

wgdi2jcvi

使用wgdi结果绘制jcvi的两种图： `Macrosynteny getting fancy` 和 `Microsynteny getting fancy`。

(一) Macrosynteny 大尺度共线性

1.数据准备及基础运行方式

(1) bed 文件

六列：染色体表示，起始，终止，id，占位0，方向。

if.bed						
1	1	1467	2621	if01g00001	0	+
2	1	2894	7353	if01g00002	0	+
3	1	8598	14142	if01g00003	0	+
4	1	8598	12903	if01g00004	0	+
5	1	8598	14142	if01g00005	0	+
6	1	12329	15178	if01g00006	0	-
7	1	20308	23646	if01g00007	0	-
8	1	28306	30861	if01g00008	0	-
9	1	45159	51084	if01g00009	0	-
10	1	56603	60757	if01g00010	0	+
11	1	56704	60589	if01g00011	0	+
12	1	63204	69364	if01g00012	0	+
13	1	69049	73625	if01g00013	0	+
14	1	69051	73590	if01g00014	0	+

(2) seqids 文件

包含需要绘制的染色体标识信息，每个物种一行，顺序与layout中的track顺序保持一致。

seqids.txt	
1	01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,19
2	01,02,03,04,05,06,07,08,09,10,11,12,13,14,15

(3) layout 文件

绘图布局文件，包含两部分信息

第一个#内(控制染色体的排布)

y轴，x轴起始，x轴终止，旋转，颜色，文本标签，垂直对齐方式，bed文件。

第二个#内(控制染色体内的连线)

固定占位e, 连线的track1, 连线的track2, simple文件

```
layout.txt
1 # y, xstart, xend, rotation, color, label, va, bed
2 .6, .2, .85, 0, , Grape, top, grape.bed
3 .4, .2, .85, 0, , I.trifida, top, sweetpotato.bed
4 # edges
5 e, 0, 1, grape.sweetpotato.anchors.simple.colored
```

(4) 运行方式

```
# 两个位置参数: seqids文件 和 layout文件 的路径。
python -m jcv.graphics.karyotype seqids layout
```

2. 数据生成方式

(1) 生成 bed 文件

```
python wgd2jcv.py gff2bed -gff input_gff_file -bed output_bed_file

#
# input_gff_file          wgd支持的gff格式
# output_bed_file        jcv支持的bed格式
```

(2) 生成 simple 文件

```
python wgd2jcv.py block2simple -c input_collinearity_file -o output_simple_file
[--filter filter_blockid_file]

# -c input_collinearity_file    wgd生成的collinearity文件
# -o output_simple_file        jcv支持的simple文件
# --filter filter_blockid_file 需要保留的block, 可使用wgd生成的筛选后的blockinfo文件的ID列。

# 举例1 不使用block筛选
python wgd2jcv.py block2simple -c coca_schi.collinearity.txt -o
coca.schi.simple

# 举例2 使用block筛选
# coca_chi_filter_blockid.txt 来源于最终版blockinfo的ID列。
python wgd2jcv.py block2simple -c coca_schi.collinearity.txt -o
coca.schi.simple --filter coca_chi_filter_blockid.txt
```

(二) Microsynteny 小尺度共线性

2. 数据准备及基础运行方式

(1) blocks 文件

包含共线的基因对，可以从wgdi的alignment中截取一部分。

```
blocks
1  GSVIVT01012261001  .
2  GSVIVT01012259001  .
3  GSVIVT01012258001  .
4  GSVIVT01012257001  .
5  GSVIVT01012255001  Prupe.1G290900.1
6  GSVIVT01012253001  Prupe.1G290800.2
7  GSVIVT01012252001  Prupe.1G290700.1
8  GSVIVT01012250001  Prupe.1G290600.1
9  GSVIVT01012249001  Prupe.1G290500.1
10 GSVIVT01012248001  Prupe.1G290400.1
11 GSVIVT01012247001  Prupe.1G290200.2
12 GSVIVT01012246001  Prupe.1G290100.3
13 GSVIVT01012245001  Prupe.1G290000.1
14 GSVIVT01012244001  Prupe.1G289800.1
15 GSVIVT01012243001  .
16 GSVIVT01012242001  .
17 GSVIVT01012241001
```

(2) block.layout 文件

绘图布局文件，包含两部分信息

第一个#内(控制基因对的排布)，每个物种的基因排布占一行

```
# x, y, rotation, ha, va, color, ratio, label
x轴, y轴, 旋转, 水平对齐方式, 垂直对齐方式, 缩放比例, 文本标签
```

第二个#内(控制基因对间的连线)

固定占位e, 连线的track1, 连线的track2

```
blocks.layout
1 # x, y, rotation, ha, va, color, ratio, label
2 0.5, 0.6, 0, left, center, m, 1, grape Chr1
3 0.5, 0.4, 0, left, center, #fc8d62, 1, peach scaffold_1
4 # edges
5 e, 0, 1
```

(3) 合并的 bed 文件

```
cat xxx.bed yyy.bed > xxx_yyy.bed
```

(4) 运行方式

```
python -m jcvi.graphics.synteny blocks xxx_yyy.bed blocks.layout
```

```
# blocks          想要展示局部blocks文件
```

```
# xxx_yyy.bed     合并的bed文件
```

```
# blocks.layout   blocks.layout文件
```

```
# 范例:
```

```
python -m jcvi.graphics.synteny blocks grape_peach.bed blocks.layout
```