ECE 454 – Assignment 3 Design Document

Joshua Kalpin – 201414492
John Zanutto – 20418726

Part 1

For this part we chose to treat this exclusively as a map job. For each sample we iterate through all the genes storing a queue of max values. Once we have gone through all of them, we output the max result at the end. This problem could have also been solved by mapping samples to individual genes and then finding the max in a reducer, but we felt like this was unnecessary.

Part 2

For this part we utilized a counter to count the total number of genes while mapping whether a gene was related to the current sample or not. In the reducer, we then added up all the samples the gene was related to and utilized the count to find the score.

Part 3

For this part we used 2 map and 2 reduce jobs.

First we mapped genes to their samples and expr value and filtered out zeros. In the corresponding reducer we then took those gene, sample pairs and generated sample pairs that took the multiplication of the two corresponding gene expr values (the pre-summing operation of the dot product) leaving us with two samples corresponding to a gene product.

The next mapper essentially does nothing except more filtering and preparing the values for the next reducer. The final reducer takes all of the sample product mapping and sums them together to get the dot product.

Part 4

Part 1

The design is similar to the Hadoop implementation of part 1 where the UDF handles the role of the mapper.

Part 2

For this part we first read in the input and use the group samples ALL command to get the count of all the samples. We then utilize our UDF to determine whether a gene is related to the sample. We then group everything by gene_id and find the score similarly to part 2, except using pig.

Part 3

Unlike the previous part 3 we have our UDF do the entire dot product sum, while pig generates all the pairs. These are then just generated to strings and outputted.