Name: Michael Rogers
ID: 105667404

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 1 **Spring 2019, CU-Boulder**

Your colleagues from the biology department have reached out to you for help. They are trying to complete an analysis for an important study, but the program they have written is taking too long to complete. Here is their email:

> Dear Computer Scientist I know,
>
> We are studying a specific set of genes that we think play some role in chemoresistance. To test how these genes respond to chemo, we tested the activity of all genes (i.e., the expression) before and after treatment in both normal healthy skin tissue and the tumor. We now need to measure how these genes respond to treatment by comparing their expression before and after being exposed to the chemo. Our data includes
>
> - A list of genes we are studying
> - A list of all genes
> - Gene expression before treatment in normal skin tissue (e.g., TP53 5.1, BCRA1 3.1, ...).
> - Gene expression before treatment in the tumor (e.g., TP53 6.2, BCRA1 4.1, ...).
> - Gene expression after treatment in normal skin tissue (e.g., TP53 10.5, BCRA1 2.1, ...).
> - Gene expression after treatment in the tumor (e.g., TP53 20.5, BCRA1 1.1, ...).
>
> For each gene on our list, we want to find the difference in expression from before and after treatment, then take the mean of differences for the tumor and for the normal skin tissues. To compute these values, we take one gene from our list and scan the four expression files until we find the matching gene name. We then use the corresponding four expression values to compute the two differential values. We repeat this for all of the genes we care about, then find the mean of the normal and tumor values. While this seems to work, we would like to test many different gene sets, but our current program is taking too long. Please help.
>
> Thanks in advance,
>
> Your colleague in the bio department

Name: Michael Rogers

ID: 105667404

CSCI 3104, Algorithms **Profs. Grochow & Layer**

Explain-It-Back 1 **Spring 2019, CU-Boulder**

Help your colleagues understand why their method is taking so long, suggest to them a different algorithm and explain why they can expect this new method to run faster.

Name: Michael Rogers
ID: 105667404

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 1 **Spring 2019, CU-Boulder**

Hello Colleague from the Bio department,

While your method may get the job done for one gene, it does way more work than is necessary to get it done, which is causing the delay in processing. Going through all the different lists for each gene means you must look at every single expression in each list and compare it with the gene that you care about until you find the correct gene. In the world of computer science, we would call that a brute force algorithm because it is forcing its way through the entire set of data without using any kind of 'shortcut'. This brute force strategy works for smaller data sets, but because there are so many genes that you are comparing, it begins to seriously affect how efficiently you can get through this data. I have a suggestion for a strategy that should save you time and be much more versatile for different gene sets. To begin, you need to sort the genes so that the program doesn't have to go through the entire list and we can instead take a 'shortcut' to the gene we want to evaluate. We can do this by sorting all the genes by the first letter in the sequence and putting each expression list into alphabetical order. This alone will significantly cut down runtime because now instead of having to go through every gene in the list now we can just search through the genes with the same beginning letter. Now that we have the genes sorted, we can begin to look at how we are going to skip to the specific letter that matches the first letter in the sequence of the gene. This can be accomplished by using a kind of placement test(what we call a hash function in CS). We will run the gene through the placement test that will determine where we need to skip to. Once we skip to the letter that we need, we can do a similar algorithm to what you originally came up with. Now that we have cut down the searching being done significantly, we can go through the list of alphabetically sorted genes until we find match. To store the expression for each gene from each list we are going to add it to a sort of profile that is associated with each gene in the list of genes we care about. As we get the information it will be added to each gene's profile, so all the information is readily available when we are ready to compute the differential values. Once the whole gene set has been run through the lists, we can start computing the differential values of each gene. Now we can just go through the list of genes we care about and compute the values from the information we collected when we ran it though all four of the expression lists. I think you will find that this strategy will work significantly faster that what you are using right now. I would be happy to help you implement this program so that you don't have to get into the specifics of the code.

Good Luck!
A Computer Scientist you know