The pickled data frames for Eukaryotic species are located in:
/home/users/fardokht/EUK_pickled_df

The script gets a file with qseqids separated by '\n'. Next, it looks for qseqids in the pickled data frames, for equal qseqid from the input and pickled data frames, the sseqids, and e-values are extracted. Then, the script gets proportion of count (how many times each sseqid is hit by query sequences divided by the total number of query sequences) and the average mean value for each sseqid.

The aforementioned steps result into a data frame with three columns of 'accession, 'count_proportion', and 'mean_evaue'. The script uses this data frame and gets the taxon id fro each accession number. The taxon ids are then converted to full taxonomic rank. The script is mainly looking for 'phylum' in the full taxonomy of each taxon id, meaning that if 'phylum' is found in the full taxonomy, the name under 'phylum' is taken, otherwise the very first element from the full taxonomy is retrieved (For example, superkingdom, or kingdom, …).

Now, the data frame has six columns of 'accession, 'count_proportion', 'mean_evaue', 'taxid', 'rank', 'name'. The column 'rank' stores the taxonomic rank (ex: phylum, kingdom, …) and the column 'name' stores the name found under 'rank'. Next, the data frame is sorted by two columns of 'count_proportion' (values descending), 'mean_evaue' (values ascending).

In the very last step, the script, goes through the dataframe, and selects the best three hits and two random hits for each 'name'.

I have created a fixed list called phylum_only. Phylum_only contains all the phylums found under Bacteria in NCBI.

phylum_only = [**'Acidobacteria'**, **'Aquificae'**, **'Caldiserica'**, **'Candidatus Cryosericota'**, **'Calditrichaeota'**, **'Chrysiogenetes'**, **'Coprothermobacterota'**, **'Deferribacteres'**, **'Dictyoglomi'**, **'Elusimicrobia'**, **'Bacteroidetes'**, **'Balneolaeota'**, **'Candidatus Kapabacteria'**, **'Chlorobi'**, **'Ignavibacteriae'**, **'Rhodothermaeota'**, **'candidate division LCP-89'**, **'candidate division Zixibacteria'**, **'Candidatus Cloacimonetes'**, **'Candidatus Fermentibacteria'**, **'Candidatus Hydrogenedentes'**, **'Candidatus Kryptonia'**, **'Candidatus Latescibacteria'**, **'Candidatus Marinimicrobia'**, **'Fibrobacteres'**, **'Gemmatimonadetes'**, **'Fusobacteria'**, **'Krumholzibacteriota'**, **'Candidatus Tectomicrobia'**, **'Nitrospinae'**, **'Nitrospirae'**, **'Proteobacteria'**, **'Candidatus Abyssubacteria'**, **'Candidatus Aureabacteria'**, **'Candidatus Omnitrophica'**, **'Chlamydiae'**, **'Kiritimatiellaeota'**, **'Lentisphaerae'**, **'Planctomycetes'**, **'Verrucomicrobia'**,

'Spirochaetes', **'Synergistetes'**, **'Abditibacteriota'**, **'Actinobacteria'**, **'Armatimonadetes'**, **'Candidatus Dormibacteraeota'**, **'Candidatus Eremiobacteraeota'**, **'Chloroflexi'**, **'Candidatus]**

The script iterates through this list and if elements of this list are found in the final data frame, it stores the respective accession number and the position found in the original sorted data frame.