A1163_UNION_after_TPM_GC-CONTENT_Lengths_GH

August 21, 2022

[1]:
```
# GC content  and Lengths of new candidate LncRNAs and potential coding genes␣
 ↪compared to transcripts in annotations
# Libraries needed
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats
from scipy.stats import mannwhitneyu
```

[2]:
```
# Novel Protein Coding
dfPP_All=pd.read_csv("formatted_A1163PT_UX_6_8_TPM_CUTOFF_6_8.fasta",␣
 ↪sep=("\t"), header=None).loc[::2]
dfPP_All2=pd.read_csv("formatted_A1163PT_UX_6_8_TPM_CUTOFF_6_8.fasta",␣
 ↪sep=("\t"), header=None).loc[1::2]
dfPP_All.columns=["info"]
dfPP_All=dfPP_All.reset_index()
dfPP_All["merger"]=dfPP_All.index
dfPP_All2.columns = ["seq"]
dfPP_All2=dfPP_All2.reset_index()
dfPP_All2["merger"]=dfPP_All2.index
dfPP_merge=dfPP_All.merge(dfPP_All2, on="merger")
dfPP_merge=dfPP_merge[["info", "seq"]]

dfPP_merge["classes"]=dfPP_merge.loc[:,"info"].str.
 ↪split("code=",expand=True)[1].str.split(";", expand=True)[0]
dfPP_merge["length"]=dfPP_merge.seq.apply(lambda x:len(x))
dfPP_merge["GC"]=dfPP_merge.seq.apply(lambda x:(x.upper().count('C'))+(x.
 ↪upper().count('G')))
dfPP_merge["GC_content"]=dfPP_merge["GC"]/dfPP_merge["length"]
```

[3]:
```
# Candidate LncRNAs
dfLNCR_All=pd.read_csv("formatted_A1163_ML_UX_LncRNA_T_TPM_CUTOFF_6_8.fasta",␣
 ↪sep=("\t"), header=None).loc[::2]
dfLNCR_All2=pd.read_csv("formatted_A1163_ML_UX_LncRNA_T_TPM_CUTOFF_6_8.fasta",␣
 ↪sep=("\t"), header=None).loc[1::2]
dfLNCR_All.columns=["info"]
dfLNCR_All=dfLNCR_All.reset_index()
```

```
dfLNCR_All["merger"]=dfLNCR_All.index
dfLNCR_All2.columns = ["seq"]
dfLNCR_All2=dfLNCR_All2.reset_index()
dfLNCR_All2["merger"]=dfLNCR_All2.index
dfLNCR_merge=dfLNCR_All.merge(dfLNCR_All2, on="merger")
dfLNCR_merge=dfLNCR_merge[["info", "seq"]]

dfLNCR_merge["classes"]=dfLNCR_merge.loc[:,"info"].str.
 ↪split("code=",expand=True)[1].str.split(";", expand=True)[0]
dfLNCR_merge["length"]=dfLNCR_merge.seq.apply(lambda x:len(x))
dfLNCR_merge["GC"]=dfLNCR_merge.seq.apply(lambda x:(x.upper().count('C'))+(x.
 ↪upper().count('G')))
dfLNCR_merge["GC_content"]=dfLNCR_merge["GC"]/dfLNCR_merge["length"]
```

[4]:
```
# Candidate LncRNAs from Weaver screen
weaver_All=pd.read_csv("WEAVER_ALL.fasta_F.fasta", sep=("\t"), header=None).
 ↪loc[::2]
weaver_All2=pd.read_csv("WEAVER_ALL.fasta_F.fasta", sep=("\t"), header=None).
 ↪loc[1::2]
weaver_All.columns=["info"]
weaver_All=weaver_All.reset_index()
weaver_All["merger"]=weaver_All.index
weaver_All2.columns = ["seq"]
weaver_All2=weaver_All2.reset_index()
weaver_All2["merger"]=weaver_All2.index
weaver_merge=weaver_All.merge(weaver_All2, on="merger")
weaver_merge=weaver_merge[["info", "seq"]]

weaver_merge["classes"]=weaver_merge.loc[:,"info"].str.
 ↪split("code=",expand=True)[1].str.split(";", expand=True)[0]
weaver_merge["length"]=weaver_merge.seq.apply(lambda x:len(x))
weaver_merge["GC"]=weaver_merge.seq.apply(lambda x:(x.upper().count('C'))+(x.
 ↪upper().count('G')))
weaver_merge["GC_content"]=weaver_merge["GC"]/weaver_merge["length"]
```

[5]:
```
#annotated mRNAs Aspergillus_fumigatusa1163.ASM15014v1.53
# make sure transposns not included
mRNA_All=pd.read_csv("/home/marian-linux/Documents/Project2/
 ↪INTERSECT_A1163_after_TPM_CUTOFF/formatted_A1163_ALL_GENOME_24_7.fasta",␣
 ↪sep=("\t"), header=None).loc[::2]
mRNA_All2=pd.read_csv("/home/marian-linux/Documents/Project2/
 ↪INTERSECT_A1163_after_TPM_CUTOFF/formatted_A1163_ALL_GENOME_24_7.fasta",␣
 ↪sep=("\t"), header=None).loc[1::2]
mRNA_All.columns=["info"]
mRNA_All=mRNA_All.reset_index()
mRNA_All["merger"]=mRNA_All.index
```

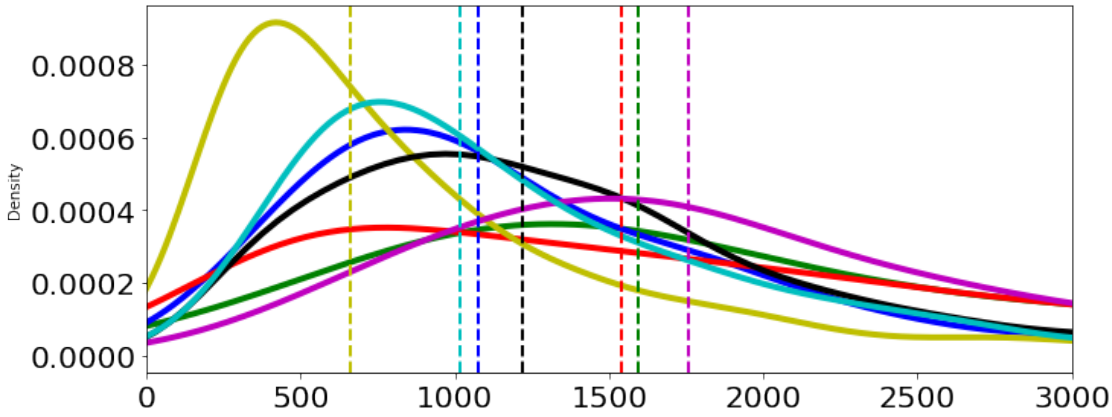
```
mRNA_All2.columns = ["seq"]
mRNA_All2=mRNA_All2.reset_index()
mRNA_All2["merger"]=mRNA_All2.index
mRNA_All.loc[:,"info"].str.split("|",expand=True)[2].str.split(",", expand=True)
mRNA_All=mRNA_All[mRNA_All.loc[:,"info"].apply(lambda x: "CDS" in x)]
mRNA_All=mRNA_All[mRNA_All.loc[:,"info"].apply(lambda x: "transposon"  not in␣
 ↪x)]
mRNA_merge=mRNA_All.merge(mRNA_All2, on="merger")
mRNA_merge=mRNA_merge[["info", "seq"]]
mRNA_merge["length"]=mRNA_merge.seq.apply(lambda x:len(x))
mRNA_merge["GC"]=mRNA_merge.seq.apply(lambda x:(x.upper().count('C'))+(x.
 ↪upper().count('G')))
mRNA_merge["GC_content"]=mRNA_merge["GC"]/mRNA_merge["length"]
```

[6]:
```
# DENSITY PLOT Length - medians highlighted
a=dfPP_merge[dfPP_merge.classes=="x"].length
b=dfPP_merge[dfPP_merge.classes=="u"].length
c=dfLNCR_merge[dfLNCR_merge.classes=="x"].length
d=dfLNCR_merge[dfLNCR_merge.classes=="u"].length
e = mRNA_merge.length
k=weaver_merge[weaver_merge.classes=="x"].length
l=weaver_merge[weaver_merge.classes=="u"].length

f = dfPP_merge[dfPP_merge.classes=="x"].length.median()
g= dfPP_merge[dfPP_merge.classes=="u"].length.median()
h = dfLNCR_merge[dfLNCR_merge.classes=="x"].length.median()
i= dfLNCR_merge[dfLNCR_merge.classes=="u"].length.median()
j= mRNA_merge.length.median()
m = weaver_merge[weaver_merge.classes=="x"].length.median()
n= weaver_merge[weaver_merge.classes=="u"].length.median()
plt.figure(figsize=(10,4))
a.plot(kind='density',color='g',linewidth=4.0)
b.plot(kind='density',color='b',linewidth=4.0)
c.plot(kind='density',color='r',linewidth=4.0)
d.plot(kind='density',color='y',linewidth=4.0)
e.plot(kind='density',color='k',linewidth=4.0)
k.plot(kind='density',color='m',linewidth=4.0)
l.plot(kind='density',color='c',linewidth=4.0)


plt.axvline(x=f, linestyle='--',color='g',linewidth=2.0)
plt.axvline(x=g, linestyle='--',color='b',linewidth=2.0)
plt.axvline(x=h, linestyle='--',color='r',linewidth=2.0)
plt.axvline(x=i, linestyle='--',color='y',linewidth=2.0)
plt.axvline(x=j, linestyle='--',color='k',linewidth=2.0)
plt.axvline(x=m, linestyle='--',color='m',linewidth=2.0)
plt.axvline(x=n, linestyle='--',color='c',linewidth=2.0)
```

```
plt.xticks(size=20)
plt.yticks( size=20)
plt.tight_layout()
plt.xlim([0, 3000])
plt.show()
```

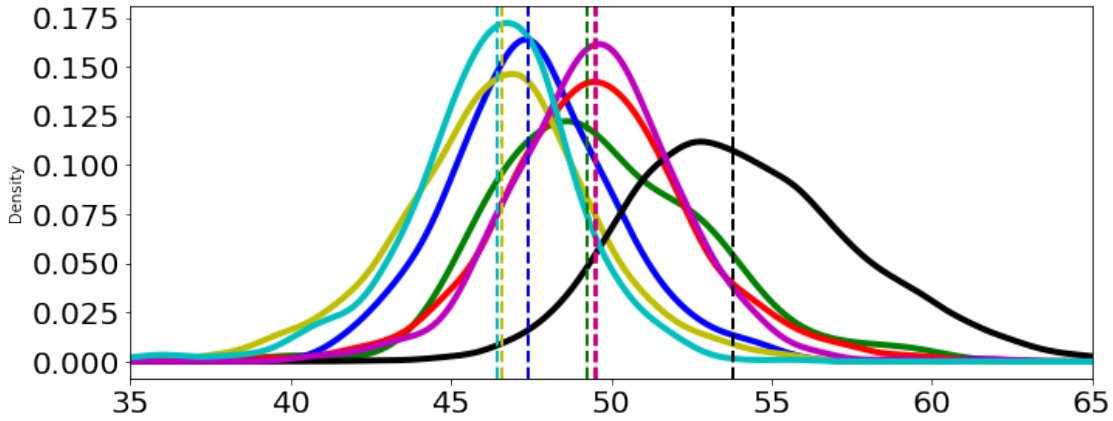


[7]:
```
# DENSITY PLOT GC-content transcripts - medians highlighted
a2=dfPP_merge[dfPP_merge.classes=="x"].GC_content*100
b2=dfPP_merge[dfPP_merge.classes=="u"].GC_content*100
c2=dfLNCR_merge[dfLNCR_merge.classes=="x"].GC_content*100
d2=dfLNCR_merge[dfLNCR_merge.classes=="u"].GC_content*100
e2 = mRNA_merge.GC_content*100
k2=weaver_merge[weaver_merge.classes=="x"].GC_content*100
l2=weaver_merge[weaver_merge.classes=="u"].GC_content*100

f2 = dfPP_merge[dfPP_merge.classes=="x"].GC_content.median()*100
g2= dfPP_merge[dfPP_merge.classes=="u"].GC_content.median()*100
h2 = dfLNCR_merge[dfLNCR_merge.classes=="x"].GC_content.median()*100
i2= dfLNCR_merge[dfLNCR_merge.classes=="u"].GC_content.median()*100
j2= mRNA_merge.GC_content.median()*100
m2 = weaver_merge[weaver_merge.classes=="x"].GC_content.median()*100
n2= weaver_merge[weaver_merge.classes=="u"].GC_content.median()*100

plt.figure(figsize=(10,4))
a2.plot(kind='density',color='g',linewidth=4.0)
b2.plot(kind='density',color='b',linewidth=4.0)
c2.plot(kind='density',color='r',linewidth=4.0)
d2.plot(kind='density',color='y',linewidth=4.0)
e2.plot(kind='density',color='k',linewidth=4.0)
k2.plot(kind='density',color='m',linewidth=4.0)
l2.plot(kind='density',color='c',linewidth=4.0)
```

```
plt.axvline(x=f2, linestyle='--',color='g',linewidth=2.0)
plt.axvline(x=g2, linestyle='--',color='b',linewidth=2.0)
plt.axvline(x=h2, linestyle='--',color='r',linewidth=2.0)
plt.axvline(x=i2, linestyle='--',color='y',linewidth=2.0)
plt.axvline(x=j2, linestyle='--',color='k',linewidth=2.0)
plt.axvline(x=m2, linestyle='--',color='m',linewidth=2.0)
plt.axvline(x=n2, linestyle='--',color='c',linewidth=2.0)
plt.xticks(size=20)
plt.yticks( size=20)
plt.tight_layout()
plt.xlim([35, 65])
plt.show()
```



[8]:
```
# Example stats tests
LNC_U=dfLNCR_merge[dfLNCR_merge.classes=="u"].length.tolist()
LNC_X=dfLNCR_merge[dfLNCR_merge.classes=="x"].length.tolist()
# test form noraml distribution
scipy.stats.normaltest(LNC_U)
```

[8]:
```
NormaltestResult(statistic=1219.6263854578292, pvalue=1.45042543160757e-265)
```

[9]:
```
mannwhitneyu(LNC_U,LNC_X)
```

[9]:
```
MannwhitneyuResult(statistic=682059.5, pvalue=3.2340739808513053e-108)
```