Problem Set – Mapping and Loops

Starting out, the easiest way to write a lap or to map a function is to first teach yourself how to do the desired operation once. Once you've figured out a series of commands that does the job, you may then be able to cut and paste the commands into the body of the loop or mapping operation. All that remains is to specify how to iterate the appropriate number of times and (in the case of loops) how to collect the individual answers into a result.

1. Write a loop that says hello 10 times and get the same result with a mapping operation.
(Write a command that displays "hello", then cut and paste it into the body of FOR-EACH or APPLY-FUNCTION)

2. Write a loop that prints the name of each cyanobacterium, followed by its genome size (length) and the number of genes it possesses. Do the same thing by mapping.

3. Genomes are not random (if they were, we wouldn't be having this conversation). Write a loop that counts the actual occurrences of all 6-nucleotide sequences in the genome of the cyanobacterium *Anabaena* PCC 7120 (A7120) composed solely of C and G (confined to this subset to limit the time of execution). Then sort the list of counts so you can try to make sense out of them. Here's how:

   a. Figure out how to get the COUNT-OF a specific hexanucleotide sequence in A7120 (you've done something like this before in *What is a Gene?*). Put this in the body of a loop, and iterate over all possible hexanucleotides. The function ALL-COMBINATIONS-OF (STRING-SEQUENCE menu, String-Production submenu) will be helpful. COLLECT a LIST, consisting of the sequence and the counts.

   b. DEFINE the resulting list as a variable named however you wish. The PREVIOUS-RESULT function may be convenient. You can find it on the OTHER-FUNCTIONS menu.

   c. SORT the list, making use of the SORT function found in the LIST-TABLES menu, List-Production submenu. Use the BY-POSITION option to specify that you're sorting by the second element of the list, i.e., the number of counts. SORT it both ASCENDING and DESCENDING, in order to see the sequences that are most common and those that are least common.

   d. What generality can you discern regarding the least common hexanucleotides? Why do you suppose they are least common?

4. Write a loop that calculates the probability of encountering a given nucleotide sequence (e.g. "CGCGCG") in a genome with [A] = 0.2938 (the actual frequency of A in *Anabaena* PCC 7120). The template on the following page may be helpful. Use the function to compare the counts you got in problem 3 to what you would predict by chance. (Of course the loop is gross overkill! There are much easier ways of doing this.)

(*Suggestion for Problem 4)*