon \in \epsilon_j = t_j * \lambda_j^F$ **do**
Compute

$$\widehat{\mathcal{N}}_j = \{k \in VxV : \|\widehat{B}_k\|_F > \epsilon\}$$

for $k \in [K]$ **do**
Re-estimate $\tilde{\boldsymbol{B}}_{\boldsymbol{l}}$ for $l \in 2[p-1]$ by solving

$$\tilde{\mathbf{B}} \in \arg\min_{\tilde{B}} \left\{ \frac{1}{2n} \left\| A^Y - A^X_{eff} \mathbf{B}^T \right\|_F^2 \right\}$$

with the k-th-fold training set, *NOTE:* A^X_{eff} contains only the score of the functions in $\widehat{\mathcal{N}}_j$;
Evaluate the estimate on the k-th-fold test data computing the resisuals of the model and using the SCV-RSS criterion

$$\sum_{i \in I_{\text{test}}} \|\hat{\mathbf{re}}_i\|_2^2 + \log(|I_{\text{test}}|) \cdot |\hat{N}_j(\lambda_n, \epsilon_n)|$$

end for
Calculate the mean of the criterion across all K folds;
end for
end for
Pick the (λ_n, ϵ_n) pair that minimizes the mean criterion;
Estimate $\hat{N}_j$ as above using the optimal hyperparameters
end for
Combine all neighborhoods into the estimated edge set using AND/OR rule
Output: Return $\hat{E}$

Algorithm 1: Functional neighborhood selection

Implemented method

Parallelize on HPC lunching on job per node:
Run Sbatch_parallel_luncher.sh from the terminal using: `alessia.mapelli@hnode02$ sh Sbatch_parallel.sh`

```
#!/bin/bash

JOBS_LIMIT=70
r_script=/group/diangelantonio/users/alessia_mapelli/conditional_neurofgm/Simulation_studies/Sim_1_sbatch_parallel.R


for i in $(seq 1 64); do
  while [ "$(squeue -u $USER |wc -l)" -ge "${JOBS_LIMIT}" ]; do
    echo "Jobs limit reached, I sleep for a while";
    sleep 240
  done
  echo "Processing: Node ${i}"
  input=${i}
  RES=$(sbatch --parsable "Sbatch_parallel.sbatch" "${input}" "${r_script}" )
  echo "running job id: ${RES}"
done
```

This script runs Sbatch_parallel.sbatch giving it:
input: the node on which we want to compute the neighborhood
r_script: the path to the R script that performs the neighborhood computation

```
#!/bin/bash
#SBATCH --mail-type=ALL
#SBATCH --mail-user=alessia.mapelli@fht.org
#SBATCH --output=/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/logs_parall/%j_Node.log
#SBATCH --partition=cpuq
#SBATCH --mem=500MB
#SBATCH --nodes=1
#SBATCH --ntasks=16
#SBATCH --job-name=Brain_sim
#SBATCH --time=00:20:00

source /center/healthds/singularity_functions
cd $HOME

input=$1
r_script=$2

Rscript "${r_script}" "${input}"
```

This script lunch a job to run the Sim_1_sbatch_parallel.R giving it the node on which to compute the neighborhood
-- output is used to define the location of the log files that will contain all the messages coming from running the R script

Implemented method

```
################################################
## 0. USER DEFINED PARAMETERS (MODIFY THIS PART)
################################################
score_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/freq_scores_reord.csv"
grouping_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/grouping_factor.csv"
output_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/results/1504_run/"
name_output = "rand_hyper_search"
n_basis = 8
L = 100
K = 5
thres.ctrl = c(0, 0.2, 0.4, 0.8, 1.2, 1.6, 2.0)
tol.abs =1e-4
tol.rel = 1e-4
eps = 1e-08
verbose = FALSE
p.rand.lam = 0.5
p.rand.thr = 1
################################################
```

This part of the R scripts define the parameters that will be used to compute the neighboorhood:
score_path: path where the score of each patient are stored. It must be a dataframe with a row per patient and the concatenated scores of each nodes as column.

	Score1 of node 1	Score 2 of node 1	....	Score M of node 1	Score1 of node 2	Score 2 of node 2	.....	Score M of node p
Pat 1								
Pat 2								

Implemented method

```
################################################
## 0. USER DEFINED PARAMETERS (MODIFY THIS PART)
################################################
score_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/freq_scores_reord.csv"
grouping_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/grouping_factor.csv"
output_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/results/1504_run/"
name_output = "rand_hyper_search"
n_basis = 8
L = 100
K = 5
thres.ctrl = c(0, 0.2, 0.4, 0.8, 1.2, 1.6, 2.0)
tol.abs =1e-4
tol.rel = 1e-4
eps = 1e-08
verbose = FALSE
p.rand.lam = 0.5
p.rand.thr = 1
################################################
```

This part of the R scripts define the parameters that will be used to compute the neighboorhood:
grouping_path: path where the grouping variable is available for each patient

	group
Pat 1	
Pat 2	

Implemented method

```
################################################
## 0. USER DEFINED PARAMETERS (MODIFY THIS PART)
################################################
score_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/freq_scores_reord.csv"
grouping_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/grouping_factor.csv"
output_path = "/group/diangelantonio/users/alessia_mapelli/Brain_simulations/Sim_1/results/1504_run/"
name_output = "rand_hyper_search"
n_basis = 8
L = 100
K = 5
thres.ctrl = c(0, 0.2, 0.4, 0.8, 1.2, 1.6, 2.0)
tol.abs =1e-4
tol.rel = 1e-4
eps = 1e-08
verbose = FALSE
p.rand.lam = 0.5
p.rand.thr = 1
################################################
```

This part of the R scripts define the parameters that will be used to compute the neighboorhood:

ouput_path & name_output: the neighborhood linked to the best hyperparameters are going to be saved as an rda in the ouput_path with name name_output_node.rda

n_basis: number of bases used for the dimensionality reduction (number of score per node)

L: number of λ to try

K: number of folds to use in CV

thres.ctrl: Set of neighbr recognition threshold tj (define significant regression coeffici cent) to try. NOTE: $\epsilon = t * \lambda$

p.rand.lam: proportion of λ to try in a random search (1 means full search)

p.rand.thr: proportion of t to try in a random search (1 means full search)