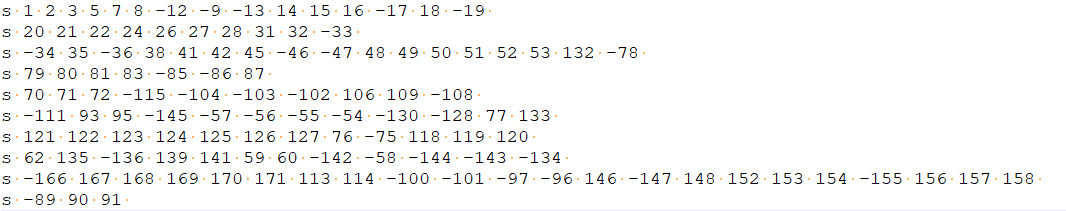
**User Guide**

**Title: IAGS: Inferring Ancestor Genome Structure under a wide range of evolutionary scenarios**

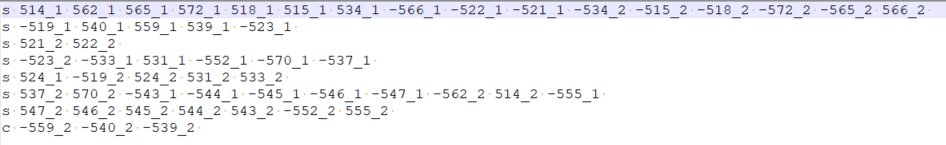
**IAGS file format**

IAGS take the GRIMM format. For example,



**Fig. 1| Example block sequence file format for IAGS.** Brassica rapa block sequence.

The first item represents the chromosome type. Since the result of IAGS may produce circular genome structure (Discussed in Discussion part), we used “s” represents a string chromosome and “c” represents a circular chromosome. The next items are synteny block order and “-” represent reverse blocks. All number are synteny block index and split by space. For some output, the block may contain bar, like “\_1”, “\_2”. For example,



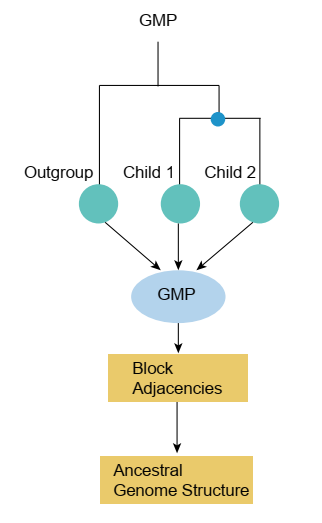
**Fig. 2| Example block sequence file with bar.**

which used to mark blocks with multi-copy.

We allow users to build synteny blocks in different ways and encourage user to use DRIMM-Synteny (<https://doi.org/10.1093/bioinformatics/btq465> ) to build non-overlapping synteny blocks. But the copy number of input blocks should satisfy target copy number based on whole genome duplication (no WGD block copy number is 1, one WGD block copy number is 2 and two WGD block copy number is 4).

**Core functions**

1. **GMP model:** ./model/GMPmodel.py



**Fig. 3| GMP workflow.**

GMP model is suitable for single copy block ancestor with no WGD.

**Parameters for GMP:**

|  |  |
| --- | --- |
| **Parameters** | **Meaning** |
| species\_file\_list | input species block sequence file list |
| outdir | output directory |
| ancestor\_name | ancestor name |

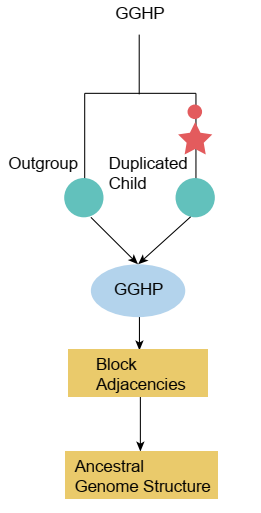
**Example usage:**

./scenarios/Brassica.py

**Important output:** ancestor\_name.block, for example:

./outputdata/Brassica/Brassica.block

**2. GGHP model:** ./model/GGHPmodel.py



**Fig. 4| GGHP workflow.**

GGHP model is suitable for single copy block ancestor with one WGD.

**Parameters for GGHP:**

|  |  |
| --- | --- |
| **Parameters** | **Meaning** |
| dup\_child\_file | block sequence file for duplicated child species |
| outgroup\_file | block sequence file for outgroup species |
| outdir | output directory |
| ancestor\_name | ancestor name |
| dup\_copy\_number | target copy number of duplicated child species |
| out\_copy\_number | target copy number of outgroup child species |

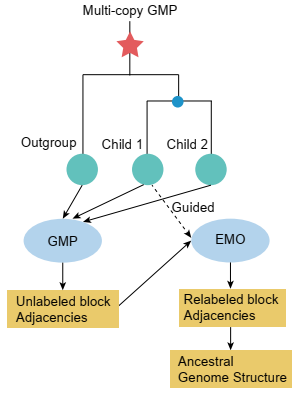
**Example usage:**

./scenarios/Yeast.py

**Important output:** ancestor\_name.block, for example:

./outputdata/Yeast/preWGD\_yeast.block

**3. Multi-copy GMP model:** ./model/MultiGMPmodel.py



**Fig. 5| Multi-copy GMP workflow.**

Multi-copy GMP model is suitable for multicopy block ancestor with shared WGD. The block copy number of ancestor is same with child species.

**Parameters for Multi-copy GMP:**

|  |  |
| --- | --- |
| **Parameters** | **Meaning** |
| species\_file\_list | input species block sequence file list |
| outdir | output directory |
| guided\_species\_for\_matching | a guided child species block sequence file |
| ancestor\_name | ancestor name |
| ancestor\_target\_copy\_number | target copy number of ancestor species |

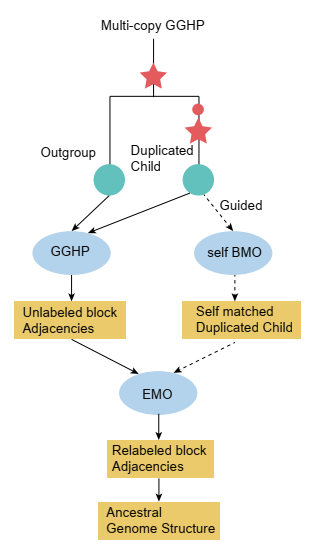
**Example usage:**

./scenarios/Gramineae.py (Ancestor 2)

**Important output:** ancestor\_name.block, for example:

./outputdata/Gramineae/Ancestor2/Ancestor2.block

**4. Multi-copy GGHP model:** ./model/MultiGGHPmodel.py



**Fig. 6| Multi-copy GGHP workflow.**

Multi-copy GGHP model is suitable for multicopy block ancestor with shared WGD. But the block copy number of child species is duplicated compared with ancestor.

**Parameters for Multi-copy GGHP:**

|  |  |
| --- | --- |
| **Parameters** | **Meaning** |
| dup\_child\_file | block sequence file for duplicated species |
| outgroup\_file | block sequence file for outgroup species |
| outdir | output directory |
| ancestor\_name | ancestor name |
| dup\_copy\_number | target copy number of duplicated species |
| out\_copy\_number | target copy number of outgroup species |
| ancestor\_target\_copy\_number | target copy number of ancestor species |

**Example usage:**

./scenarios/Papaver.py (Ancestor 3)

**Important output:** ancestor\_name.block, for example:

./outputdata/Papaver/Ancestor3/Ancestor3.block

**Supporting functions**

1. **Evaluation of inferred ancestor:**

./util/calculatedCRBrateAndEstimationAccuracy.py

IAGS provides inferred ancestor evaluation function to calculate completely rearranged endpoint ratio and estimated accuracy.

**Parameters for statisticsAdjacency:**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Meaning** | |
| matching\_target\_file | | block sequence file for target species |
| matching\_target\_copy\_number | | target copy number of target species |
| matching\_target\_name | | target species name |
| speciesAndCopyList | | all species block sequences file,  target copy number and species name |
| outdir | | output directory |
| model\_type | | model used for obtaining ancestor,  including GMP, GGHP, MultiCopyGMP and MultiCopyGGHP |

**Example usage:**

./scenarios/Gramineae.py (Ancestor 4)

**Important output:**

Calculates CRB ratio and estimation accuracy, for example:

./outputdata/Gramineae/Ancestor4/ev.txt

1. **Counting shuffling events:**

./util/calculateFissionAndFussions.py

IAGS provides downstream analysis for counting shuffling events, like fissions and fusions, which takes into two species block sequences and copy number of species 2 (ancestor) cannot larger than species 1 (descendant).

**Parameters for calculateFissionAndFussions:**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Meaning** | |
| species1\_file | | species 1 block sequence file |
| species2\_file | | species 2 block sequence file |
| sp1\_copy\_number | | target copy number of species 1 |
| sp2\_copy\_number | | target copy number of species 2 |
| outdir | | output directory |

**Example usage:**

./scenarios/PapaverShufflingEvents.py

**Important output:**

./outputdata/Papaver/shufflingEvents.txt

1. **Rearrangement painting:**

./util/chromosomeRearrangementPainting.py

IAGS allows output chromosomes rearrangement painting which takes into two species block sequences files. One is target species (ancestor) and the other is rearranged species (descendant).

**Parameters for calculateFissionAndFussions:**

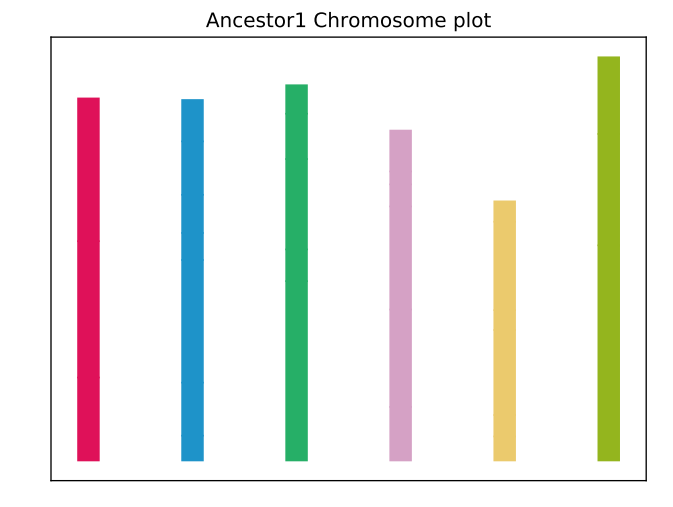
|  |  |  |
| --- | --- | --- |
| **Parameters** | **Meaning** | |
| block\_length\_file | | a table recorded each block length |
| rearranged\_species\_block\_file | | rearranged species block sequence file |
| rearranged\_species\_name | | name of rearranged species |
| rearranged\_species\_copy\_number | | target copy number of rearranged species |
| target\_species\_block\_file | | target species block sequence file |
| target\_species\_name | | name of target species |
| target\_species\_copy\_number | | target copy number of target species |
| colorlist | | colors for chromosomes in target species |
| outdir | | output directory |

**Example usage:**

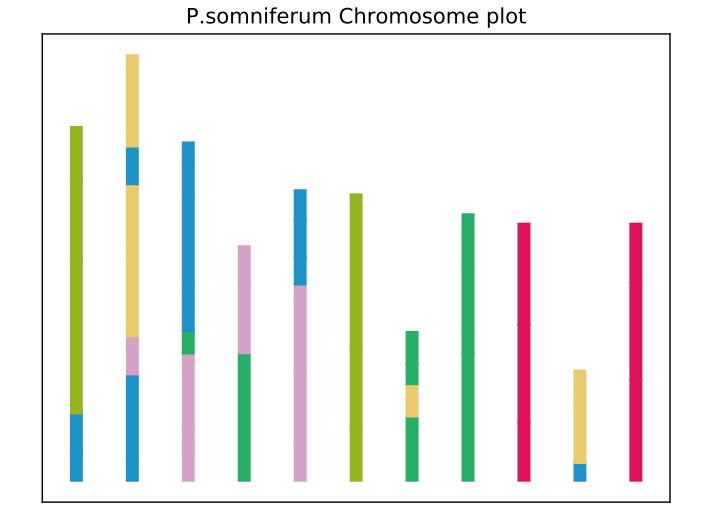
./scenarios/PapaverChromosomePainting.py

**Important output:**

./outputdata/Papaver/plot/



**Fig. 7| Target species chromosome painting.**



**Fig. 8| Rearranged species chromosome painting.**