

# Gene Annotation

*Fatemeh*

*June 17, 2019*

## FinalGeneSetAnnotation.R

This script has the following functions:

### 0. `annotate.final.geneset.round1`

**Input:** `integrated_tse_ara.txt`

It reads the main integrated gene file from `integrated_tse_ara.txt`

Keeps genes with either their ara score is  $> 106$  or their tse score is  $> 49$

Column **genefun** is added to the genefile table and filled in the following order:

1. genes with same tse and ara identity are assign the same identity
2. genes with different identity, pseudo|truncated genes, genes with un assigned identity and genes with letter N in their sequence are shown as #

Column **note** is added and fill it in the following 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!

I kept track of the remained number of genes at each step and wrote it as comments.

### 1. `genome.nuc.composition`:

**Input:** `GenomeNucComposition.txt` and **output of function** `annotate.final.geneset.round1`

**output:** output is saved in latex format in this file `GenomeNucComposition_Latex.txt`

The complete script for calculating the nucleotide composition is in `FinalGeneSetAnnotation.R`. Reading 46 genomes and calculating the compositions is a bit time consuming, so I have already done that part in function `genome.nuc.composition()` in the main R script and saved the table as `GenomeNucComposition.txt`. Here, in function `genome.nuc.composition2()` I only read file `GenomeNucComposition.txt`, and marge it with a new column called `genecounts` which I calculate from the output of function `annotate.final.geneset.round1()`. In the end the file is written in latex format.

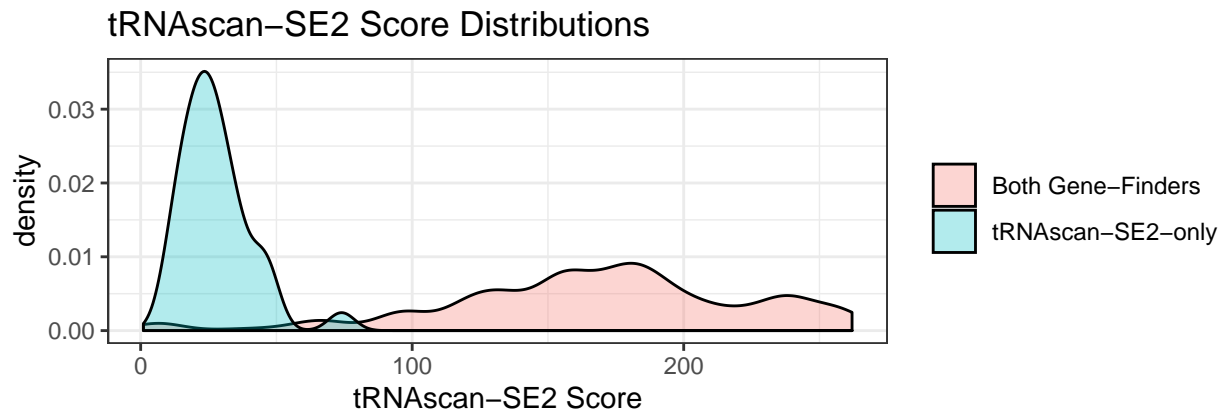
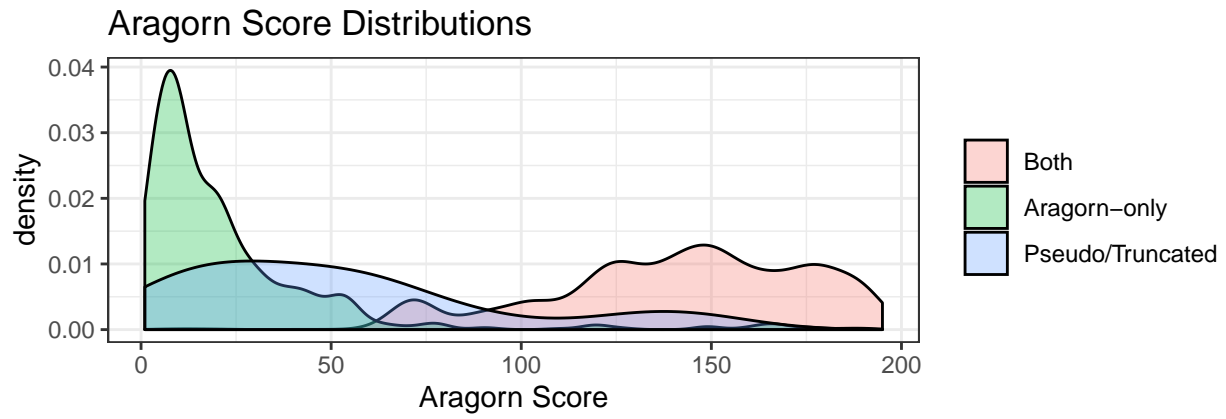
##	organism_shortname	Aperc	Tperc	Cperc	Gperc	GCp	genecounts
##	TrangeliSC58	24	23	27	26	53	6
##	TcruziCLBrenner	23	23	27	27	53	18
##	TcruziDm28c	25	25	26	25	50	51
##	TcruzimarinkelleiB7	22	22	23	23	45	57
##	TcruziCLBrennerEsmeraldo-like	20	20	20	20	40	57
##	TcruziCLBrennerNon-Esmeraldo-like	21	21	22	22	43	57
##	PconfusumCUL13	18	18	28	28	57	61

##	TbruceigambienseDAL972	26	26	24	24	47	64
##	LamazonensisMHOMBR71973M2269	20	20	30	30	59	66
##	TevansiSTIB805	27	27	23	23	47	67
##	TbruceiLister427	26	26	22	23	45	67
##	TcruziSylvioX10-1	24	23	25	25	50	69
##	BayalaiB08-376	22	22	27	27	55	69
##	TcruziSylvioX10-1-2012	24	24	26	26	51	72
##	TcongolenseIL3000	21	21	20	20	40	72
##	TbruceiTREU927	27	27	23	23	45	73
##	TcruziJRcl4	24	23	26	24	50	74
##	TcruziEsmeraldo	23	22	24	23	47	74
##	LpanamensisMHOMPA94PSC1	21	21	28	28	56	74
##	LtarentolaeParrotTarII	21	21	27	27	55	79
##	LspMARLEM2494	20	20	30	29	59	80
##	LgerbilliLEM452	20	20	30	29	59	81
##	LmajorSD75.1	20	20	30	30	59	82
##	LenriettiiLEM3045	20	20	29	29	59	82
##	Tvivaxy486	21	21	23	23	46	82
##	LbraziliensisMHOMBR75M2904	21	21	29	29	58	83
##	LaethiopicaL147	19	20	30	29	59	83
##	LdonovaniBHU1220	19	20	29	29	57	84
##	LinfantumJPCM5	20	20	30	30	60	84
##	LmajorFriedlin	20	20	30	30	60	84
##	LmexicanaMHOMGT2001U1103	20	20	30	30	60	84
##	LmajorLV39c5	20	20	29	29	59	84
##	LarabicaLEM1108	20	20	29	29	58	85
##	LdonovaniBPK282A1	19	20	29	29	57	85
##	LturanicaLEM423	19	20	30	29	59	86
##	LbraziliensisMHOMBR75M2903	19	20	27	27	53	86
##	LtropicaL590	19	19	29	28	57	87
##	LpanamensisMHOMCOL81L13	21	21	29	28	57	88
##	LseymouriATCC30220	22	22	28	28	55	94
##	TgrayiANR4	23	23	27	27	54	95
##	TcruzicruziDm28c	24	24	26	26	52	97
##	EmonterogeiiLV88	23	23	26	26	52	104
##	LpyrrhocorisH10	21	22	28	28	56	104
##	CfasciculataCfCl	21	22	29	28	57	105
##	TcruziTulac12	22	21	23	23	46	121
##	TtheileriEdinburgh	26	26	17	17	35	159

## 2. Score.visualization

Input: Input: integrated\_tse\_ara.txt

This functions read the main integrated gene file integrated\_tse\_ara.txt and shows the distribution of gene scores



### 3. create.summary.table

**Input:** output of function `annotate.final.geneset.round1()`

**output:** output is saved in latex format in this file **SummaryTable\_\_Latex.txt**

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 sepearte column, however it is considered in the union set.

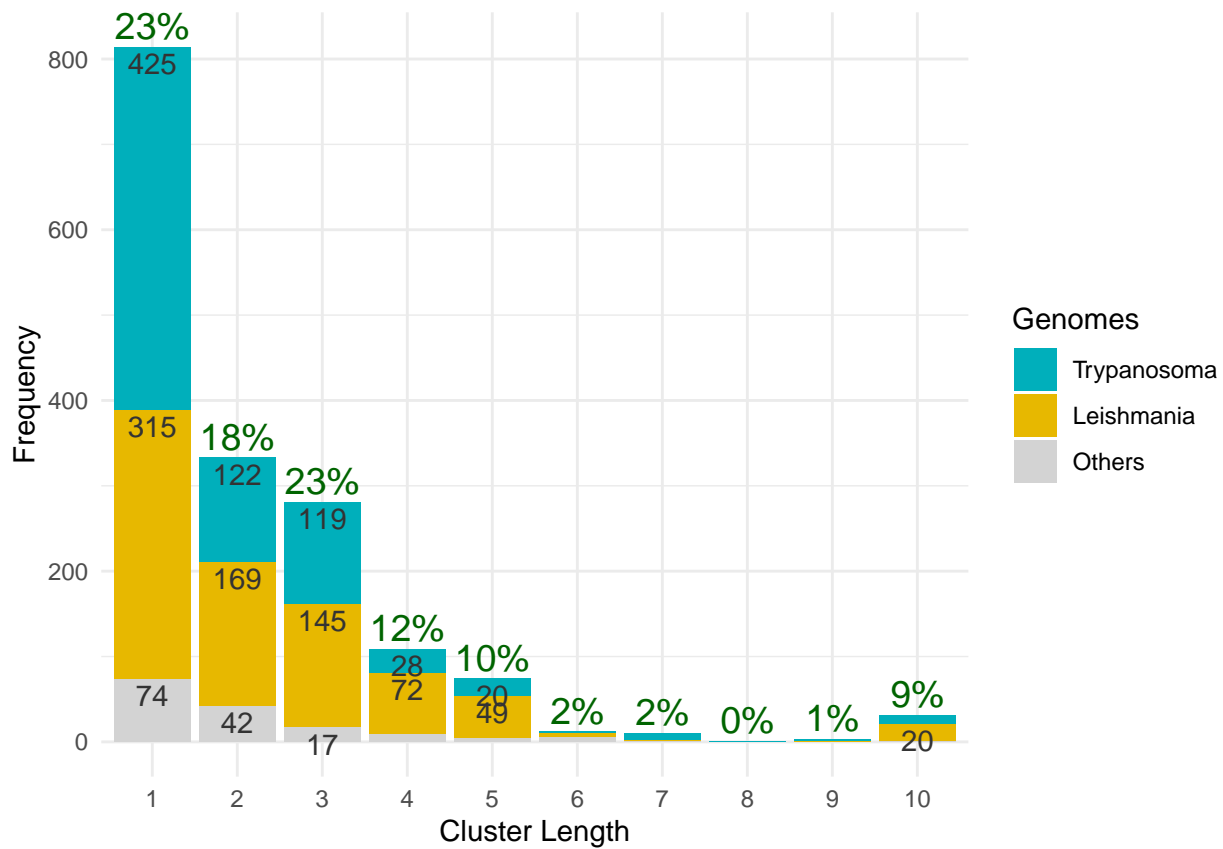
##	Annotation	Intersection	ARAonly	Union
##	#tRNA	3579	36	3616
##	#N/#G	74	98	75
##	Min Gene Length	68	71	68
##	Max Gene Length	89	206	206
##	%intron	2	28	3
##	%G	32	33	32
##	%C	26	26	26
##	%T	23	23	23
##	%A	19	18	19
##	A	210	2	212
##	C	64	1	65
##	D	105	1	106
##	E	160	1	161
##	F	104	2	106
##	G	228	3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

#### 4. clustersize.dist.visualize

**Input:** output of function `annotate.final.geneset.round1()`

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.
```



## 5. prepare.tsfm.input

**Input:** output of function `annotate.final.geneset.round1()`

**Output:** `tsfm_input_geneset.txt`

Read the annotated gene file, removes **genes with ambiguity** marked as # (35 genes), genes with function **Secs** (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 for the purpose of creating CIFs.

We have **3478** genes left in this file to be aligned

## 6. write.tsfm.input.in.fasta()

This function will write the `tsfm_input_geneset.txt` file into a fasta format: **final\_CIF\_geneset.fasta**

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
  - 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.

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