Analyze gene information from fly reference transcriptome GTF file

Jun Yin

05/01/18

GTF file: tab-delimited text format

Reference transcriptome file is usually stored as GTF or GFF file

	chr	source	feature	begin	end	score	strand	frame	attribute
0	3R	FlyBase	gene	567076	2532932	.	+	.	gene_id "FBgn0267431"; gene_name "Myo81F"; gen...
1	3R	FlyBase	transcript	567076	2532932	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
2	3R	FlyBase	exon	567076	567268	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
3	3R	FlyBase	exon	835376	835491	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
4	3R	FlyBase	CDS	835378	835491	.	+	0	gene_id "FBgn0267431"; transcript_id "FBtr0392...

```
gene_id "FBgn0267431"; gene_name "Myo81F"; gene_source "FlyBase"; gene_biotype "protein_coding";
```

GTF/GFF file could reach to several Gb. Directly read it will use all of memory.

Desired output file

Input file

GTF

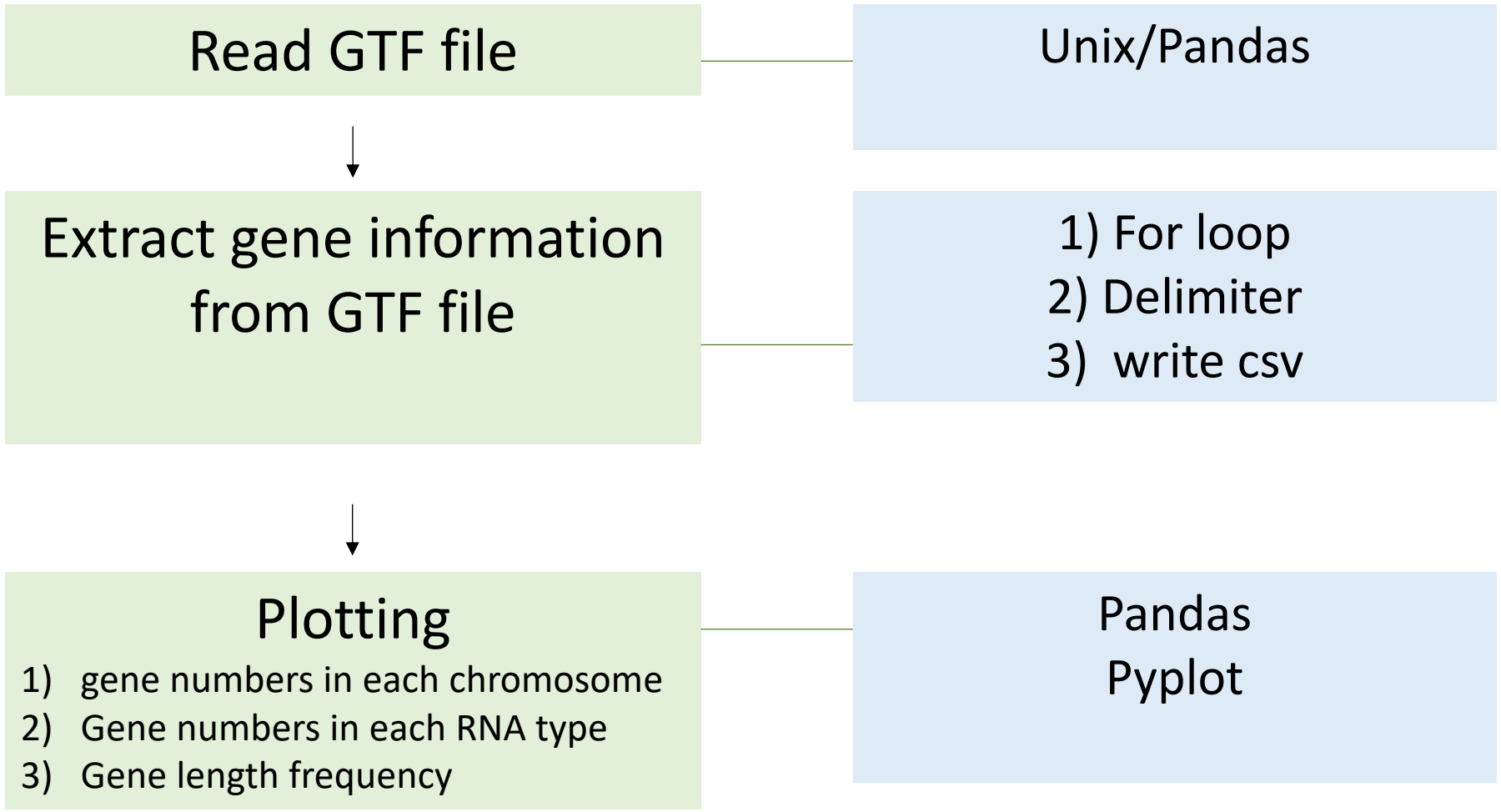
4 5 6 1 2 3

	chr	source	feature	begin	end	score	strand	frame	attribute
0	3R	FlyBase	gene	567076	2532932	.	+	.	gene_id "FBgn0267431"; gene_name "Myo81F"; gen... "Protein coding"

Output file

	gene_id	gene_name	gene_type	chromosome	begin	end	gene_length
0	FBgn0267431	Myo81F	protein_coding	3R	567076	2532932	1965857
1	FBgn0085804	CR41571	pseudogene	3R	722370	722621	252
2	FBgn0039987	CR12798	pseudogene	3R	1031171	1031354	184
3	FBgn0267798	CR46123	ncRNA	3R	1366234	1366601	368
4	FBgn0267797	CR46122	ncRNA	3R	1865108	1866008	901

Workflow


Read GTF file
Unix/Pandas
Extract gene information from GTF file
1) For loop
2) Delimiter
3) write csv
Plotting
1) gene numbers in each chromosome
2) Gene numbers in each RNA type
3) Gene length frequency
Pandas
Pyplot

Use pandas to read whole GTF file

In [2]:
```
#read whole GTF file as a table by using "pd.read_csv" or "pd.read_table" function
import pandas as pd
GTF=pd.read_csv('Fly_BDGP6.91.gtf',sep='\t',header=None)
GTF.columns=["chr","source","feature","begin","end","score","strand","frame","attribute"]
GTF.head()
```

Out[2]:

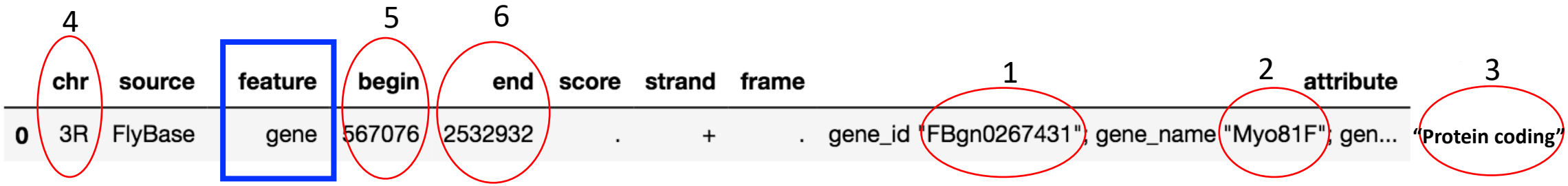
	chr	source	feature	begin	end	score	strand	frame	attribute
0	3R	FlyBase	gene	567076	2532932	.	+	.	gene_id "FBgn0267431"; gene_name "Myo81F"; gen...
1	3R	FlyBase	transcript	567076	2532932	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
2	3R	FlyBase	exon	567076	567268	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
3	3R	FlyBase	exon	835376	835491	.	+	.	gene_id "FBgn0267431"; transcript_id "FBtr0392...
4	3R	FlyBase	CDS	835378	835491	.	+	0	gene_id "FBgn0267431"; transcript_id "FBtr0392...

How many features in GTF file

```
GTF['feature'].value_counts()
```

```
exon                187373
CDS                 160859
five_prime_utr       46299
transcript           34767
three_prime_utr      33892
start_codon          30492
gene                 17737
Selenocysteine           4
```

Python script to extract desired information




```
# read GTF file, extract information of gene_id, gene_name, gene_type, chr,  begin, end, gene_length

import csv
with open("Fly_BDGP6.91.gtf") as fp:
    for line in fp:
        cols = line.strip().split("\t") # strip will get rid of \n at the end of each line)
        # only process lines with feature as "gene" in the third column
        if cols[2]=="gene":
            # parse attribution column
            s = next(csv.reader([cols[8]], delimiter='"'))
            # calculate gene length
            gene_length = abs(int(cols[4])-int(cols[3]))+1
            with open("result_BGDR_GTF.csv",'a') as f:  # 'a' will append each line. Don't rerun.
                w = csv.writer(f)
                #write gene_id, gene_name, gene_type, chr,  begin, end, gene_length
                w.writerow([s[1], s[3], s[7],cols[0], cols[3], cols[4],gene_length])
fp.close
f.close()
```

Work csv file with data frame in pandas

In [5]:
```
# reset column names
GTF.columns=["gene_id","gene_name","gene_type","chromosome","begin","end","gene_length"]
GTF.head()
```

Out[5]:

	gene_id	gene_name	gene_type	chromosome	begin	end	gene_length
0	FBgn0267431	Myo81F	protein_coding	3R	567076	2532932	1965857
1	FBgn0085804	CR41571	pseudogene	3R	722370	722621	252
2	FBgn0039987	CR12798	pseudogene	3R	1031171	1031354	184
3	FBgn0267798	CR46123	ncRNA	3R	1366234	1366601	368
4	FBgn0267797	CR46122	ncRNA	3R	1865108	1866008	901

Replace all numerical chromosome to unlocalized chromosome

```
# change those unknown chromosomes to "unlocalized"
GTF.loc[~GTF['chromosome'].isin (['2L','2R','3L','3R','4','X','Y','rDNA','mitochondrion_genome']),"chromosome"]="unloca
GTF.loc[GTF['chromosome']=="mitochondrion_genome","chromosome"]="Mito"
GTF.groupby('chromosome').size()
```

Out[12]:

chromosome	
211000022278279	1
211000022278436	1
211000022278449	1
211000022278760	1
211000022279165	1
211000022279188	1
211000022279264	1
211000022279392	1
211000022279681	1
211000022280328	1
211000022280341	1
211000022280347	1
211000022280481	1
211000022280494	2
211000022280703	1
2L	3496
2R	3621
3L	3457
3R	4191
4	111
Unmapped_Scaffold_8	2
X	2671
Y	113
mitochondrion_genome	38
rDNA	21

Out[77]:

chromosome	
2L	3496
2R	3621
3L	3457
3R	4191
4	111
Mito	38
X	2671
Y	113
rDNA	21
unlocalized	18

Convert row names to a new column

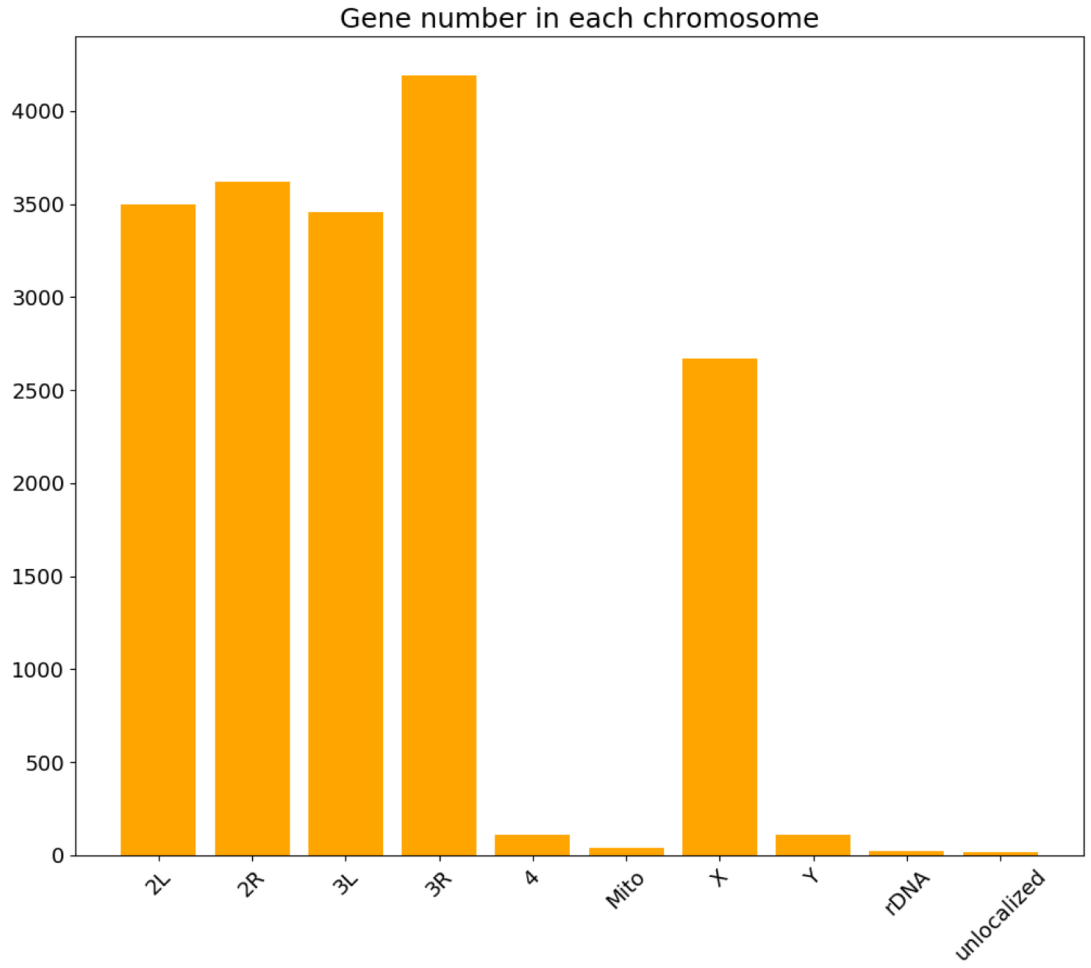
In [79]:
```
#convert the row names to a column
Df_Chr.index.name = 'chromosome'
Df_Chr.reset_index(inplace=True)
#Df_Chr.shape
Df_Chr.columns=['chr','gene number']
Df_Chr
```

Out[79]:

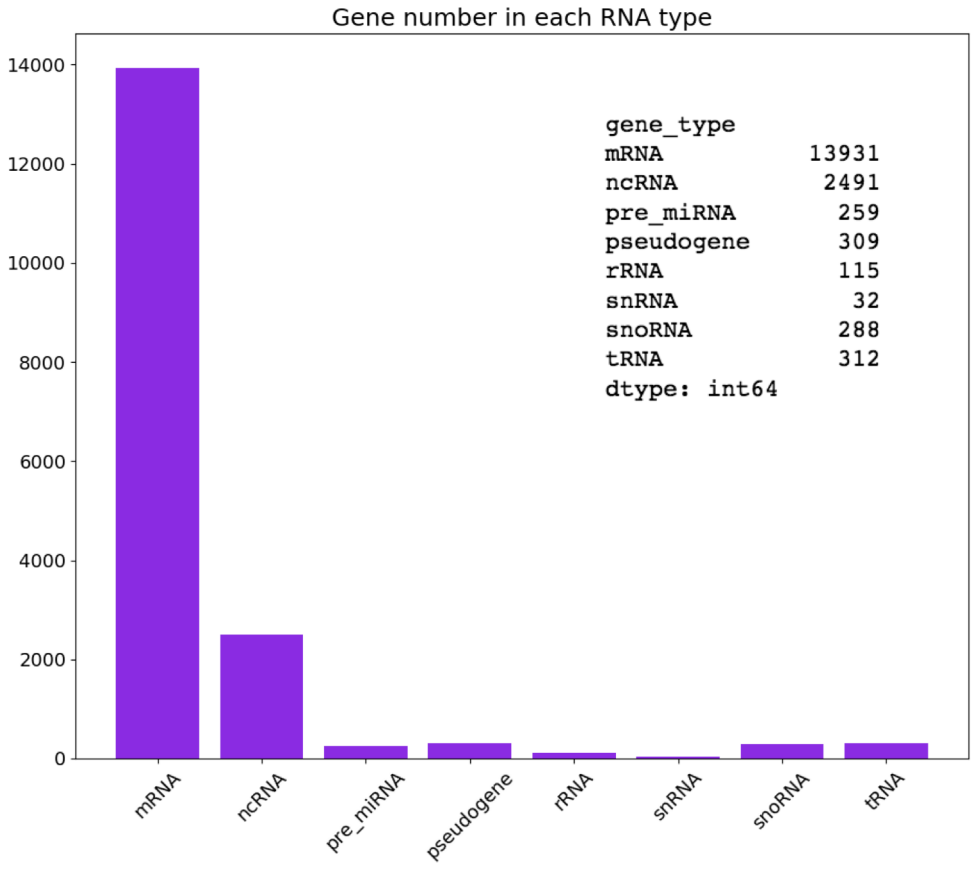
	chr	gene number
0	2L	3496
1	2R	3621
2	3L	3457
3	3R	4191
4	4	111
5	Mito	38
6	X	2671
7	Y	113
8	rDNA	21
9	unlocalized	18

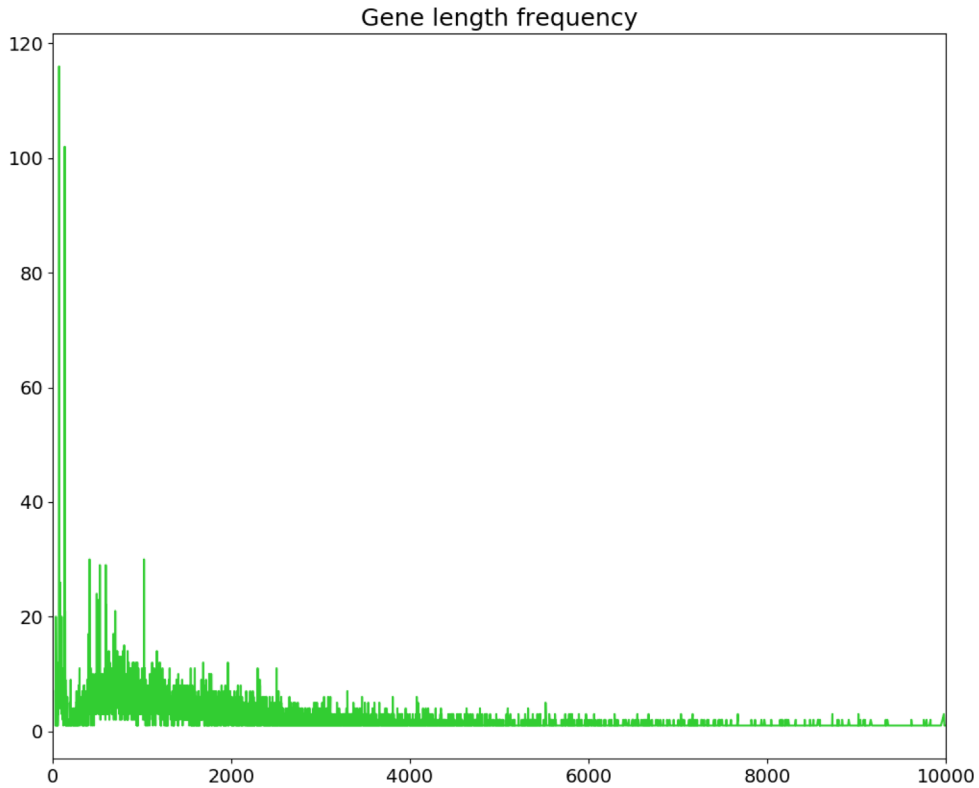
Plot gene numbers as plt with matplotlib.pyplot

In [160]:
```
# plot gene numbers in each cromosome

import matplotlib.pyplot as plt
import pylab as plt

plt.figure(figsize=(12,10))
plt.bar(Df_Chr['chr'],Df_Chr['gene number'],color='orange')
plt.xticks(rotation=45,fontsize=14)
plt.yticks(fontsize=14)
#plt.xlabel('Chromosome',fontsize=14) # add x axis label if you want to
plt.title('Gene number in each chromosome',fontsize=18)
plt.show()
#plt.savefig('Chr_gene_number_plot.png',dpi=100)
```



Plot gene number in each RNA type and gene length frequency





Thank you !