生信编程直播课程优秀学员作业展示2

Original x2yline 生信技能树 2017-03-17 12:06

收录于合集
#学徒作业

117个

题目：hg19基因组序列的一些探究

学员：x2yline

具体题目详情请参考生信技能树论坛

数据来源： http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz 下载.gz数据后解压

R语言实现（太卡，高能报警）

代码地址：
https://raw.githubusercontent.com/x2yline/coursera*notes/master/my*script/class2/fasta*file*GC*and*N.R

代码内容:

```
setwd('E:\\r\\biotrainee_demo\\class 2')
# 读入数据
t1 <- Sys.time()
df <- read.csv('chr1.fa', header=F, stringsAsFactors=F)
# index_df 为chr所在的位置
index_df <- data.frame(begin=which(sapply(df[,1], function(x){
  substr(x, start=1, stop=1)=='>'})))
# index_df1 为string所在的位置+1
index_df1 <- data.frame(rbind(matrix(index_df[-1,1]),dim(df)[1]+1))
# 把index_start和index_end存入data.frame
index_df2 <- cbind(index_df, index_df1)
remove(index_df1, index_df)
# 得出每个染色体对应string后计算其N与GC百分比
result <- apply(index_df2, 1, function(x) {  # 把提取字符串后把字符串变为大写
  y <- toupper(paste(df[(x[1]+1):(x[2]-1),1], collapse=''))
  y <- strsplit(y, split=character(0))[[1]]
  N <- length(y[y =='N'])/length(y)
  GC <- length(y[y =='G' | y == 'C'])/(length(y)-length(y[y =='N']))
  c(N,GC)
})
# 把行名改为N和GC并转秩
rownames(result) = c('N','GC')
result <- t(result)
# 取结果前几行
head(result)
```

```
26. difftime(Sys.time(), t1, units = 'secs')
```

由于电脑问题，试了一下1号染色体，电脑卡住了，于是又试了一下Y染色体，跑出来结果如下：

N	N ratio	GC	GC ratio	all base num
33720000	0.5679295	10252459	0.3996504	59[illegible]

耗时：41.44945 secs

电脑配置信息：

- R version 3.3.2 (2016-10-31)
- Platform: x86_64-w64-mingw32/x64 (64-bit)
- Running under: Windows 7 x64 (build 7601) Service Pack 1

---

python3第一种实现方法（运行速度较快，但没有3快）

数据来源： http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz

数据下载时间：2017-01-10 23:08

运行消耗时间：309 seconds

未优化速度的代码如下

```
1. import os
2. import time
3. begin = time.time()
4. os.chdir(r'F:\tmp\chromFa')
5. def count_n_and_gc(file):
6.     content = []
7.     chromsome = []
8.     g = 0; c = 0; n = 0; a = 0; t = 0
9.     with open(file) as f:
10.         raw_list = f.readlines()
11.         for i in raw_list:
12.             if not i.startswith('>'):
13.                 i = i.upper()
14.                 n +=  i.count('N')
15.                 g += i.count('G')
16.                 c += i.count('C')
17.                 a += i.count('A')
18.                 t += i.count('T')
19.             else:
20.                 if  chromsome:
21.                     content.append((n ,a, t, c, g))
22.                     g = 0; c = 0; n = 0; a = 0; t = 0
23.                 chromsome.append(i.strip())
24.         content.append(( n ,a, t, c, g))
25.     return (content,chromsome)
```

```
content = []
chromsome = []
for i in (list(range(1,23)) + ['X','Y']):
    file = 'chr'+ str(i) + '.fa'
    print('Start dealing with ' + file)
    m, n = count_n_and_gc(file)
    content += m
    chromsome += n
all_info = 'chr,GC_ratio,N_ratio,Length,N,A,T,C,G'
for i in range(len(chromsome)):
    data = '\n'+str(chromsome[i]) +',' + "%.5f"%((content[i][-1]+content[i][-2])/sum(content[i][1:])) +',
    all_info += data
with open('hg19_analysis.csv','w') as f:
    f.write(all_info)
print('Time using:'+ str(time.time() - begin) + ' seconds\n')
```

---

shell +python3（最快）

先使用shell脚本把所有chromFa.tar.gz 中的所有.fa文件合并为一个hg19.fa文件

脚本如下：

```
tar zvfx chromFa.tar.gz
cat *.fa > hg19.fa
rm chr*.fa
less hg19.fa
```

按照老师的方法对python算法进行改良

改良后的代码如下：

代码地址：

```
import os
import time
import re
import sys
from collections import OrderedDict
start = time.clock()
def count_fasta_atcgn(file_path, buffer_size=1024*1024):
    bases = ['N', 'A', 'T', 'C', 'G']
    ATCG_analysis = OrderedDict()
    with open(file_path, 'r') as f:
        line1 = f.readline()
        chr_i = re.split('\s', line1)[0][1:]
        print(chr_i)
        ATCG_analysis[chr_i] = OrderedDict()
        for base in bases:
            ATCG_analysis[chr_i][base] = 0
        while True:
            chunk = f.read(buffer_size).upper()
            if '>' in chunk:
                chromsome = re.split('>',chunk)
                if chromsome[0]:
                    for base in bases:
                        ATCG_analysis[chr_i][base] += chromsome[0].count(base)
                for i in chromsome[1:]:
                    if i:
                        chr_i = re.split('\s', i[0:i.index('\n')])[0]
                        print(chr_i)
                        strings_i = i[i.index('\n'):].upper()
```

```
                        strings_i = i[i.index('\n'):].upper()
                        ATCG_analysis[chr_i] = OrderedDict()
                        for base in bases:
                            ATCG_analysis[chr_i][base] = strings_i.count(base)
            else:
                for base in bases:
                    ATCG_analysis[chr_i][base] += chunk.count(base)
            if not chunk:
                break
    return ATCG_analysis

def write_atcg_to_csv(ATCG_analysis, file_path = '.'):
    file = os.path.join(file_path,'atcg_analysis.csv')
    csv_content = 'chromsome\tGC_content\tN_content\tLength\tN\tA\tT\tC\tG\n'
    for chr_id, atcg_count in ATCG_analysis.items():
        GC = atcg_count['G'] + atcg_count['C']
        N = atcg_count['N']
        Length = sum(atcg_count.values())
        GC_content = GC*1.0/(Length-N)
        N_content = N*1.0/Length
        csv_content += chr_id + '\t' + '%.4f'%GC_content + '\t' + '%.4f'%N_content + '\t' + str(Length) +
    with open(file, 'w') as f:
        csv_file_content = re.sub('\t', ',', csv_content)
        f.write(csv_file_content)
    print(u'File have been saved in '+ file)
    return csv_content


if sys.argv:
    result = OrderedDict()
    for f in sys.argv:
        done = 0
        f= f.strip('''"''')
        if f.count('.') != 1 or f[-2:] == 'py' or not os.path.exists(f):
            continue
        print(f)

        try:
            done = 1
            result = OrderedDict(count_fasta_atcgn(file_path = f, buffer_size = 1024*2048), **result)
        except Exception as e:
            if f.startswith('-'):
                pass
            else:
                print(type(e))

    if done == 1:
        file = write_atcg_to_csv(result)
        print(file)
        print('used %.2f s'%(time.clock()-start))
    else:
        print ('\n\nSorry! The command is invalid!\n')
else:
    directory = input('Enter your file: ')
    start = time.clock()
    if directory.count('.') != 1 or directory[-2:] == 'py' or not os.path.exists(directory):
        print('Your file is invalid!')
    else:
        result = count_fasta_atcgn(file_path = directory, buffer_size = 1024*2048)
        file = write_atcg_to_csv(result)
        print('used %.2f s'%(time.clock()-start))
```

保存上述代码为 `fasta_atcgn_summary.py` 文件后

在命令行下输入：

```
python fasta_atcgn_summary.py   F:\tmp\hg19.fa
```

部分输出结果如下

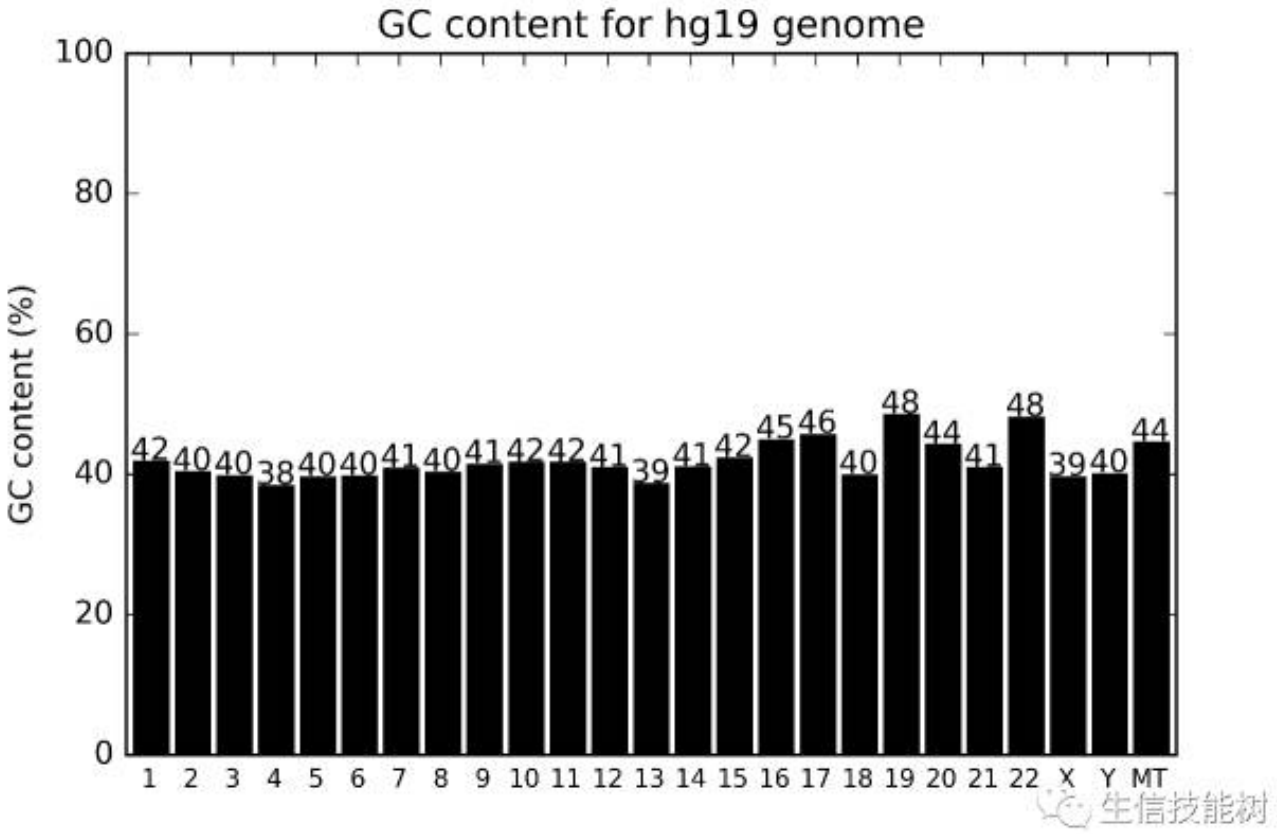
chromsome	GC_content	N_content	Length	N	A	T	C	G
CHRUN_GL000237	0.4666	0	45867	0	12273	12191	10241	11162
CHR14	0.4089	0.1776	107349540	19060000	25992966	26197495	18027132	1
8071947								
CHR9_GL000199_RANDOM	0.3791	0	169874	0	54702	50765	34981	29426
CHR2	0.4024	0.0205	243199373	4994855	71102632	71239379	47915465	479
47042								
CHR8_GL000197_RANDOM	0.5401	0.0027	37175	100	8644	8408	9883	10140
CHRUN_GL000247	0.436	0	36422	0	11002	9540	7794	8086
CHR19	0.4836	0.0561	59128983	3320000	14390632	14428951	13478255	135

使用python进一步进行可视化处理

代码如下：

```python
import os
import time
import re
import sys
from collections import OrderedDict
start = time.clock()
def count_fasta_atcgn(file_path, buffer_size=1024*1024):
    bases = ['N', 'A', 'T', 'C', 'G']
    ATCG_analysis = OrderedDict()
    with open(file_path, 'r') as f:
        line1 = f.readline().upper()
        chr_i = re.split('\s', line1)[0][1:]
        print(chr_i)
        ATCG_analysis[chr_i] = OrderedDict()
        for base in bases:
            ATCG_analysis[chr_i][base] = 0
        while True:
            chunk = f.read(buffer_size).upper()
            if '>' in chunk:
                chromsome = re.split('>',chunk)
                if chromsome[0]:
                    for base in bases:
                        ATCG_analysis[chr_i][base] += chromsome[0].count(base)
                for i in chromsome[1:]:
                    if i:
                        chr_i = re.split('\s', i[0:i.index('\n')])[0]
                        print(chr_i)
                        strings_i = i[i.index('\n'):]
                        ATCG_analysis[chr_i] = OrderedDict()
                        for base in bases:
                            ATCG_analysis[chr_i][base] = strings_i.count(base)
            else:
                for base in bases:
                    ATCG_analysis[chr_i][base] += chunk.count(base)
            if not chunk:
                break
    return ATCG_analysis

def write_atcg_to_csv(ATCG_analysis, file_path = '.'):
    file = os.path.join(file_path,'atcg_analysis.csv')

    csv_content = 'chromsome\tGC_content\tN_content\tLength\tN\tA\tT\tC\tG\n'
    for chr_id, atcg_count in ATCG_analysis.items():
        GC = atcg_count['G'] + atcg_count['C']
        N = atcg_count['N']
        Length = sum(atcg_count.values())
        GC_content = GC*1.0/(Length-N)
        N_content = N*1.0/Length
        csv_content += chr_id + '\t' + '%.4f'%GC_content + '\t' + '%.4f'%N_content + '\t' + str(Length) +
    with open(file, 'w') as f:
        csv_file_content = re.sub('\t', ',', csv_content)
        f.write(csv_file_content)
```

```
    print(u'File have been saved in '+ file)
    return csv_content

file_path = 'F:\genome\chromFa\hg19.fa'

ATCG_analysis = count_fasta_atcgn(file_path, buffer_size=1024*1024)
cg_list = []
chr_id_list = list(range(1,23)) + ['X','Y','M']
for i in chr_id_list:
    cg_list.append((ATCG_analysis['CHR'+str(i)]['G']+ATCG_analysis['CHR'+str(i)]['C'])/(ATCG_analysis['CH
import matplotlib.pyplot as plt
plt.bar(left = range(25), height = cg_list, color='k')
for i in range(len(cg_list)):
    plt.text( x=i-0.1, y=cg_list[i]+.35,s=str(round(cg_list[i])))
plt.title('GC content for hg19 genome')
plt.ylabel('GC content (%)')
pos = []
for i in range(len(chr_id_list)):
    pos.append(i + 0.35)
plt.xticks(pos, list(range(1,23)) + ['X','Y','MT'], fontsize=8)
plt.xlim(-0.2, )
plt.ylim(0, 100)
plt.savefig('F:\hg19_gc.png',dpi=600)
plt.show()
```



本文编辑：思考问题的熊

收录于合集 #学徒作业 117

上一篇
给学徒的GEO作业

下一篇
生信编程直播课程优秀学员学习心得及作业展示3

People who liked this content also liked

EDCI, HATU, CDI等酸胺缩合试剂副产物和中间态结构和MS分析
有机合成路线

TLC点板，产物极性忽大忽小，咋回事！
有机合成路线

人肺内皮细胞整合单细胞图谱
生信菜鸟团