# Mining Association Rules from Gene Expression Data

Tianle Ma, 50096690, [tianlema@buffalo.edu](mailto:tianlema@buffalo.edu)

Other team members

1. Implementation of Apriori Algorithm **(30 Points)**
2. Using **your own words** to describe Apriori algorithm. (10 points)
3. Time complexity of Apriori algorithms (compare it with the brutal force algorithms) (10 points)
4. How do you implement this algorithms in your code? What data structures do you use? Any specific considerations? (10 points)
5. Results **(60 Points)**
6. For part 1, you should list the result for different support values like this: (20 points)

Support >= 50%

Size of frequent itemset = 1: 109

Size of frequent itemset = 2: ???

Size of frequent itemset = 3: ???

Total : 174

1. For part 2, if we choose support >= 50%, confidence >= 70%, you should generate (how many) rules. You code should be able to list all the rules. In your report, just show the number of rules generated.

In your code, there should be variables to change support and confidence, so that you can get any result you want. (20 points)

Your program should be able answer the template queries. (20 points)

Your program should contain some sample queries. The following is what I wrote using python to give you a sense of what it should be like. You do not need to strictly follow this.

For template 1, we have 9 possible keywords combinations:

(result11, cnt) = asso\_rule.template1("RULE", "ANY", ['G1\_UP'])

(result12, cnt) = asso\_rule.template1("RULE", "NONE", ['G1\_UP'])

(result13, cnt) = asso\_rule.template1("RULE", 1, ['G1\_UP', 'G10\_Down'])

(result14, cnt) = asso\_rule.template1("BODY", "ANY", ['G1\_UP'])

(result15, cnt) = asso\_rule.template1("BODY", "NONE", ['G1\_UP'])

(result16, cnt) = asso\_rule.template1("BODY", 1, ['G1\_UP', 'G10\_Down'])

(result17, cnt) = asso\_rule.template1("HEAD", "ANY", ['G1\_UP'])

(result18, cnt) = asso\_rule.template1("HEAD", "NONE", ['G1\_UP'])

(result19, cnt) = asso\_rule.template1("HEAD", 1, ['G1\_UP', 'G10\_Down'])

For template 2, we have 3 keywords choices:

(result21, cnt) = asso\_rule.template2("RULE", 2, ['G1\_UP'])

(result22, cnt) = asso\_rule.template2("BODY", 2, ['G1\_UP'])

(result23, cnt) = asso\_rule.template2("HEAD", 2, ['G1\_UP'])

For template 3, I implement one AND/OR logical operator connecting two parts which can be from either template 1 or template 2. For example, “1or1” means “Template1 OR Template1”.

(result31, cnt) = asso\_rule.template3("1or1", "BODY", "ANY", ['G1\_UP'], "HEAD", 1, ['G59\_UP'])

(result32, cnt) = asso\_rule.template3("1and1", "BODY", "ANY", ['G1\_UP'], "HEAD", 1, ['G59\_UP'])

(result33, cnt) = asso\_rule.template3("1or2", "BODY", "ANY", ['G1\_UP'], "HEAD", 2, ['G59\_UP'])

(result34, cnt) = asso\_rule.template3("1and2", "BODY", "ANY", ['G1\_UP'], "HEAD", 2, ['G59\_UP'])

(result35, cnt) = asso\_rule.template3("2or2", "BODY", 1, ['G1\_UP'], "HEAD", 2, ['G59\_UP'])

(result36, cnt) = asso\_rule.template3("2and2", "BODY", 1, ['G1\_UP'], "HEAD", 2, ['G59\_UP'])

When I run your program, it should be able to output the sample query results. (Preset the Support >= 50%, Confidence >= 70%). List the numbers of rules returned for each query in your report.

Besides, **10 points** for you **code quality**: easy to read, good documentation, efficient design, etc. You code must be able to generate the right results and consistent with your report.

1. Other Considerations

If you have any other considerations or improvements, state it here. We may or may not consider to add some points (at most 10 points to boost your score) if you have done extra very nice job. The maximum score for this homework is 100.

Good luck!