**User guide**

**0.1.0.20170612\_beta**

This perl small program can analyse the gtf file (gff version 2.5), which is combined by the gtf\_rd287.pl and its library gtf\_file\_processing.pm

This two perl script should be in the same path of the computer.

**Usage: gtf\_rs206.pl options: [-g,-e,-a,-n,-h,-b,-l,-f,-m,-s,-t]**

**filename (optional, the suffix must be “.gtf”) genename[-s only]**

**Help information [-h]**

Multiple flags can be used at the same time.

The speed of using -g -e -a -n -l together is modified to be faster by set them in the same get\_result\_basic method.

The default setting is to analyse the testfile genes.gtf, people can change it after seeing the welcome message or input in the command argument.

Once the program start without input filename in command argument, it will ask user whether to change the input file, and ask for a wrong suffix.(must end with .gtf)

**Flags**

**g** Reports the number of [g]enes in the gtf file.[-g] done by the get\_result\_basic in the library.

Use hash set to calculate the unique number of genes in the file.

**e** Reports the number of [e]xons.[-e] done by the get\_result\_basic in the library.

Use hash set to calculate the unique number of exons in the file. The gene\_id is assigned by the coordinate number to avoid count the same exon for several times.

**a** Reports the [a]verage exon length. [-a] done by the get\_result\_basic in the library.

The fast way of calculate the average length, especially when using multiple flags but has the risk of data overflow.[big input file or when modified this script into other programming language]

**n** Reports the gene with the highest [n]umber of exons. [-n] done by the get\_result\_basic in the library and some lines in the gtf file it self.

The reason why the -n flag work is splited is to avoid making multiple flags works too slow. Because this flag is designed to have the ability to report the GENES rather than a random one gene with highest number of exons. (when there is some genes share the most exon number)

**t** Get the number of [t]ranscripts [-t]

Use the hash set to count the number of transcripts.

**b** Do the average calculate of exon avoiding data overflow with a [b]righer algorithm. [-b]

Use a better algorithm to calculate the average exon length to avoid dataoverflow but slower especially when applied with other flags together.

**l** Calculate the proportion between total [l]ength of exon/ total [l]ength of gene. [-l]

A rough estimate of the proportion. If it is too far away to 3-10, the data might have problem, or the data is really special.

**m** Reports the [m]ean number of gene length avoiding data overflow. [-m]

Use the algorithm like -b to calculate the average length of genes, which avoid the data overflow. No faster version for this is because this function is not that suitable to be added in the get\_result\_basic’s function, so the meaning of making a faster but more risk way is less.

**s** [S]earch the information of the query gene name agains the infile [-s] and out put it in to a outfile:

The file of -s flag is in the same path as the input file, and the output filename format is "<input file>\_<input gene>.out" as it will shown on the terminal.

It will report how many lines in the inputfile is about the query gene.

How many exons it has.

How many transcript it has, and their id.

The length of the input gene (for the gene has no start codon and stop codon recorded in the file, this line will not appear.)

The proportion length\_exon/length\_gene of that gene,(if there is not start/stop codon recorded in the file, this line will not appear. But some time it will over 100% for some exons record is out of the coordinate range of its corresponding gene start/stop region )

How many CDS that gene has, and their detail (the coordinate, the score, the strand, the CDS number,and the length of this CDS)

**Notice things:**

The input file can be a path, such as ./1/genes.gtf

when using the -s, there is some genes has over 100% proportion about the exon/gene.

This is because some exons is out of range of its corresponding gene, like line 25 to line 33 about Sox17 in the testfile.

That means the -l flag is not that precise as well. What is more, some gene in the file did not have start codon and stop codon record, so it length is not measurable, but it increase the number of gene.

However, the -m flag use the way that only calculate the mean length of genes with start and stop codon.