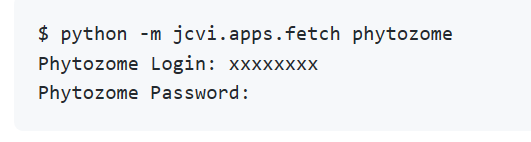
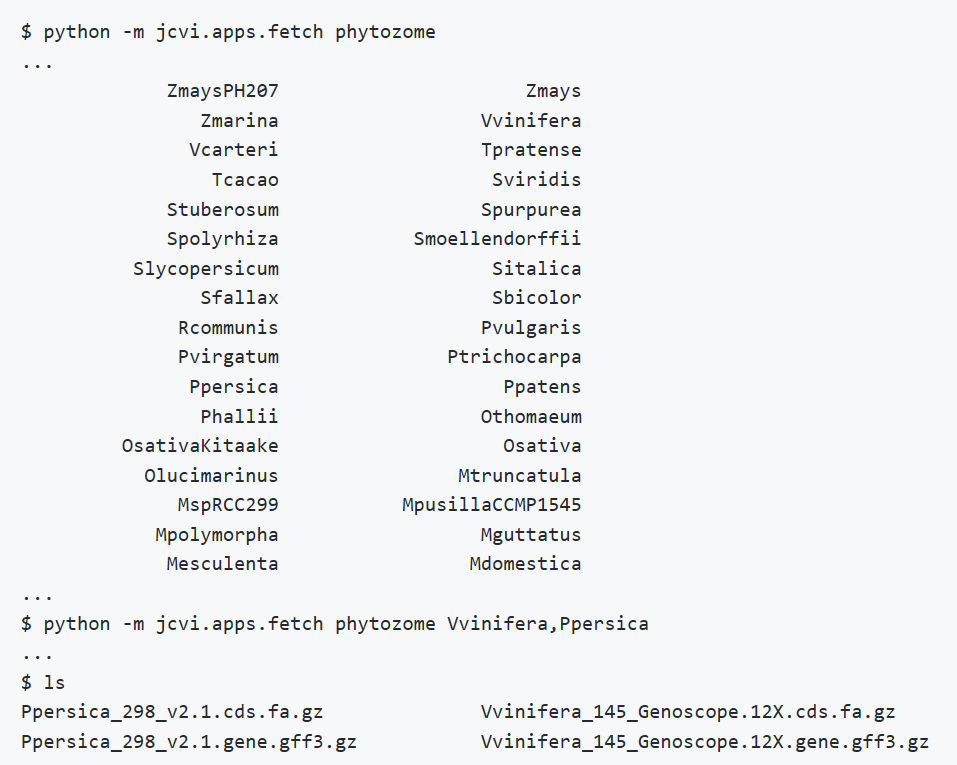
**一、用MCScanX和TBtools画物种内共线性图：**

1. **gff文件和fa文件下载**（phytozome、NCBI、ensemble等数据库），可以网上下载后上传到集群，也可以直接在集群上下载：

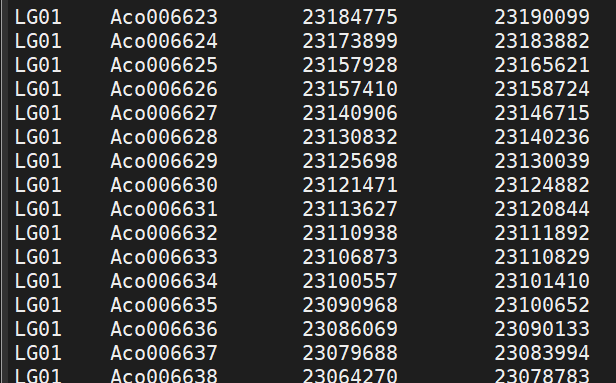




图来源于：https://github.com/tanghaibao/jcvi/wiki/MCscan-(Python-version)

2、**gff文件修改**（MCScanX识别的gff文件只有四列，分别是染色体ID, 基因ID, 起始和终止）：

grep '\bgene\b' **.Acomosus\_321\_v3.gene.gff3** | awk '{print $1"\t"$9"\t"$4"\t"$5}' | sed 's/ID=.\*;Name=//g' > **at\_vv.gff**  (这里的命名用了之前的命名，做的时候忘记改了….)



3、**fa文件修改**：

cut -d '.' -f 1 **Acomosus\_321\_v3.protein.fa** > **at\_vv.fa**

4、**核对gff文件和fa文件是否一致，无结果输出就是一致**：

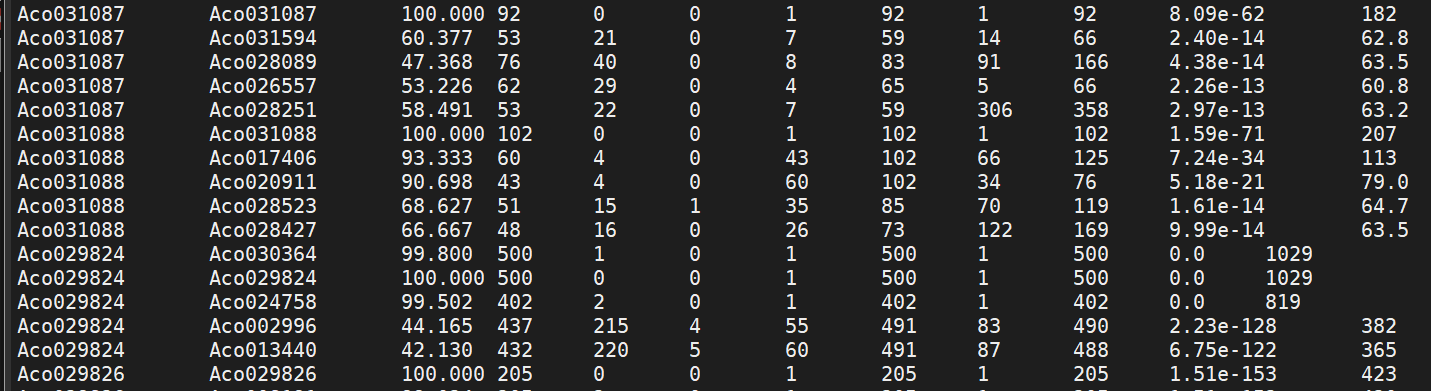
cut -f 2 at\_vv.gff > lst1; grep '>' at\_vv.fa | sed 's/>//g' > lst2; cat lst1 lst2 | sort | uniq -c | sed 's/^ \*//g' | grep -v '^2'

5、**利用fa文件建库**：

makeblastdb -in **at\_vv.fa** -dbtype prot -parse\_seqids -out **all**

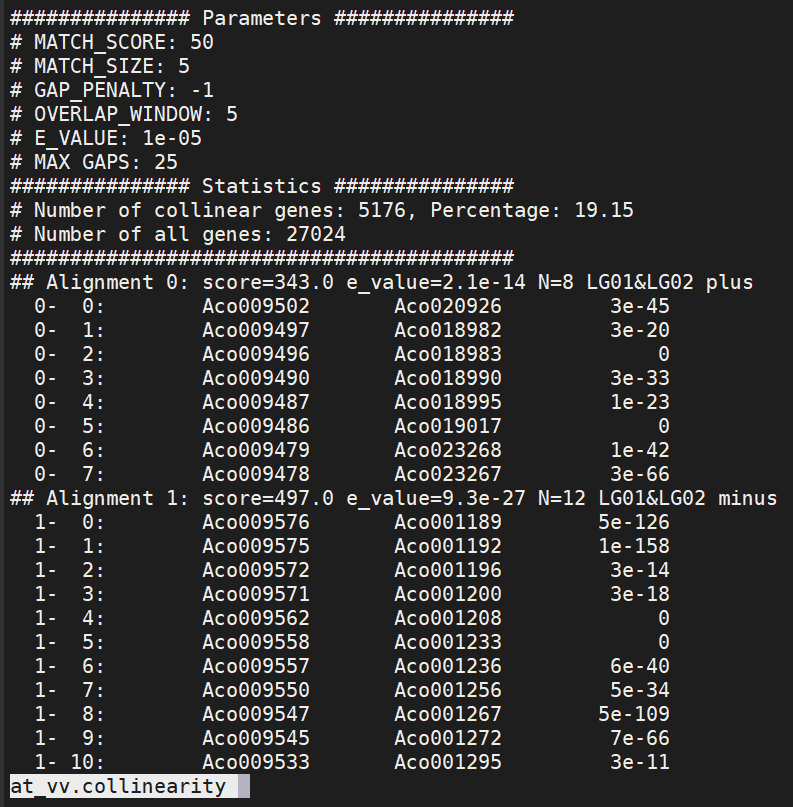
6、**blast比对**，输出文件为at\_vv.blast：

blastp -query **at\_vv.fa** -db **all** -out **at\_vv.blast** -evalue 1e-10 -num\_threads 6 -outfmt 6 -num\_alignments 5

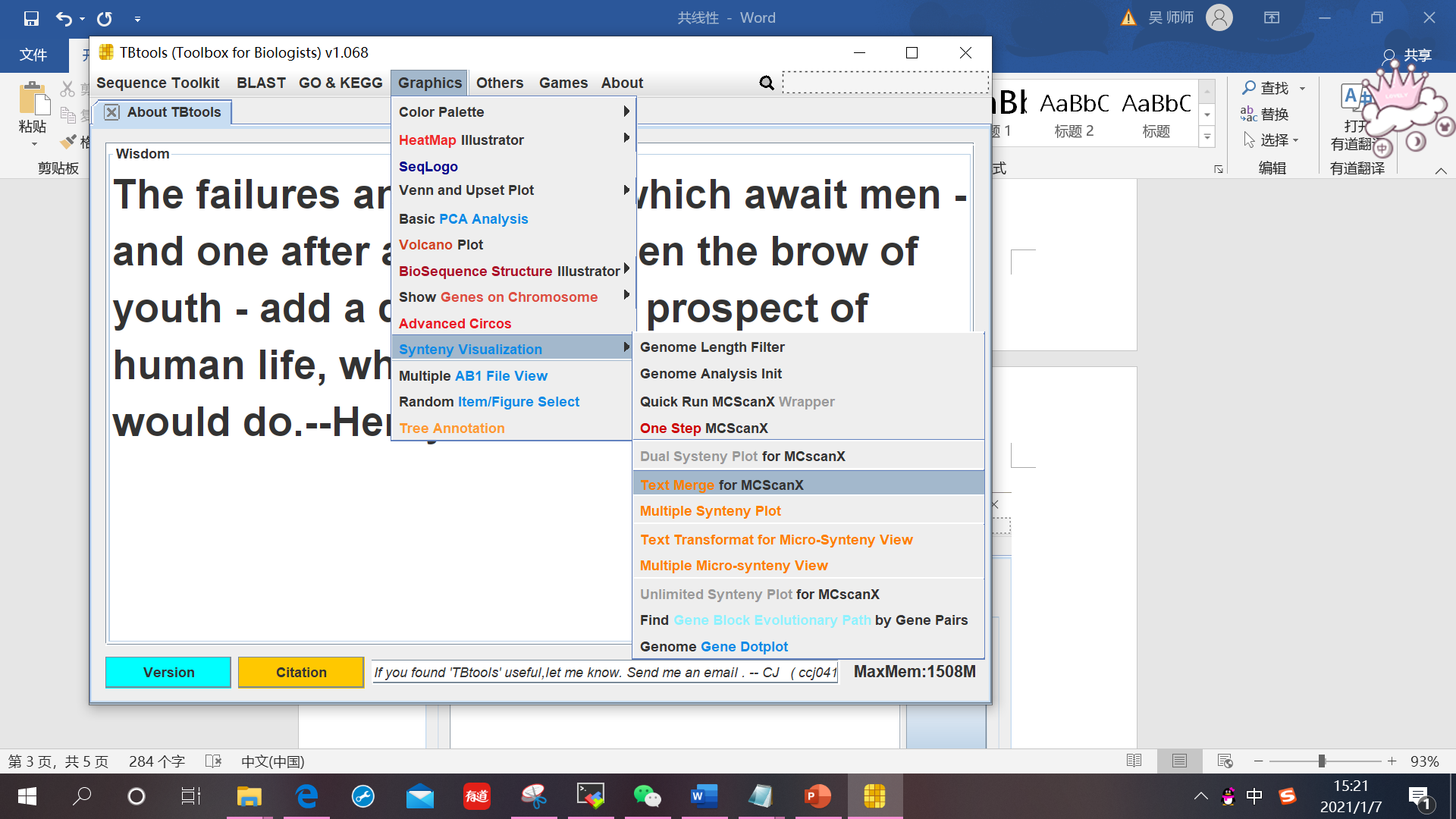


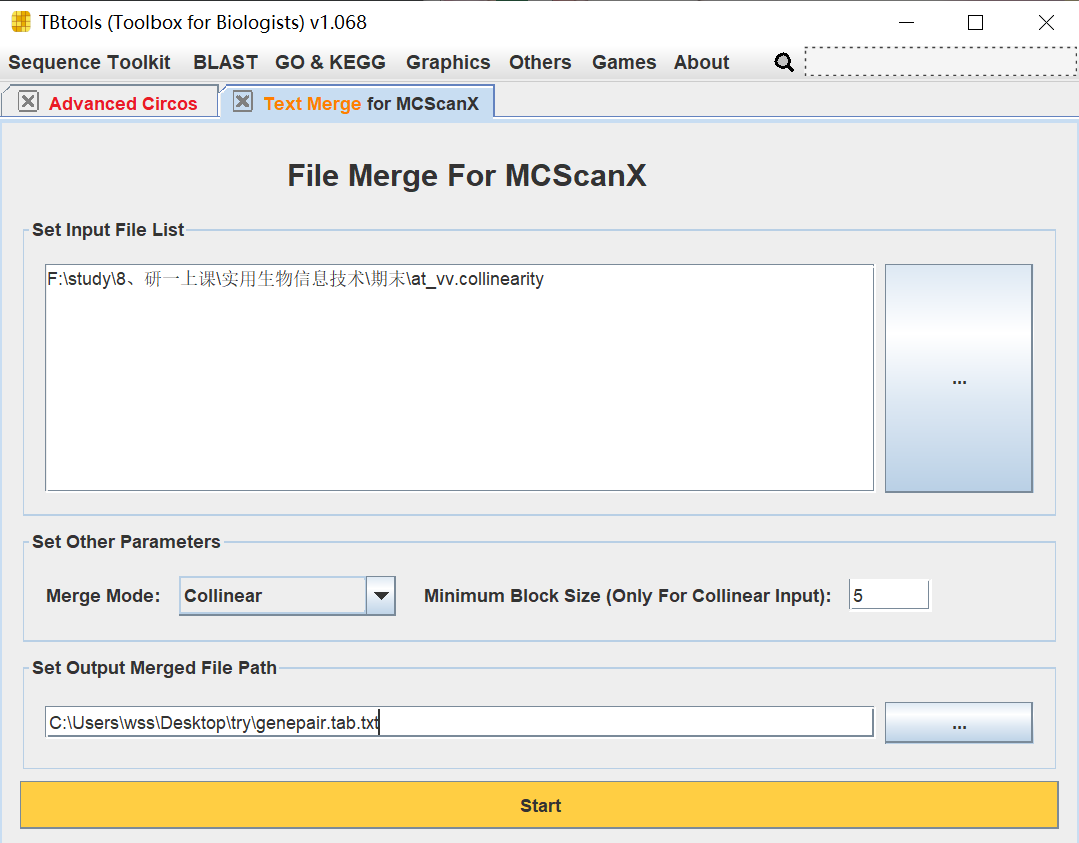
7、**共线性分析**（需要blast文件和修改后的gff文件，二者名字需相同）：

MCScanX at\_vv

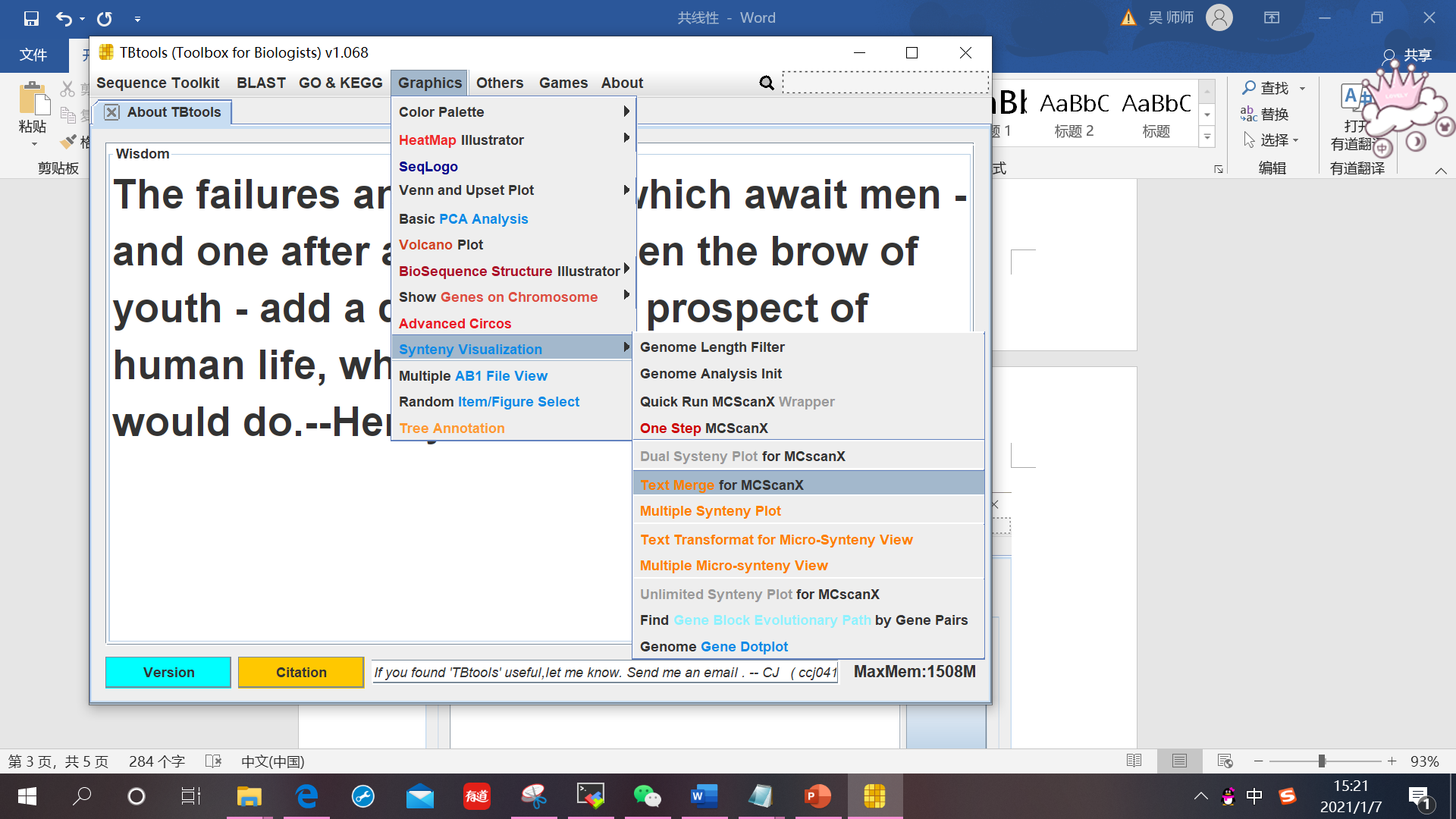


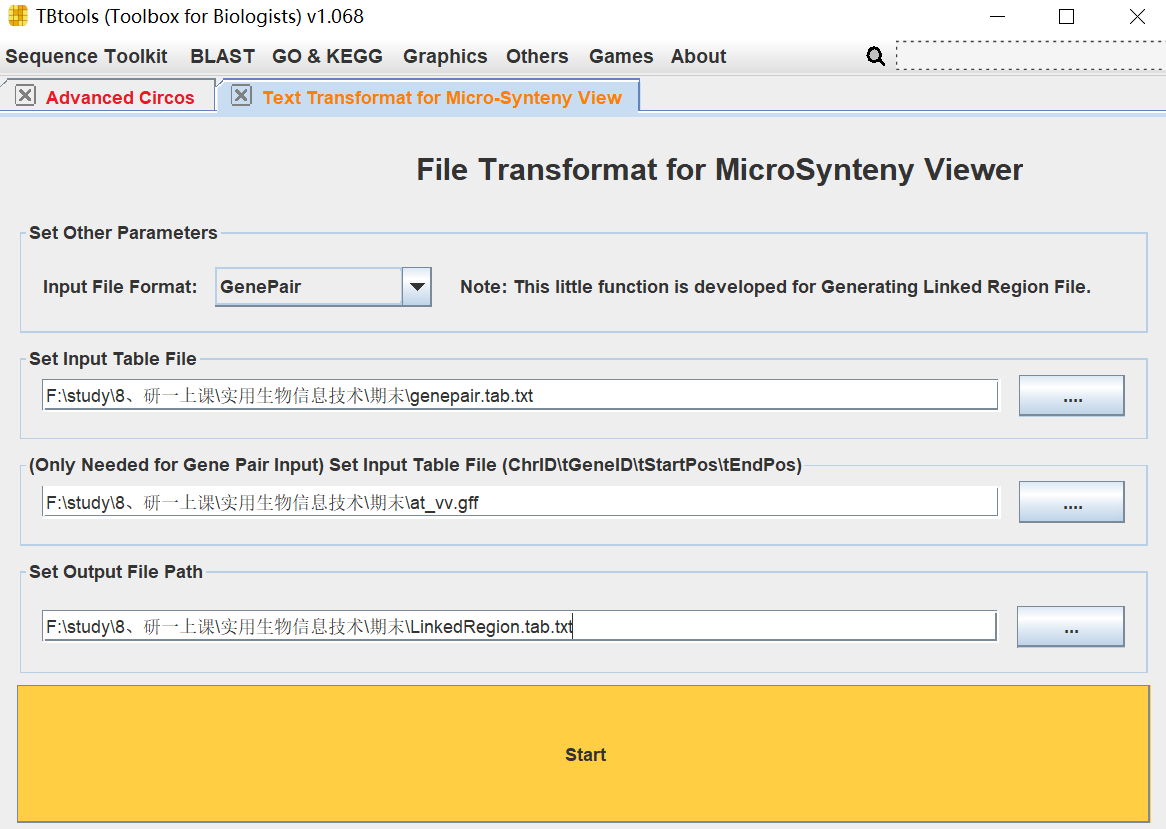
8、**用TBtools进行可视化**



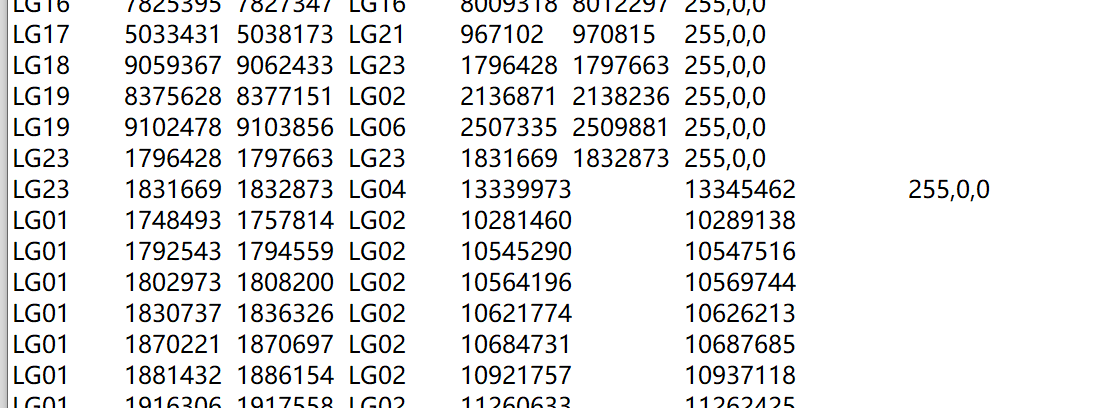


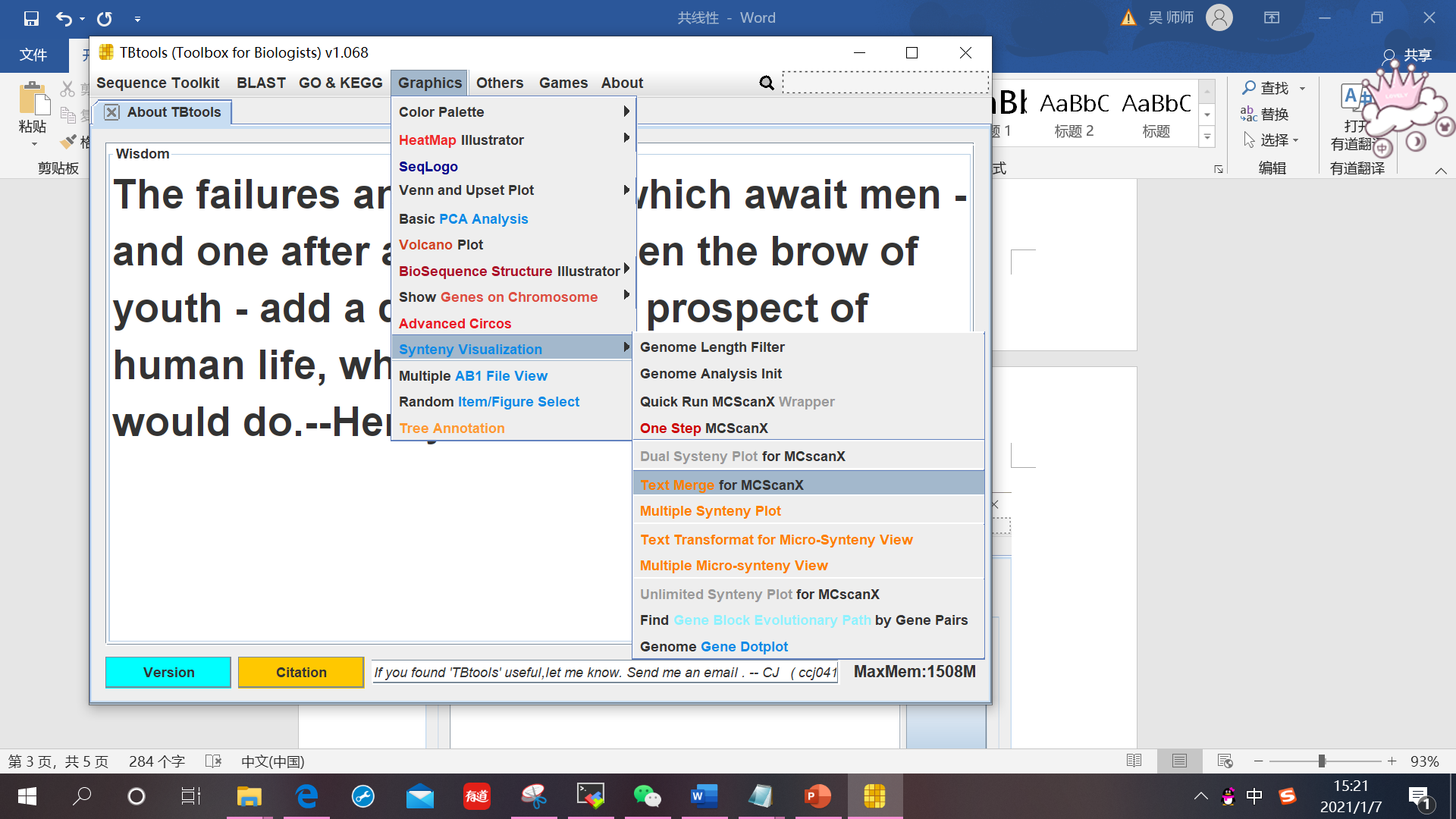
.collinearity是上一步共线性分析得到的结果文件

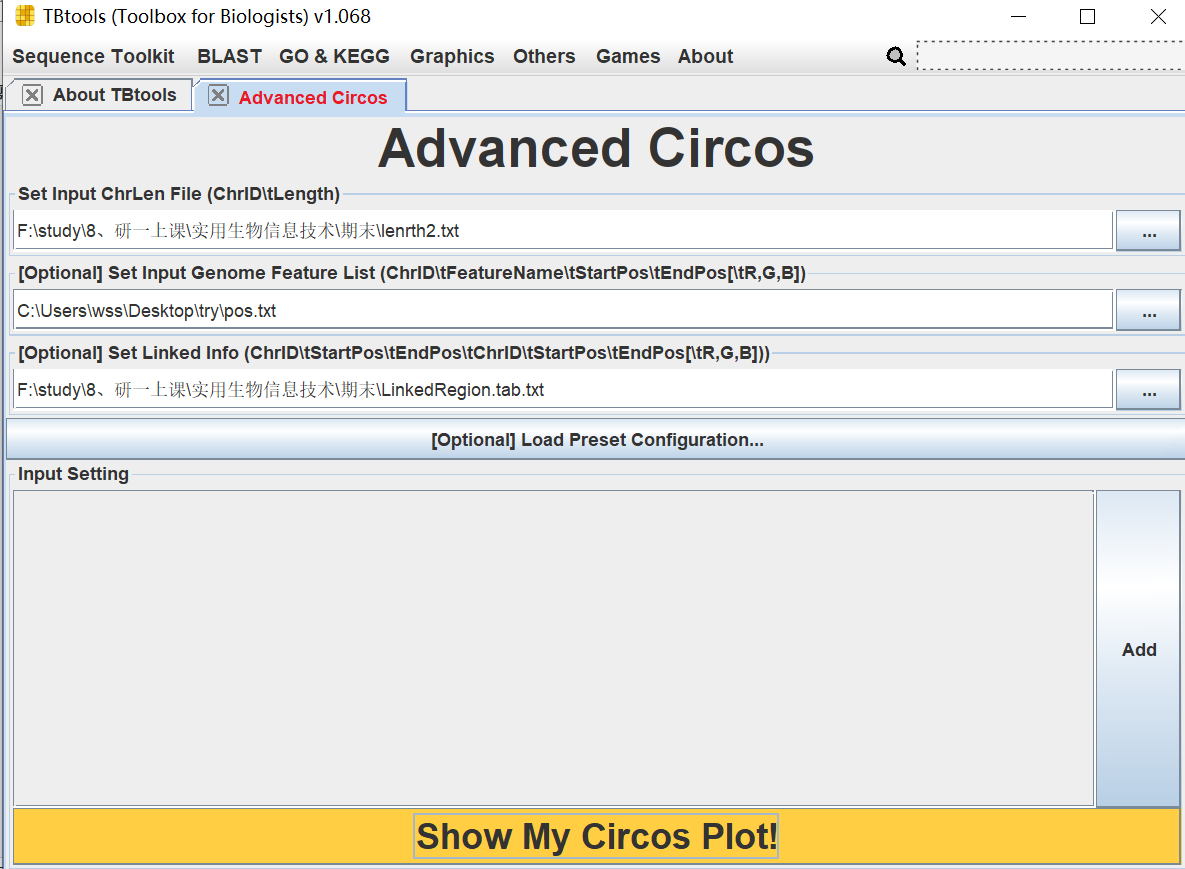




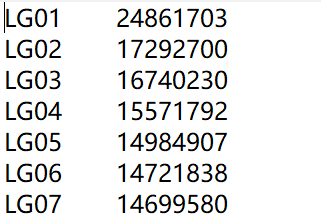
如果仅用这步生成的LinkedRegin.tab.txt，共线性图中发生片段复制的genepair无法标红，如需要标红，需对LinkedRegion.tab.txt文件进行修改，我的修改方式是：下载文献中提供的发生片段复制的genepair的文件，根据genepair文件在gff文件中提取出染色体号、基因起始和终止位点，最后一列加上颜色（255,0,0），与原LinkedRegion.tab.txt合并



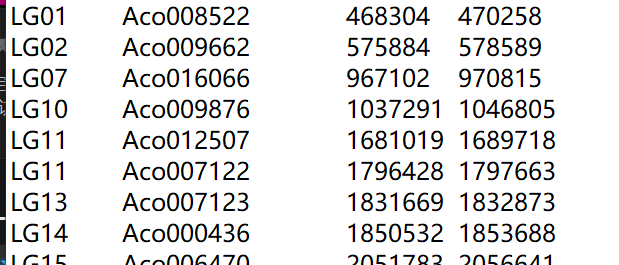


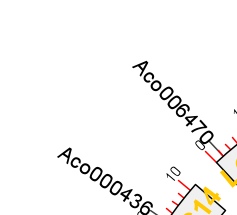


lenrth.txt (没错这个文件名是因为当时打错字了……这个是骨架文件，我是通过gff文件计算每条染色体的长度获得，菠萝的数据是一共25个染色体，截图是部分数据，这步应该有代码实现的方法，但是我还不会，于是用的excel……：)

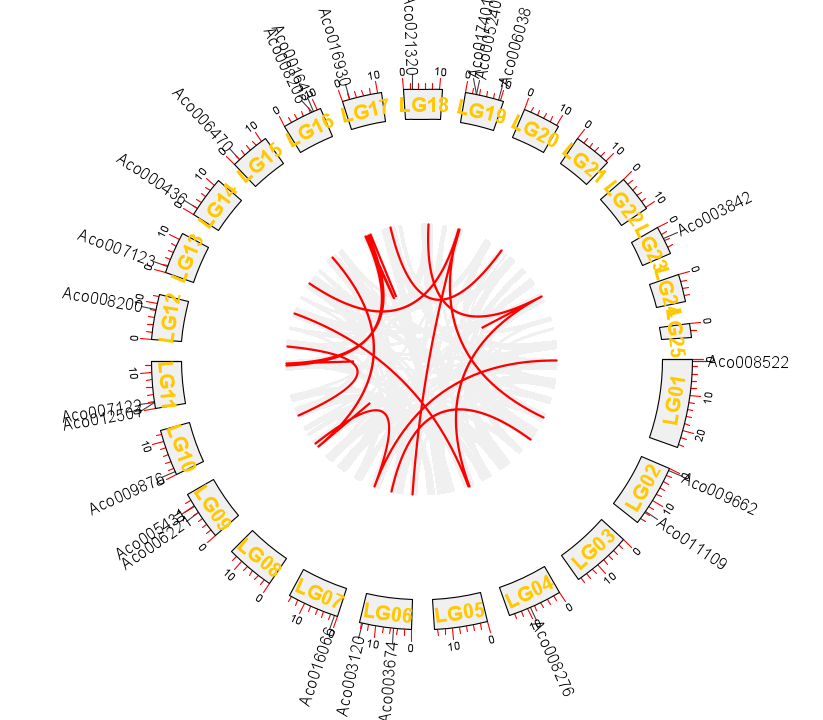


pos.txt是根据发生片段复制的genepair名单（文献附件提供），在gff文件里获得每个基因的染色体号，起始和终止位点，这个文件让共线性图外周显示基因ID，这个表格要能和前面的LinkedRegion.tab.txt中需要标红线的gene对应得上





上述步骤后得到图：



根据图上方的里的内容和鼠标拖拽美化共线性图，如：



**二、利用JCVI画物种间共线性图：**（参考了：https://github.com/tanghaibao/jcvi/wiki/MCscan-(Python-version)）

1. **gff和fa文件修改和格式转换为bed和pep**（就是先把gff变成bed，再根据bed去提取fa为pep）
   1. 提取最长转录本（不提取应该也能作图，但是自己没试过）：

1）先根据GFF文件提取出bed文件：

less -SN Triticum\_dicoccoides.WEWSeq\_v.1.0.48.gff3|awk '$3=="mRNA"{print}'|cut -f 1,4,5,7,9|cut -d ';' -f 1|sed s/ID=transcript://g|awk '{print $1"\t"$2"\t"$3"\t"$5"\t0\t"$4}' > Triticum\_dicoccoides.bed

2）利用jcvi.formats.bed去重，剩余最长转录本的bed文件：

python -m jcvi.formats.bed uniq Triticum\_dicoccoides.bed

（这种方法会抛弃一些序列，因为考虑到gene间的重叠，所以可以编程提取最长转录本的bed文件，此方法来源于课题组师兄：extract\_longest\_pep\_from\_bed2bed.py

#!/usr/bin/python3

# -\*- coding: UTF-8 -\*-

dict\_pep = {}

with open('C:/Users/chaos/Desktop/AABBDD.bed','r') as f,open('C:/Users/chaos/Desktop/AABBDD.uniq.lst','w') as w:

for transcript in f:

seq\_id = []

seq\_pep = []

length = int(transcript.split('\t')[2])-int(transcript.split('\t')[1])

pep = transcript.split('\t')[3]

seq\_pep.append(pep)

seq\_pep.append(length)

seq\_id.append(seq\_pep)

id = transcript.split('\t')[3].split('.')[0]

if id not in dict\_pep:

dict\_pep[id] = seq\_id

else:

dict\_pep[id] += seq\_id

for k in dict\_pep.keys():

if len(dict\_pep[k]) == 1:

w.write(dict\_pep[k][0][0]+'\n')

else:

max\_num = 0

for v in dict\_pep[k]:

print(v)

if v[1] >= max\_num:

max\_num = v[1]

longest\_pep = v[0]

w.write(longest\_pep + '\n')

for i in $(cat AABBDD.longest\_pep.lst);do grep $i AABBDD.bed >> AABBDD.longest\_pep.bed;done ##根据lst匹配出bed文件；除此之外，参考网站上提供的方法为：python -m jcvi.formats.gff bed --type=mRNA --key=Name Vvinifera\_145\_Genoscope.12X.gene.gff3.gz -o grape.bed 不过这里的bed文件应该不仅仅是最长转录本，文件大小也更大一些）

* 1. 根据bed文件提取蛋白序列：（seqkit）（和一、中的一样，也是要保证两个文件的一致）

seqkit grep -f <(cut -f 4 ath.uniq.bed ) Arabidopsis\_thaliana.TAIR10.pep.all.fa.gz | seqkit seq -i > ath.pep（提取pep）

* 1. 重命名xxx.uniq.bed 文件为 .bed文件

1. **寻找共线性块**（可以新建一个文件夹把**Acomosusv.bed Acomosusv.pep Athaliana.bed Athaliana.pep**放进去）

python -m jcvi.compara.catalog ortholog --dbtype prot --no\_strip\_names **Acomosusv Athaliana**

1. **建立seqids文件和layout文件**（里面参数需要修改）

①seqids文件如下：（分别是两个物种的染色体编号，注意不要有多余的空行，当初就因为有空行报错了……）

LG01,LG02,LG03,LG04,LG05,LG06,LG07,LG08,LG09,LG10,LG11,LG12,LG13,LG14,LG15,LG16,LG17,LG18,LG19,LG20,LG21,LG22,LG23,LG24,LG25

Chr1,Chr2,Chr3,Chr4,Chr5

②layout文件如下（需修改）：

#y, xstart, xend, rotation, color, label, va, bed

.6, .1, .8, 0, , Acomosus, top, Acomosus.bed

.4, .1, .8, 0, , Athaliana, top, Athaliana.bed

# edges

e, 0, 1, Acomosus.Athaliana.anchors.simple

（layout文件参数意义：

y：染色体的水平位置

xstart：画图起始位置

rotation：染色体旋转角度

color：染色体颜色

label：染色体标签

va: 染色体标签的位置

bed:画图输入的两个基因组的bed文件名

edges:表示染色体谁与谁连起来（0代表第一个，即grape；1代表第二个，即peach，以此类推））

1. **可视化：**

python -m jcvi.graphics.karyotype seqids layout