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：

```
$ python -m jcvi.apps.fetch phytozome
Phytozome Login: xxxxxxxx
Phytozome Password:
```

```
$ python -m jcvi.apps.fetch phytozome
...
             ZmaysPH207                      Zmays
                Zmarina                   Vvinifera
               Vcarteri                   Tpratense
                 Tcacao                    Sviridis
              Stuberosum                  Spurpurea
              Spolyrhiza              Smoellendorffii
           Slycopersicum                     Sitalica
                 Sfallax                    Sbicolor
               Rcommunis                   Pvulgaris
               Pvirgatum                 Ptrichocarpa
                Ppersica                      Ppatens
                 Phallii                   Othomaeum
          OsativaKitaake                     Osativa
            Olucimarinus                 Mtruncatula
               MspRCC299             MpusillaCCMP1545
             Mpolymorpha                    Mguttatus
              Mesculenta                   Mdomestica
...
$ python -m jcvi.apps.fetch phytozome Vvinifera,Ppersica
...
$ ls
Ppersica_298_v2.1.cds.fa.gz              Vvinifera_145_Genoscope.12X.cds.fa.gz
Ppersica_298_v2.1.gene.gff3.gz           Vvinifera_145_Genoscope.12X.gene.gff3.gz
```

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…)

LG01	Aco006623	23184775	23190099
LG01	Aco006624	23173899	23183882
LG01	Aco006625	23157928	23165621
LG01	Aco006626	23157410	23158724
LG01	Aco006627	23140906	23146715
LG01	Aco006628	23130832	23140236
LG01	Aco006629	23125698	23130039
LG01	Aco006630	23121471	23124882
LG01	Aco006631	23113627	23120844
LG01	Aco006632	23110938	23111892
LG01	Aco006633	23106873	23110829
LG01	Aco006634	23100557	23101410
LG01	Aco006635	23090968	23100652
LG01	Aco006636	23086069	23090133
LG01	Aco006637	23079688	23083994
LG01	Aco006638	23064270	23078783

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
```

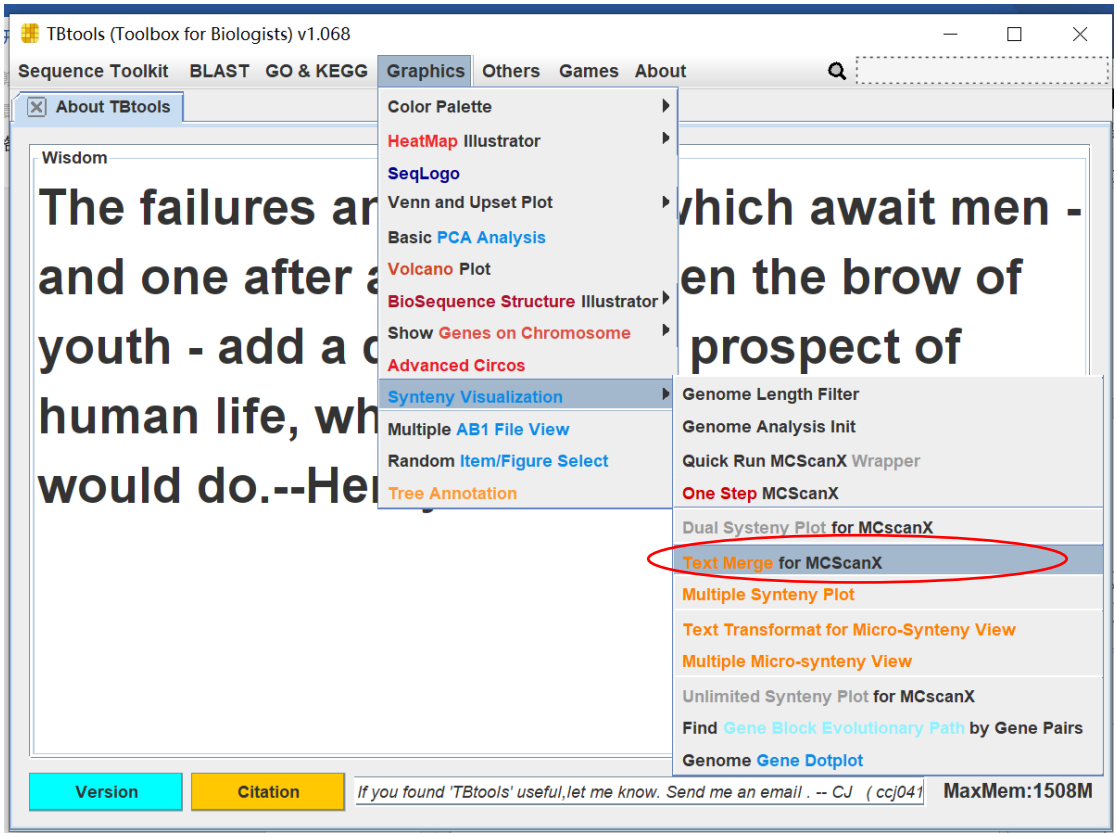
Aco031087	Aco031087	100.000	92	0	0	1	92	1	92	8.09e-62	182
Aco031087	Aco031594	60.377	53	21	0	7	59	14	66	2.40e-14	62.8
Aco031087	Aco028089	47.368	76	40	0	8	83	91	166	4.38e-14	63.5
Aco031087	Aco026557	53.226	62	29	0	4	65	5	66	2.26e-13	60.8
Aco031087	Aco028251	58.491	53	22	0	7	59	306	358	2.97e-13	63.2
Aco031088	Aco031088	100.000	102	0	0	1	102	1	102	1.59e-71	207
Aco031088	Aco017406	93.333	60	4	0	43	102	66	125	7.24e-34	113
Aco031088	Aco020911	90.698	43	4	0	60	102	34	76	5.18e-21	79.0
Aco031088	Aco028523	68.627	51	15	1	35	85	70	119	1.61e-14	64.7
Aco031088	Aco028427	66.667	48	16	0	26	73	122	169	9.99e-14	63.5
Aco029824	Aco030364	99.800	500	1	0	1	500	1	500	0.0	1029
Aco029824	Aco029824	100.000	500	0	0	1	500	1	500	0.0	1029
Aco029824	Aco024758	99.502	402	2	0	1	402	1	402	0.0	819
Aco029824	Aco002996	44.165	437	215	4	55	491	83	490	2.23e-128	382
Aco029824	Aco013440	42.130	432	220	5	60	491	87	488	6.75e-122	365
Aco029826	Aco029826	100.000	205	0	0	1	205	1	205	1.51e-153	423

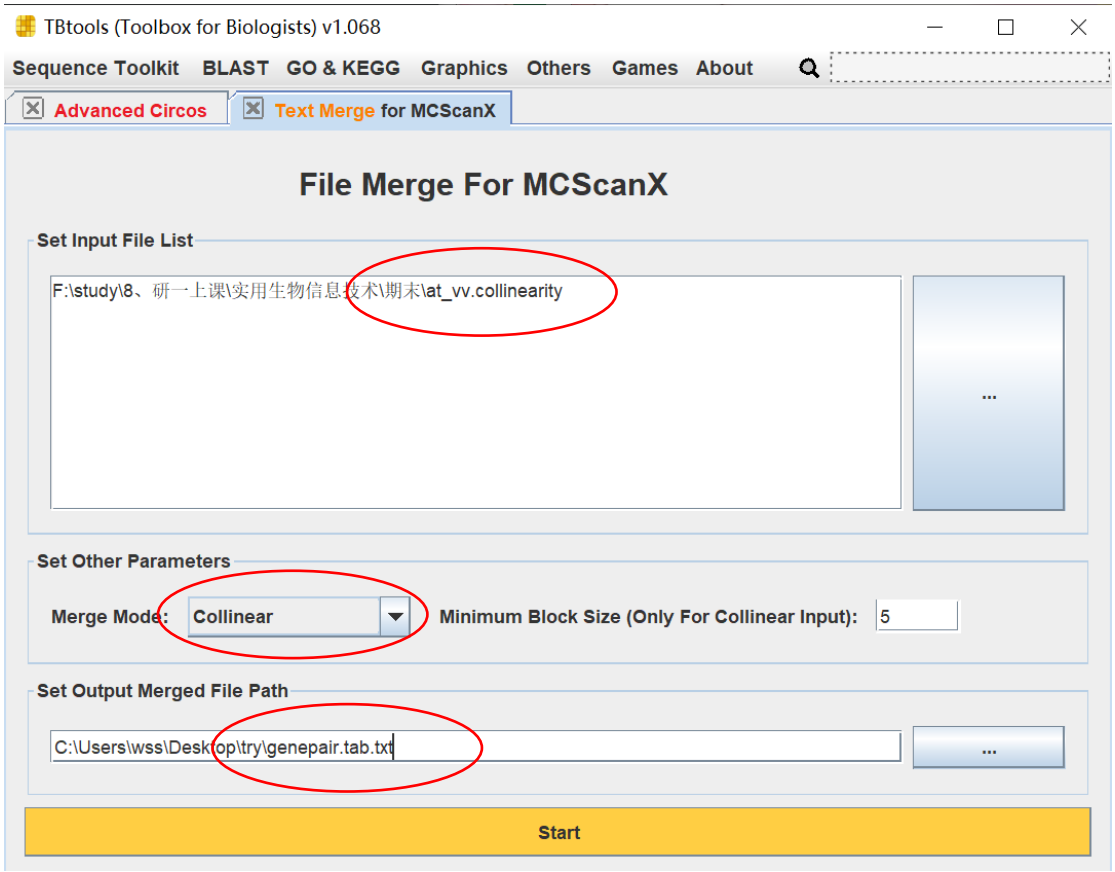
7、**共线性分析**（需要 blast 文件和修改后的 gff 文件，二者名字需相同）:

```
MCScanX at_vv
```

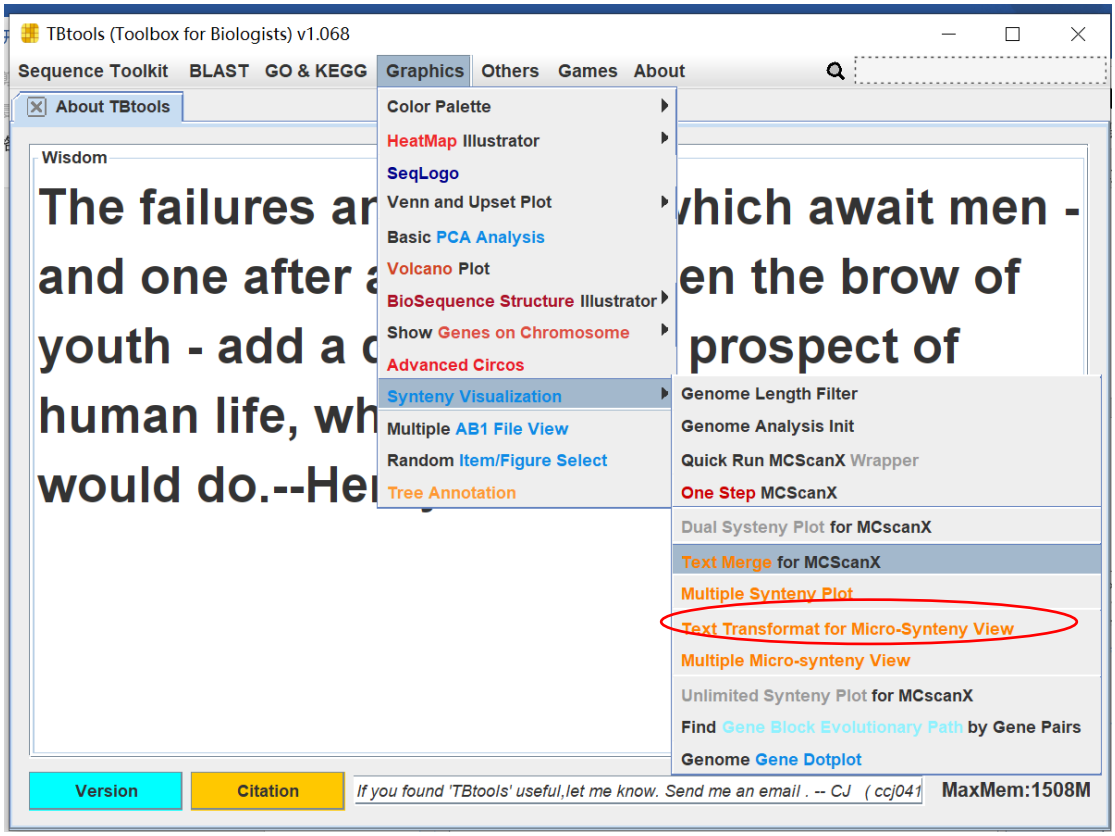
```
############### Parameters ###############
# MATCH_SCORE: 50
# MATCH_SIZE: 5
# GAP_PENALTY: -1
# OVERLAP_WINDOW: 5
# E_VALUE: 1e-05
# MAX GAPS: 25
############### Statistics ###############
# Number of collinear genes: 5176, Percentage: 19.15
# Number of all genes: 27024
##########################################
## Alignment 0: score=343.0 e_value=2.1e-14 N=8 LG01&LG02 plus
  0-  0:        Aco009502       Aco020926         3e-45
  0-  1:        Aco009497       Aco018982         3e-20
  0-  2:        Aco009496       Aco018983             0
  0-  3:        Aco009490       Aco018990         3e-33
  0-  4:        Aco009487       Aco018995         1e-23
  0-  5:        Aco009486       Aco019017             0
  0-  6:        Aco009479       Aco023268         1e-42
  0-  7:        Aco009478       Aco023267         3e-66
## Alignment 1: score=497.0 e_value=9.3e-27 N=12 LG01&LG02 minus
  1-  0:        Aco009576       Aco001189        5e-126
  1-  1:        Aco009575       Aco001192        1e-158
  1-  2:        Aco009572       Aco001196         3e-14
  1-  3:        Aco009571       Aco001200         3e-18
  1-  4:        Aco009562       Aco001208             0
  1-  5:        Aco009558       Aco001233             0
  1-  6:        Aco009557       Aco001236         6e-40
  1-  7:        Aco009550       Aco001256         5e-34
  1-  8:        Aco009547       Aco001267        5e-109
  1-  9:        Aco009545       Aco001272         7e-66
  1- 10:        Aco009533       Aco001295         3e-11
at_vv.collinearity
```

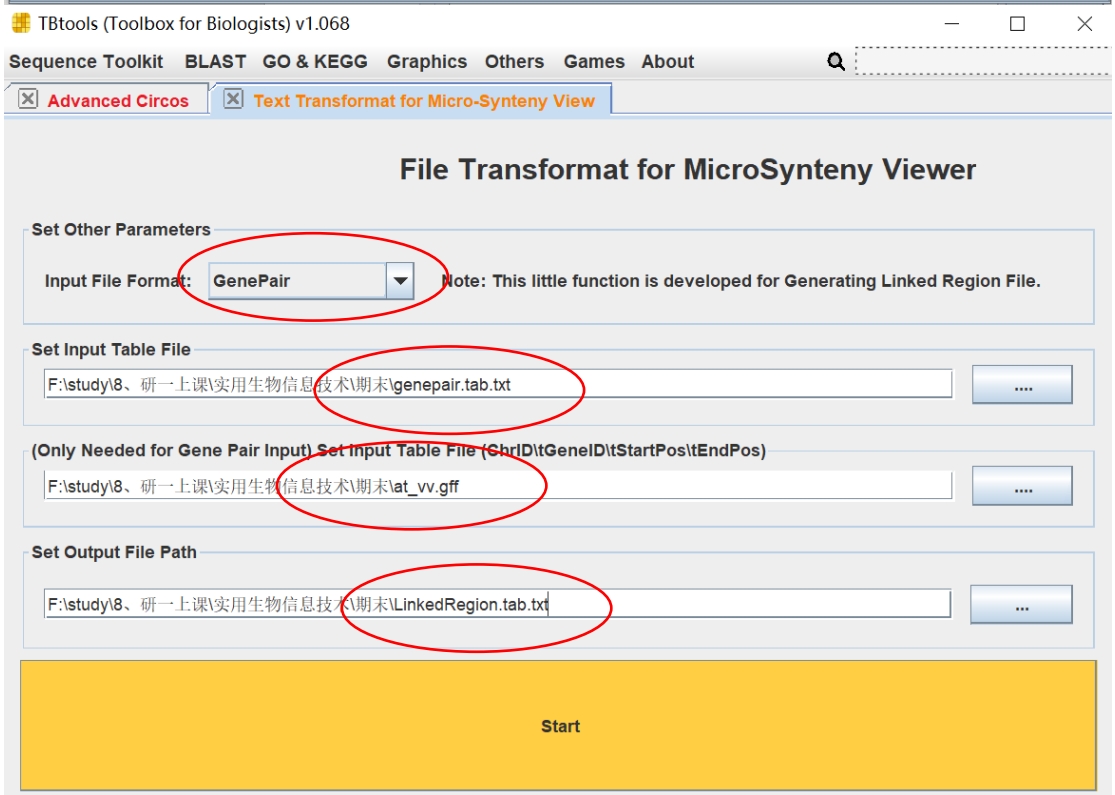
8、用 TBtools 进行可视化





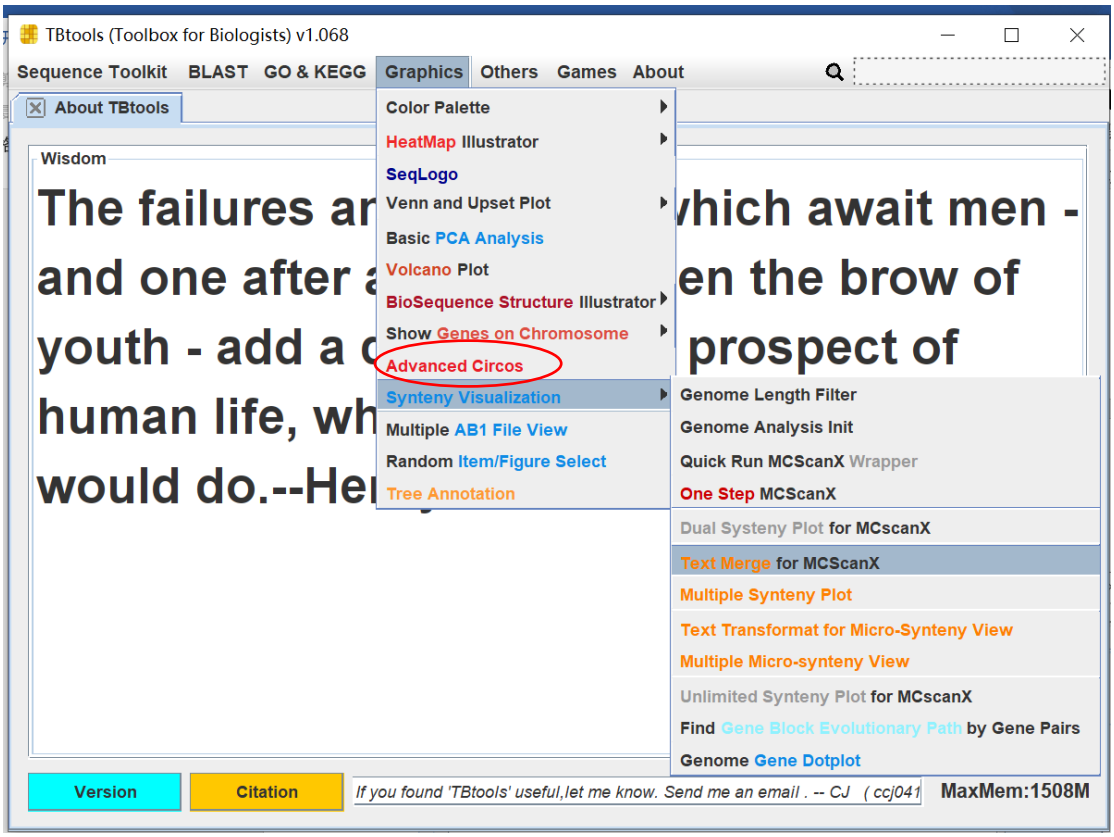
.collinearity 是上一步共线性分析得到的结果文件

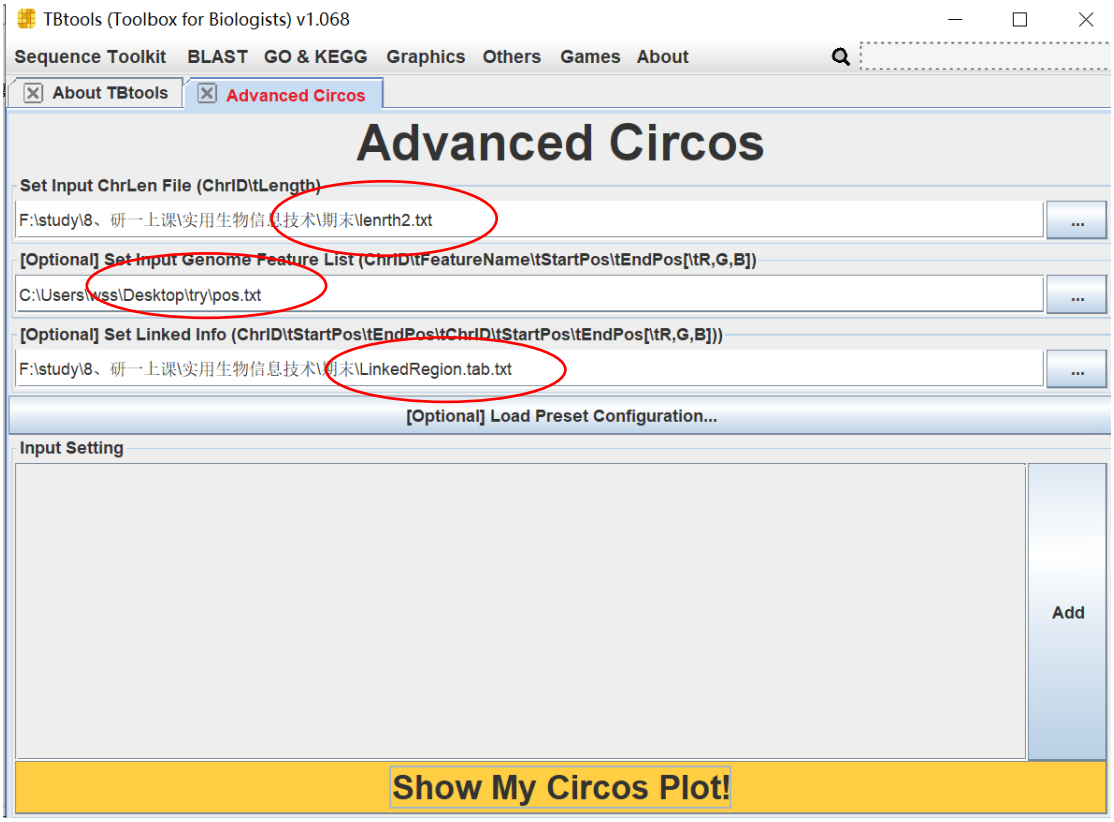




如果仅用这步生成的 LinkedRegin.tab.txt，共线性图中发生片段复制的 genepair 无法标红，如需要标红，需对 LinkedRegion.tab.txt 文件进行修改，我的修改方式是：下载文献中提供的发生片段复制的 genepair 的文件，根据 genepair 文件在 gff 文件中提取出染色体号、基因起始和终止位点，最后一列加上颜色（255,0,0），与原 LinkedRegion.tab.txt 合并

LG16	7825395	7827347	LG16	8009318	8012297	255,0,0	
LG17	5033431	5038173	LG21	967102	970815	255,0,0	
LG18	9059367	9062433	LG23	1796428	1797663	255,0,0	
LG19	8375628	8377151	LG02	2136871	2138236	255,0,0	
LG19	9102478	9103856	LG06	2507335	2509881	255,0,0	
LG23	1796428	1797663	LG23	1831669	1832873	255,0,0	
LG23	1831669	1832873	LG04	13339973	13345462	255,0,0	
LG01	1748493	1757814	LG02	10281460	10289138		
LG01	1792543	1794559	LG02	10545290	10547516		
LG01	1802973	1808200	LG02	10564196	10569744		
LG01	1830737	1836326	LG02	10621774	10626213		
LG01	1870221	1870697	LG02	10684731	10687685		
LG01	1881432	1886154	LG02	10921757	10937118		
LG01	1916306	1917558	LG02	11260633	11262425		



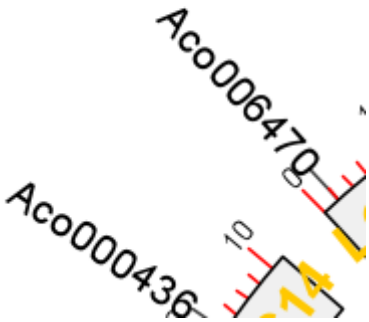


lenrth.txt（没错这个文件名是因为当时打错字了……这个是骨架文件，我是通过 gff 文件计算每条染色体的长度获得，菠萝的数据是一共 25 个染色体，截图是部分数据，这步应该有代码实现的方法，但是我还不会，于是用的 excel……：）

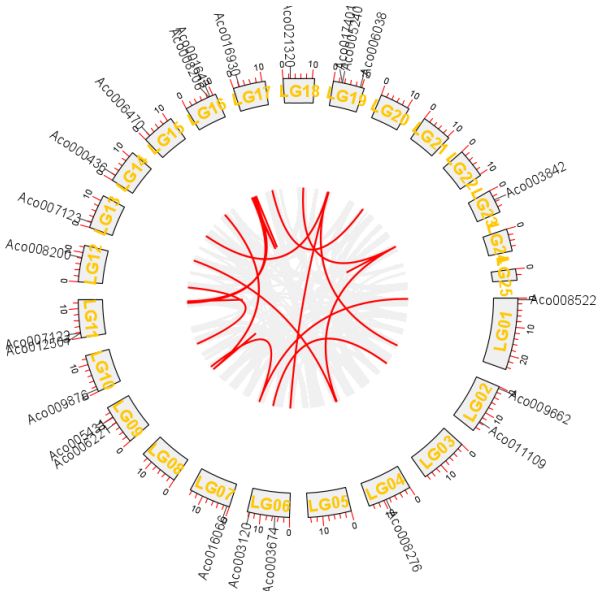
LG01	24861703
LG02	17292700
LG03	16740230
LG04	15571792
LG05	14984907
LG06	14721838
LG07	14699580

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

LG01	Aco008522	468304	470258
LG02	Aco009662	575884	578589
LG07	Aco016066	967102	970815
LG10	Aco009876	1037291	1046805
LG11	Aco012507	1681019	1689718
LG11	Aco007122	1796428	1797663
LG13	Aco007123	1831669	1832873
LG14	Aco000436	1850532	1853688
LG15	Aco006470	2051783	2056641

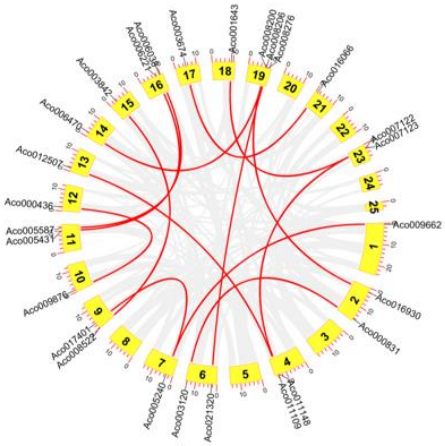


上述步骤后得到图：



Show Control Dialog

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）

② 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）

③ 重命名 xxx.uniq.bed 文件为 .bed 文件

2、**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`**

3、**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））

4、可视化：

```
python -m jcvi.graphics.karyotype seqids layout
```