Integrating the output of two genefinders, tRNAscan-SE and Aragorn

Input: Output of tRNAscan and Aragorn on each genome saved as separate files.

TSE outputs are read in two different formats: "*.tse.out" and "*.SS.tse.out"

ARA outputs are read in one format ".ara.out".

Script has four functions:

1. Integrate_Tse_Ara() 2. making_ara_df() 3. making_tse_df() 4. integrate() 5. formatoutput()

For each genome the fallowing pipeline will be run to make the integrated gene file.

1. making_ara_df :

This function will extract genes' information found by aragorn as a dataframe format.

The output will be in a table with columns:

aragenename , arasourceOrg ,araidentity , aradirection ,arabegin ,araend , araac, arasourceSO, arasourceseq ,arascore ,arageneseq , arageness

Note: this function will find the tmrnas and won't include them in the output!

I found just one tmrna which was unusually the longest gene we had. My guess is that it was the long gene we had in previous data.

2. making_tse_df:

This function will extract the genes' information found by tRNAscan in a dataframe format.

It reads each found gene along with its secondary structure from two files: "*.tse.out" and "*.SS.tse.out". The output will be in a table with columns:

tsegenename, tsesourceOrg, tseidentity, tsedirection, tsebegin, tseend, tseac, tsesourceseq, tsescore, tsegeneseq, tsegeness, tseintronbegin, tseintronend, tseacloc, note

NOTE1: identities that were not determined by tse are shown as "Undet" and their anticodon is shown as "NNN". ALL these genes were either not found by Aragorn or they were found but their identity was not certain like "?(Arg|Glu)". However, there were genes which has two option identity by Ara but Tse had just one certain identity for them.

NOTE2: We had few anticodons for genes found by ara with four letters like "(gtag)"

The last column, **note**, will take the value psudo if the gene is known as Psudogene by tse.

If gene is not found by tse, but only is found by aragorn, it will take the value "notfound" and vice versa.

3. integrate:

This function will take the tables made by previous two functions as input. It will switch the start and end positions found by tse, if they are on reverse strand. Using the functions "Granges" and "findOverlaps" which are implemented based on the idea of interval tree to find overlapped genes using their coordinates (sourceseq,position,strand), we found genes that are found by both gene finders (they overlap), saved in a dataframe called "overlapdf". Using the function "setdiff" we found genes that are found by only one of the genefinders. Finally, we bind both as one dataframe called "integrated_Tse_Ara". At this step another column called "foundby" is added to the table which shows by which genefinder the gene is found. The value for this column can be "both", "tse", or "ara".

Geneid at this step is made by SourceOrg + sourceseq + index of gene within that organism(not sourceseq!) seprated by "_".

4. formatoutput:

This function will format the **integrated_Tse_Ara** dataframe as a fixed width format to print out.

Also the integrated_Tse_Ara table is printed as four files: **secondaryS.txt, identities.txt, coordinates.txt, introns.txt.** This is the summery of what each file contains:

1. introns.txt :

"geneid" "tseintronbegin" "tseintronend"

2. coordinates.txt:

"geneid" "sourceOrg" "sourceseq" "sourceSO" "direction" "tsebegin" "tseend" "arabegin" "araend"

3. identities.txt:

"geneid" "tseidentity" "araidentity" "tseac" "araac" "tseacloc" "arascore" "tsescore" "foundby"

4. secondaryS.txt

geneid, tseseq, tsess, araseq, arass

Column note will be added to the files!