A1163_GC_content_features_GH

August 21, 2022

1 GC content of various features of A1163 annotation Aspergillus_fumigatusa1163.ASM15014v1.53.gff3

[1]:
```
# Import libraries
import pandas as pd
import numpy as np
import scipy.stats
```

2 get number of genes

cat Aspergillus_fumigatusa1163.ASM15014v1.53.gff3|awk '$3~ "gene" {print $0}'|wc -l

10109 (10106 gffcompare)

2.1 GC content genome 49.54%

in bash: cat Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa | grep -v ">" | awk 'BEGIN{a=0; c=0; g=0; t=0;} {a+=gsub("A",""); c+=gsub("C",""); g+=gsub("G",""); t+=gsub("T"," ");} END{print a,c,g,t}'

7346850 7202298 7214161 7333693 49.546%

[2]:
```
a=7346850
c=7202298
g=7214161
t=7333693
(c+g)/(a+c+g+t)*100
```

[2]:
```
49.54620067043333
```

2.2 Sequenced genome size - 29097002

2.2.1 GC content CDS - 53.91%

in bash: cat Aspergillus_fumigatusa1163.ASM15014v1.53.gff3|awk '$3~ "CDS" {print $0}'>A1163_CDS.gff3

conda activate aligners

bedtools nuc -fi Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa -bed A1163_CDS.gff3 > AF1163_CDS_GC.bed

conda deactivate

[3]:
```
# read GC files (Had to check which columns were which)
df=pd.read_csv("AF1163_CDS_GC.bed",comment="#", sep="\t",header=None)
# don't include anything with more than 5 ns
df=df[df.iloc[:,15]<5]
df["GC"]=df.iloc[:,13]+ df.iloc[:,12]
df["ACGT"]= df.iloc[:,11]+df.iloc[:,12]+df.iloc[:,13]+df.iloc[:,14]
sum(df.GC)/sum(df.ACGT)
```

[3]: 0.5391493140194445

[4]:
```
## percent sequenced genome coding 50.01%

sum(df.ACGT)/28809969
```

[4]: 0.5001624611258693

[5]:
```
df=pd.read_csv("A1163_mRNA.GC.bed",comment="#", header=None,sep="\t")

df=df[df.iloc[:,10]<5]
df["GC"]=df.iloc[:,12]+ df.iloc[:,13]
df["ACGT"]= df.iloc[:,11]+df.iloc[:,12]+df.iloc[:,13]+df.iloc[:,14]
sum(df.GC)/sum(df.ACGT)
```

[5]: 0.5319151239202807

2.3 GC content mobile elements - from annotation 33.66%

in bash: cat Aspergillus_fumigatusa1163.ASM15014v1.53.gff3|grep ena_mobile_element> a1163_mobile.gff3 conda activate aligners bedtools nuc -fi Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa -bed a1163_mobile.gff3 > a1163_mobile.gc.bed conda deactivate

[6]:
```
df=pd.read_csv("a1163_mobile.gc.bed",comment="#", header=None,sep="\t")
# don't include anything with more than 5 ns
df=df[df.iloc[:,15]<5]
df["GC"]=df.iloc[:,13]+ df.iloc[:,12]
df["ACGT"]= df.iloc[:,11]+df.iloc[:,12]+df.iloc[:,13]+df.iloc[:,14]
sum(df.GC)/sum(df.ACGT)
```

[6]: 0.3366425454899046

2.4 size of assembly gaps -106411

in bash: cat Aspergillus_fumigatusa1163.ASM15014v1.53.gff3|grep ena_assembly_gap >a1163_aasembly_gaps.gff3

[7]:
```
#size of assembly gaps
df=pd.read_csv("a1163_aasembly_gaps.gff3",comment="#", header=None,sep="\t")
sum(df.iloc[:,4]-df.iloc[:,3])
```

[7]:
```
106411
```

2.5 GC content of feature that are not coding not including mobile elements or assembly gaps

in bash:

bedtools intersect -a Aspergillus_fumigatusa1163.ASM15014v1.53.gff3 -b A1163_CDS.gff3 -v > A1163_transcribed_nc.bed

bedtools intersect -a A1163_transcribed_nc.bed -b a1163_mobile.gff3 -v > A1163_transcribed_nc_no_mob.bed

bedtools intersect -a A1163_transcribed_nc_no_mob.bed -b a1163_aasembly_gaps.gff3 -v > A1163_transcribed_nc_no_mob_no_ass.bed

#And now get GC content bedtools nuc -fi Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa -bed A1163_transcribed_nc_no_mob_no_ass.bed > A1163_transcribed_nc_no_mob_no_ass.GC.bed

[8]:
```
df=pd.read_csv("A1163_transcribed_nc_no_mob_no_ass.GC.bed",comment="#",␣
 ↪header=None,sep="\t")
# don't include anything with more than 5 ns
df=df[df.iloc[:,15]<5]
df["GC"]=df.iloc[:,13]+ df.iloc[:,12]
df["ACGT"]= df.iloc[:,11]+df.iloc[:,12]+df.iloc[:,13]+df.iloc[:,14]
df["GCP"]=df["GC"]/df["ACGT"]
df.columns= [     "chr",      "ena",      "feature",      "start",      "end", ␣
 ↪    "x",      "strand",      7,      "info",
            9,     10,     11,     12,     13,     14,     15,     16,     17,
        'GC', 'ACGT',  'GCP']

df_tRNA=df[df.iloc[:,2].apply(lambda x: "tRNA"in x)]
df_5prime=df[df.iloc[:,2].apply(lambda x: 'five_prime_UTR'in x)]
df_3prime=df[df.iloc[:,2].apply(lambda x: 'three_prime_UTR'in x)]
df_ncexon=df[df.iloc[:,2].apply(lambda x: 'exon'in x)]
df_pseudogene=df[df.iloc[:,2].apply(lambda x: 'pseudogene'in x)]
df_pseudogenic_transcript=df[df.iloc[:,2].apply(lambda x:␣
 ↪'pseudogenic_transcript'in x)]
df_ncRNA_gene=df[df.iloc[:,2].apply(lambda x: 'ncRNA_gene'in x)]
```

2.6 5' (#10) 3' (#8) > 30 nt GC-content 47.98%, 47.19% respectively

[9]:
```
len(df_3prime)
```

[9]: 28

[10]:
```
len(df_5prime)
```

[10]: 32

[11]:
```
sum(df_5prime.GC)/sum(df_5prime.ACGT)
```

[11]: 0.47980259550356424

[12]:
```
len(df_5prime[df_5prime.ACGT>30])
```

[12]: 10

[13]:
```
(sum(df_5prime[df_5prime.ACGT>30].GC))/(sum(df_5prime[df_5prime.ACGT>30].ACGT))
```

[13]: 0.4793958605258251

[14]:
```
(sum(df_3prime[df_3prime.ACGT>30].GC))/(sum(df_3prime[df_3prime.ACGT>30].ACGT))
```

[14]: 0.4719321148825065

[15]:
```
len(df_3prime[df_3prime.ACGT>30])
```

[15]: 8

2.7 GC content tRNAs 55.45% #175

[16]:
```
sum(df_tRNA.GC)/sum(df_tRNA.ACGT)
```

[16]: 0.5544775598520276

[17]:
```
len(df_tRNA.GC)
```

[17]: 175

2.8 INTRON GC 46.53% and sizes, 80.92 mean, sem (0.64)

in bash: ### extract splice sites HISAT2 (https://github.com/DaehwanKimLab/hisat2/blob/master/hisat2_extr
python hisat2_extract_splice_sites.py Aspergillus_fumigatusa1163.ASM15014v1.53.gtf > A1163_splice_sites.bed

2.9 get intron gc content

conda activate aligners bedtools nuc -fi Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa -bed A1163_splice_sites.bed > A1163_splice_GC.bed conda deactivate

[18]:
```
df=pd.read_csv("A1163_splice_GC.bed",comment="#", header=None,sep="\t")
df=df[df.iloc[:,10]<5]
df["GC"]=df.iloc[:,7]+ df.iloc[:,8]
df["ACGT"]= df.iloc[:,6]+df.iloc[:,7]+df.iloc[:,8]+df.iloc[:,9]
sum(df.GC)/sum(df.ACGT)
```

[18]:
```
0.4652782349941584
```

[19]:
```
df=df[df.iloc[:,10]<5]
df["GC"]=df.iloc[:,7]+ df.iloc[:,8]
df["ACGT"]= df.iloc[:,6]+df.iloc[:,7]+df.iloc[:,8]+df.iloc[:,9]
sum(df.GC)/sum(df.ACGT)
```

[19]:
```
0.4652782349941584
```

[20]:
```
np.mean(df.iloc[:,2]-df.iloc[:,1])
```

[20]:
```
80.91664055155124
```

[21]:
```
scipy.stats.sem(df.iloc[:,2]-df.iloc[:,1])
```

[21]:
```
0.6369252116687439
```

2.10 Get Intergenic region in bash - without assembly gaps and transposons = 45.5%

In bash: ### make genome -mobile elements - assembly and subtract genes ### first make a genome as a bed file conda activate GFF_utils gff2bed <a1163_mobile.gff3>a1163_mobile.bed gff2bed <a1163_aasembly_gaps.gff3>a1163_aasembly_gaps.bed conda deactivate

2.10.1 from top of gff3 file get info on chromosomes and make into a bed file

cat forA1163_genome.txt|awk 'BEGIN { OFS="^" } {print $2""0""$4}'> A1163_chromSizes.bed

2.10.2 need to sort the gene side

cat Aspergillus_fumigatusa1163.ASM15014v1.53.gff3|awk '$3~ "gene" {print $0}'>A1163_gene.gff3

2.10.3 need to make this in to a bed file

conda activate GFF_utils gff2bed < A1163_gene.gff3>A1163_gene.bed conda deactivate

2.11 now need to sort the bed file

#sort the bed file sort -V -k1,1 -k2,2 A1163_gene.bed >A1163_gene.sorted.bed conda activate aligners ### now can subtract to get intergenic regions bedtools subtract -a A1163_chromSizes.bed -b A1163_gene.sorted.bed> A1163_intergenic_regions.bed

2.11.1 now subtract mobile elements and assembly gaps

bedtools subtract -a A1163_intergenic_regions.bed -b a1163_mobile.bed>A1163_intergenic_no_transposons.bed bedtools subtract -a A1163_intergenic_no_transposons.bed -b a1163_aasembly_gaps.bed>A1163_intergenic_no_transposons_no_ass_gap.bed

2.11.2 now get gc content

conda activate aligners bedtools nuc -fi Aspergillus_fumigatusa1163.ASM15014v1.dna.toplevel.fa -bed A1163_intergenic_no_transposons_no_ass_gap.bed > a1163_intergenic.gc.bed

conda deactivate

[22]:
```
df=pd.read_csv("a1163_intergenic.gc.bed",comment="#", header=None,sep="\t")
# don't include anything with more than 5 ns
df=df[df.iloc[:,9]<5]
df["GC"]=df.iloc[:,6]+ df.iloc[:,7]
df["ACGT"]= df.iloc[:,5]+df.iloc[:,6]+df.iloc[:,7]+df.iloc[:,8]
sum(df.GC)/sum(df.ACGT)
```

[22]:
```
0.45525616162995786
```