Gene Annotation

Fatemeh

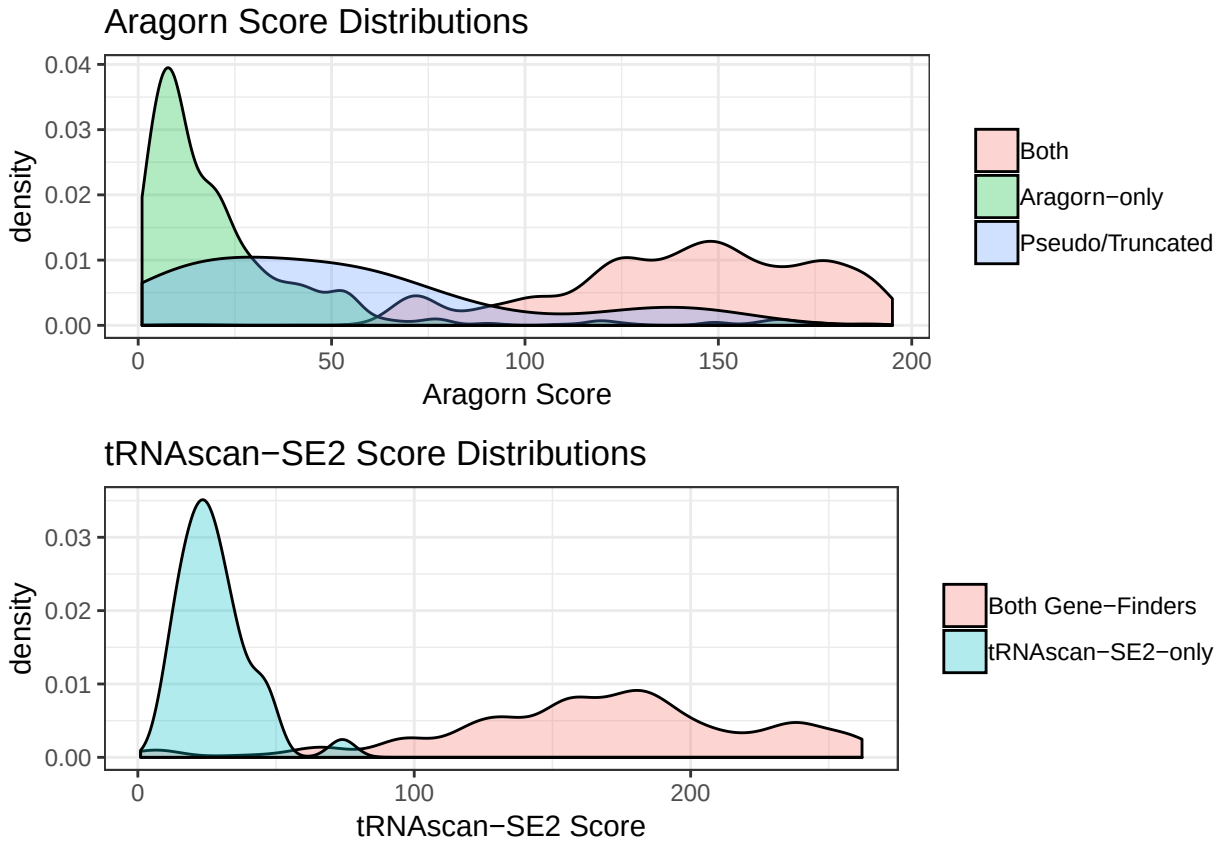
June 17, 2019

FinalGeneSetAnnotation

This script has the fallowing functions:

1. annotate.final.geneset.round1()
annotate.final.geneset.round1() will read the main integrated gene file from integrated_tse_ara.txt
Keep genes that either their ara score is > 106 or their tse score is > 49
add column: genefun and fill it in the fallowing order:
#1. genes with same tse and ara identity are assign the same idenitity
#2. genes with different identity, pseudo|truncated genes, genes with un assigned
identity and genes with letter N in their sequence are shown as ??
add column: note and fill it in the fallowing order:
#1. genes with the same ara and tse marked as "T"
#2. genes with unassigned identity by both tse and ara are set as "UnAssigned"
#3. genes with letter N in their sequence are set as "ContainsN"
#4. genes with unmatched identity between ara and tse are set as "Undet"
these top 4 cases had no overlap!

2. Score.visualization()
This functions read the main integrated gene file integrated_tse_ara.txt and shows the distribution of gene scores

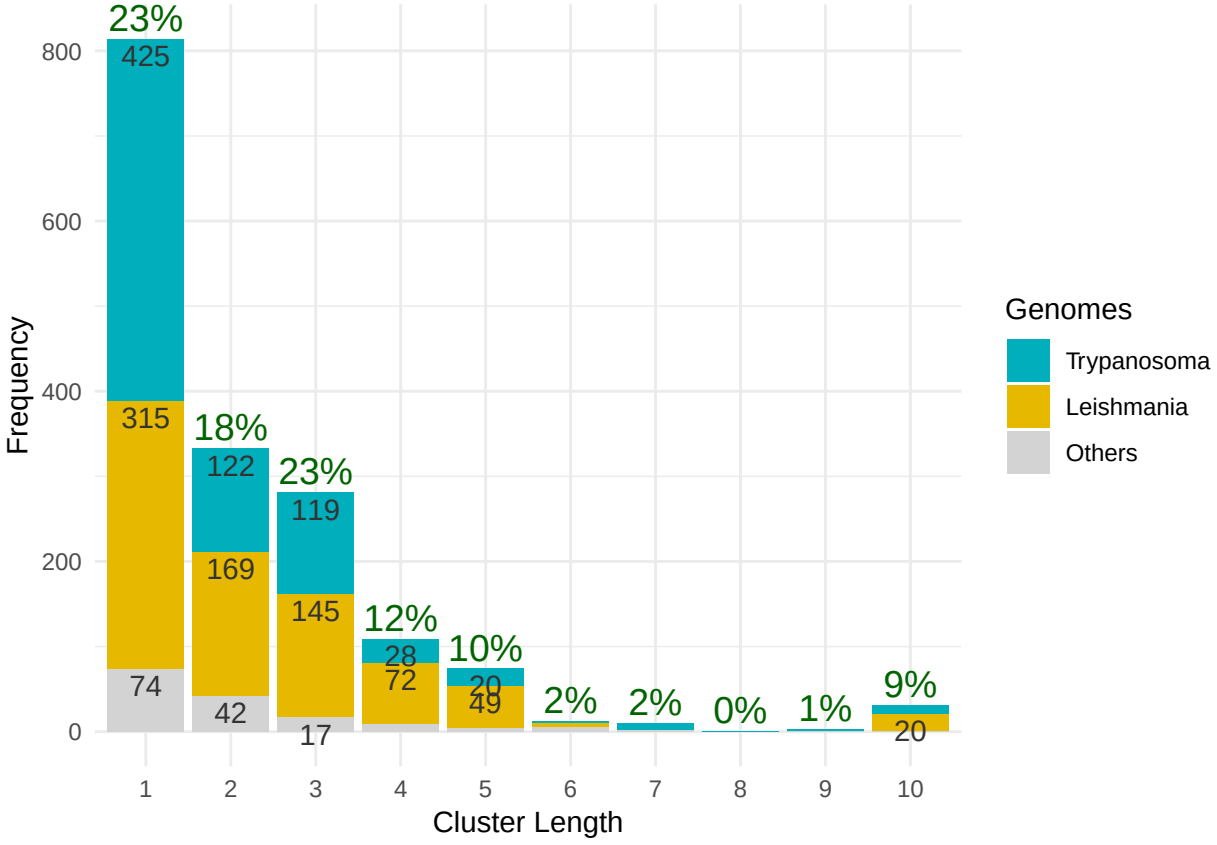

Aragorn Score Distributions
density
0.04
0.03
0.02
0.01
0.00
0
50
100
150
200
Aragorn Score
Both
Aragorn−only
Pseudo/Truncated
tRNAscan−SE2 Score Distributions
density
0.03
0.02
0.01
0.00
0
100
200
tRNAscan−SE2 Score
Both Gene−Finders
tRNAscan−SE2−only

2. create.summary.table()
   Read the annotated gene file (output of function annotate.final.geneset.round1) and Creates a summary table of genes after the score filtering.
   Last three columns are each for one gene set. Sets are defined as: a) Intersection Of two gene finders. Two genes are considered same gene if their coordinate overlaps at least one base. Displacement of overlapped genes between ARA and TSE does not pass 4bp. b) Union of two gene finders. c) Genes found by only ARA. Genes marked as # include: pseudo|truncated genes (6 genes), genes with different predicted identity by two genefinders(23 genes), genes with unassigned identity|anticodon by any of genefinders (2 genes), and genes with letter N in their sequence(we had 4 of these genes). we also had 1 genes preicted by only TSE labeled as # which is not shown in this table as a seperate column, however it is considered in the union set.

```
##          Annotation Intersection ARAonly Union
## 1             #tRNA         3579      36  3616
## 2             #N/#G           74      98    75
## 3  Min Gene Length            68      71    68
## 4  Max Gene Length            89     206   206
## 5           %intron             2      28     3
## 6                %G            32      33    32
## 7                %C            26      26    26
## 8                %T            23      23    23
## 9                %A            19      18    19
## 10                A           210       2   212
## 11                C            64       1    65
## 12                D           105       1   106
## 13                E           160       1   161
## 14                F           104       2   106
## 15                G           228       3   231
## 16                H            80       4    84
## 17                I           171       1   172
## 18                K           183       1   184
## 19                L           335       6   341
## 20                M            97       0    97
## 21                N           125       0   125
## 22                P           200       0   200
## 23                Q           161       0   161
## 24                R           348       2   350
## 25                S           228       7   235
## 26                T           218       4   222
## 27                V           236       0   236
## 28                W            52       1    53
## 29                X            76       0    76
## 30                Y            88       0    88
## 31                Z            76       0    76
## 32                #            34       0    35
```

3. clustersize.dist.visualize()
   Read the annotated gene file (output of function annotate.final.geneset.round1) and visualizes the Cluster size distribution for three categories of TryTryp genomes. Labels in green on top of each bar show the percentage of total number of genes as cluster of a specific length. Each color refers to one category of TriTryp genomes. Numbers within each color section of the bar shows the counts of clusters with a specific length.

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```



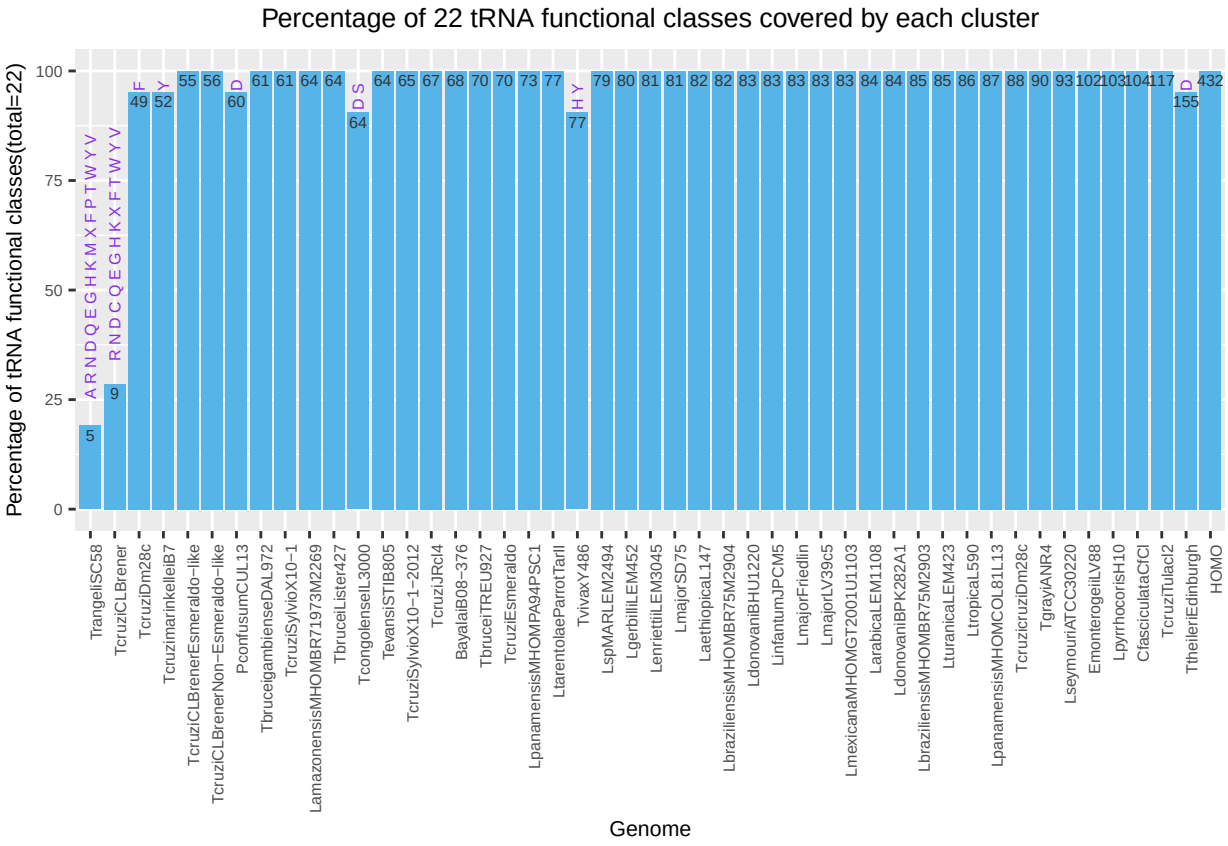
3. prepare.tsfm.input()
Read the annotated gene file (output of function annotate.final.geneset.round1), removes genes with ambiguty marked as # (35 genes), genes with function Sec (76 genes) and genes from two genomes genes of genomes TrangeliSC58 and TcruziCLBrener and writes the selected genes in file tsfm_input_geneset.txt (17 genes). tsfm_input_geneset.txt file should be used for further alignment.
We have 3478 genes left to be aligned

The gene set tsfm_input_geneset.txt is passed to the script TriTrypAlignment.R to be aligned with Human tRNA genes.

TriTrypAlignment.R

Alignment Steps:
1. Genes from tsfm_input_geneset.txt are read and functional classes are added at the end of the geneID
2. Variable arms are removed based on reported secondary structure from genefinders
3. Gene introns are removed based in the secondary sctructure
4. the result is merged with the Human tRNA genes (the headers for Homo genes are also updated) and the result is saved in file coveainput.fasta.
5. coveainput.fasta is aligned to the Eukaryote model using covea
6. the result (Aligned_TriTryp_Homo.covea) is edited based on the fallowing criteria in order:
a) sites that have more than 98% gap are removed
b) sequences with more than 3 gaps are removed
c) sequences with two or more gaps next to eachother are removed
7. the alignment result is saved as fasta file in Aligned_TriTryp_Homo.fasta, with secondary structure saved as Aligned_TriTryp_Homo_structfile.txt

SplitAlignedGenes.R
The alignment Result Aligned_TriTryp_Homo.fasta will be passed to this script to be splitted either by genome, or clusters of genomes.
The result fasta file for each genome is saved as a file in tsfm/input folder.
The missing functional class for each genome or cluster of genomes is visualized as a bar plot.
(**********This figure is not updated yet!**********)



Fasta gene files will be splitted based on tRNA functional class by running script splitFuncClass.sh.