**Meeting agenda.**

1. Reviewed progress last week.
2. Dr. Jiang gave feedback to Garrett’s work.
3. Dr. Jiang answered Garrett’s questions.

**Progress made in the past week.**

**Issues/Questions and Comments**

Dr. Jiang’s feedback to Garrett’s work per her email reply.

Hello,

I am glad you made progress. See some of my comments below in yellow highlight. I’d like to review your progress in detail and give my feedback following your report tomorrow morning during our ZOOM meeting.

# MBIL

MBIL (Markov Blanket and Interactive risk factor learner) is an algorithm that utilizes Bayesian networks and information theory to determine direct and interactive risk factor for metastatic breast cancer (MBC). (See papers for details and citations.)

## Install

MBIL can be installed from PyPI:

`pip install mbil-py`

Has the installation instruction been tested? Does it truly work?

### For all examples shown:

This is the data:

|B|         C|           D|           F|            E|

|-|--|---|---|---|

|2|         1|           0|           1|           0|

|3|         1|           1|           2|           1|

|3|         0|           1|           1|           0|

|3|         0|           1|           1|           1|

|2|         1|           2|           2|           0|

These are the set variables:  No these are not variable, rather, they are parameters. Please consider add explanation of each of the parameters and value ranges.

`alpha = 4`

`target = "E"`

`top = 20`

`max\_single\_predictors = 20`

`max\_interaction\_predictors = 20`

`max\_size\_interaction = 3`

`threshold = 0.05`

`maximum\_number\_of\_parents=7`

These are the basic score and search objects:

`score\_test\_obj = mbilscore.mbilscore(dataset\_df=dataset\_df, target=target, alpha=alpha)`

`search\_test\_object = mbilsearch.mbilsearch(threshold=threshold,

                                           max\_single\_predictors= max\_single\_predictors,

                                           max\_interaction\_predictors=max\_interaction\_predictors,

                                           max\_size\_interaction= max\_size\_interaction,

                                           dataset\_df = dataset\_df,

                                           alpha = alpha,

                                           target = target)`

## MBILScore functions and their uses:

### mbilscore.calculate\_score

Parameters: an mbilscore object and subset\_size

Return value: a hashmap storing the results

Calculates the BDeu score for all possible subsets (maybe P(data | DAG))

The BDeu score is a score that measures the probability of the data given the directed acyclic graph using a parameter alpha to represent prior equivalent sample size.

Example:

`scores = score\_test\_obj.calculate\_score(top = top, subset\_size = 2)`.   Is top supposed to be an mbilscore object based on your description rather than a number (20)?

`print(scores)`

Output:

`[("['B', 'C']", -3.753417975251508), ("['B', 'F']", -4.158883083359674), ("['B', 'D']", -4.382026634673884), ("['C', 'D']", -4.382026634673884), ("['D', 'F']", -4.382026634673884), ("['C', 'F']", -4.85203026391962)]`

### mbilscore.calculate\_information\_gain

Parameters: an mbilscore object and subset\_size

Return value: a hashmap storing the results

Calculates the information gain for all possible subsets

Information gain is the expected reduction in entropy of a variable conditional on a seperate variable

Example:

`ig\_scores = score\_test\_obj.calculate\_information\_gain(top = top, subset\_size = 2)`.  )`.   Is top supposed to be an mbilscore object based on your description rather than a number (20)?

`print(ig\_scores)`

Output:

`[("['D', 'F']", 0.5709505944546686), ("['B', 'C']", 0.5709505944546684), ("['B', 'F']", 0.5709505944546684), ("['C', 'D']", 0.5709505944546684), ("['B', 'D']", 0.4199730940219749), ("['C', 'F']", 0.17095059445466854)]`

## MBILSearch functions and their uses:

### mbilsearch.get\_single\_predictors\_score

Parameters: an mbilsearch object

Return value: a list of all direct risk predictors and their corresponding Bayesian score as a float

Calculates the Bayesian score for every direct risk predictor

Example:

When creating an mbilsearch object the function is called when initializing the variable `single\_list\_score`

Thus the variable can be printed as so

`print(search\_test\_object.single\_list\_score)`

Output:

`[('B', -3.5835189384561104)]`.     Shouldn’t it be a list of single predictors? In your example, it should include B, C, D, and F each separatedly?

### mbilsearch.get\_interaction\_predictors\_score

Parameters: an mbilsearch object

Return value: a list of the top interactive risk predictors and their corresponding score as a float

Calculates the score between all interactions of predictors

Example:

When creating an mbilsearch object the function is called when initializing the variable `interaction\_list\_score`

Thus the variable can be printed as so

`print(mbilsearch.interaction\_list\_score)`

Output:

`[("['B', 'C']", -3.753417975251508), ("['B', 'F']", -4.158883083359674), ("['C', 'D']", -4.382026634673884), ("['D', 'F']", -4.382026634673884), ("['C', 'F']", -4.85203026391962)]`

**Questions from Garrett and Dr. Jiang Answers**

**Ongoing tasks that cover more than a week**

**Specific tasks for the coming week**

1. Write a readme file for the python version of MBIL.
2. Explanation in details in your own word of the MBIL algorithm.
3. For each function of the MBIL-py provides, write an example to help users to understand the function and how they can use the function. Using figures is encouraged.
4. Learn from other existing readme files (such as the one provided by Shap https://github.com/slundberg/shap#readme) as to how to write a good readme file.
5. Other than the ideas that I give in (1) and (2), be creative. …….
6. Deadline: 2022.12.13

**Less urgent tasks**