Computational analysis of medical genomic data

CS workshop, Fall 2021

0368-3553

Project goals and guidelines

Project goals

In this project we will use gene expression data to predict how Acute Myeloid Leukemia (AML) patients respond to drugs. You will use two datasets for this purpose: BEAT AML and TCGA. BEAT AML includes 198 samples, and contains gene expression and drug response data (for 79 drugs) for each sample. TCGA includes the gene expression and mutation profile (for 11 common AML mutations) of 167 samples. We will aim to solve three different computational tasks using these data:

1. Predict tumor response to multiple drugs given gene expression data for BEAT AML samples. In this task you will first train on gene expression data, and response to all drugs will be available. The output of this task will be a predictor, which given gene expression data will predict response to all drugs that were used for training. Evaluation of your predictor will be done on held-out samples.
2. Predict tumor response to multiple drugs given gene expression data. Unlike task 1, in this task during training you can also use the gene expression profiles of TCGA samples (but not their mutation data). The learned algorithm will be used to make predictions on held-out BEAT AML samples.
3. The goal in this task is to discover mutation – drug interactions. You are given gene expression and drug response data for BEAT AML samples, and gene expression and mutation data for TCGA samples. The output should be a mutation x drug Spearman correlation matrix (see further description below). We will also supply you with a ground truth mutation x drug correlation matrix, calculated for a subset of the drugs. This matrix only serves to guide you during algorithm development, and should not be used in a training framework.

You are free (and are encouraged!) to suggest and solve additional goals, or additional variants to the existing goals.

Project data

The data are available here:

https://drive.google.com/file/d/1GlVJA3NyGOw3x7Llss6LjnxEUV7wi-DZ/view?usp=sharing

- The beat_rna and beat_drug files contain the RNA-seq and drug response data for BEAT. In both files, rows represent features (genes and drugs respectively), and columns represent patients.
- The tcga_rna and tcga_mut files contain the RNA-seq and mutation data for TCGA. Here also rows are features and columns are patients.
- The drug_mut_cor file contains the mutation x drug response correlation. The exact calculation that created this matrix is explained in more detail below.

- Finally, drug_mut_cor_labels contains the BEAT samples for which mutation data was available and were used to calculate drug_mut_cor (again, see details below).

You are given two datasets on AML patients: BEAT AML and TCGA. BEAT AML contains drug response data (IC50 values) for 79 drugs. Note that some of these IC50 values are missing, and it is up to you to decide how to handle them. Additionally, BEAT AML contains gene expression data (measured using RNA sequencing). TCGA contains gene expression data (also measured using RNA sequencing) and mutation data (binary values).

We advise you to transform the gene expression data as follows: x -> log2(1+x), and to transform the IC50 values: x -> log10(x) (but this is not mandatory if this does not improve performance).

To assess the performance of the methods, use the sum of squared error across all drugs (the square of the Frobenius norm) in all three tasks. We will use the sum squared errors of the log10 IC50 values, rather than the IC50 values themselves.

In task 3, the ground truth correlation matrix was calculated as follows: for a specific drug and mutation, only the samples with a known IC50 were considered. Denote the number of samples with IC50 values by n. The IC50 values for these patients were ranked – the lowest value was replaced with 1, the second with 2, and so on until the highest was replaced with n. In case of ties, the rank of all equal values will be the same, and will be equal to their average ranking. For example, the vector: c(1, 2, 2, 2, 3, 3) is replaced with c(1, 3, 3, 3, 5.5, 5.5).
This vector of ranks was correlated with the binary mutation vector using standard Pearson correlation.
Note that we used mutation data for the BEAT samples that you do not observe. Only 174 of the 198 BEAT samples have mutation data, so only these samples were used (and for drugs in which they have missing IC50 values, some of these 174 samples weren't used as well). You are given the list of 174 BEAT samples with mutation data that we used (in file drug_mut_cor_labels), although using the gene expression data of all 198 BEAT samples in task 3 may yield better results than using the 174 samples.

Milestones

Milestone 1: 16.11.21

Each group will meet with us (via zoom) and will present their suggested approaches for the tasks. You are expected to think creatively and present several approaches per task, as it is likely that some of your approaches will not produce adequate results.

Submission:

- A ~20 minute presentation and discussion via zoom.

Grade: 10 points.

Milestone 2: 7.12.21

Each group will submit initial results for the three computational tasks, including the code used to produce these results, and will present it to us (in zoom). At this milestone you will not be graded based on your results, but based on your efforts and depth. Therefore, do not present only your best results, but show different approaches that you tried. You are also required to submit a report that

must include the squared error for all tasks. The squared errors in tasks 1 and 2 should be calculated using 5-fold cross validation, using 5 folds that will be provided by us. In task 3 the squared error doesn't need to be calculated using cross validation.

(Optional, to be determined later) We will create a scoreboard showing the results of each milestone. The scoreboard will be anonymous and its goal is to give the groups a feeling of "where they stand". It will not be part of the grade.

Submission:

- A document describing the methods and results, including supporting evidence. Size limit: 7 pages
- Code for producing the analysis. This code will be reviewed by us for organization and documentation but will not be executed.
- A ~20 minute presentation via zoom.

Grade: 15 points.

Milestone 3: 4.1.22

Same guidelines as for Milestone 2. The submitted document and code should be updates of the previous versions, highlighting new approaches and improvements and the main differences and updates.

Grade: 15 points.

Final submission: 16.2.22

Submissions are due 16.2.22. As before, the submission should include a document describing the analysis and results, plus the code. The document is limited to 10 pages. It should contain the most prominent results and conclusions. You can add an appendix of additional results.

In this submission, we will execute your algorithms for tasks 1 and 2 on new data that were not previously given to you, and the performance of the methods will be part of your grade. For task 3, we will compare your mutation – drug interaction matrix to a held out matrix.

Final joint meeting of all groups: 18.2.22

In the joint meeting, each group will have ~15 minutes for a final presentation of their project.

Grade:

- 30 points for the submission.
- Up to 25 points for the performance of your methods on data that you were not previously given. The grade for the performance will be calculated as follows:
  27.5 – group_rank * 2.5
  where group_rank = (group_rank1 + group_rank2 + group_rank3) / 3
  and group_rankX is the group's ranking in task X.
  The scores in tasks 1 and 2 will be calculated as follows:
  group_rankX = 0.8 * group_performance_rankX + 0.2 * group_runtime_rankX
  where group_performance_rank is the group's ranking based on the squared error (where 1

is the best group, 2 is second, 3 third etc. Ties will be scored in accordance to the House of Hillel) and group_runtime_rank is the rank based on runtime (where 1 is the fastest executing method).
The scores in task 3 will equal the group_performance_rank.

- 10 points for the final presentation.
- Up to 10 bonus points can be awarded for suggesting and solving additional project goals.

Code guidelines

Code should be written in python or R. If you want to use other languages – you need to get our permission.

All code (for all milestones) should be submitted on nova. Make sure to change permissions for that directory such that other users can run it. The code should be clear and documented, and tested to run smoothly on nova.

The submission must include a command line script for performing prediction with your preferred method in tasks 1 and 2, with arguments for the paths of the input files and the output paths, and for the mode of execution (task 1 or 2). A readme file should explain how to execute this script. For task 3, you should submit the mutation-drug correlation matrix.

In milestones 2 and 3, we will not execute the code. In the final submission, we will use the code on nova for prediction on data that you were not given, and the prediction of your method will be used as part of your grade. You should make sure that your code is functional and running on nova, using the training data provided for the workshop. We strongly advise you to make sure your code runs on nova throughout the workshop. Do not wait till the final submission as it may not be operational.

More detailed instructions on the scripts' input and output will be given later in the course.