**User Guide**

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

**Environmental requirements**

Python 3.6

|  |  |
| --- | --- |
| **Packages** | **Version used in Research** |
| numpy | 1.16.4 |
| pandas | 0.20.3 |
| matplotlib | 3.0.3 |

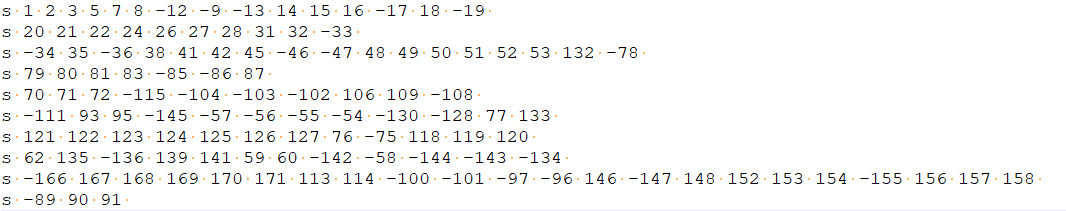
Gurobi solver 9.0.2 (<https://www.gurobi.com/> ) with Academic License.

Development environment: Windows 10

Development tool: Pycharm

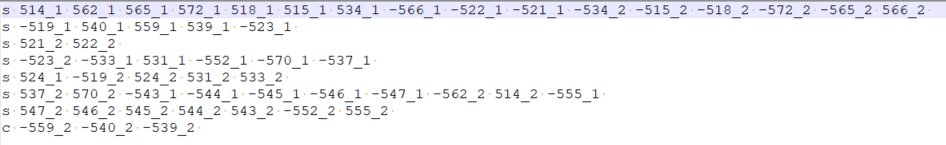
**IAGS file format**

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, 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,



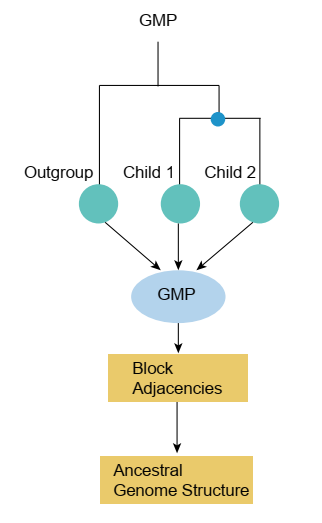
**Fig. 2| Example block sequence file with bar.**

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 takes into some species block sequence files and transforms block sequence into block adjacencies. IAGS uses GMP integer programming formulations based on these block adjacencies to get ancestral block adjacencies and then directly transforms to block sequence.

**Parameters for GMP:**

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

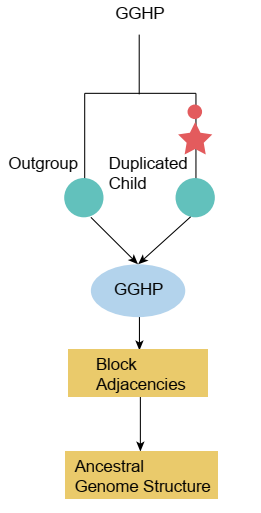
**Example usage:**

./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 takes into duplicated and outgroup species block sequences. Ancestor block copy number should be only one. IAGS transforms both block sequences into block adjacencies. IAGS uses GGHP integer programming formulations based on block adjacencies to get ancestral block adjacencies and then directly transforms to block sequence. For basic GGHP, target copy number of duplicated species is two and outgroup species is one. IAGS allow multiple species as input which duplicated species block sequences and outgroup species block sequences should be merged together, respectively and the input target block copy number should be summed, respectively.

**Parameters for 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 |

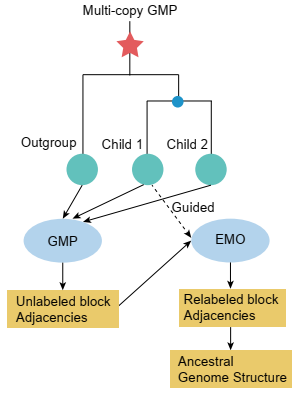
**Example usage:**

./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 takes into some species block sequence files and transforms block sequence into block adjacencies which is same with GMP. But GMP integer programming formulations can just obtain ancestral block adjacencies. Ancestral block adjacencies are multi-copy. IAGS followed child guide strategy to transform multi-copy ancestral block adjacencies to sequences using EMO integer programming formulations.

**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 |

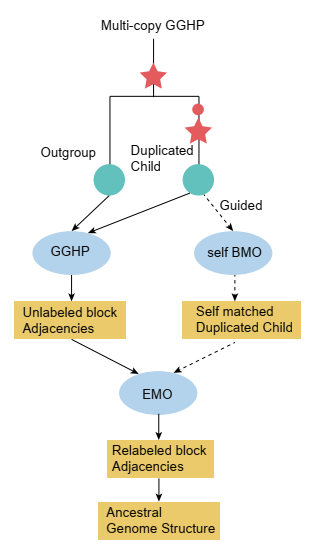
**Example usage:**

./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 takes into duplicated and outgroup species block sequences. Ancestor block copy number can be more than one. IAGS transforms both block sequences into block adjacencies which is same with GGHP. But GGHP integer programming formulations can just obtain ancestral block adjacencies. Ancestral block adjacencies are multi-copy. IAGS followed child guide strategy to transform multi-copy ancestral block adjacencies to sequences. IAGS first used self-BMO integer programming formulation to remove the influence of WGD in child species and then used EMO integer programming formulation.

**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/statisticsAdjacency.py

IAGS provides inferred ancestor evaluation function which contains three part. Firstly, calculating ancestral adjacencies support table. All species used for calculating this ancestor should first match with this ancestor by BMO integer programming formulations and then counting the number of block adjacencies appeared in all species, respectively. Secondly, ancestors may appear circular chromosomes. Here, IAGS allows user to cut one adjacency in circular chromosomes with minimum number of support to make circular to strings. Finally, IAGS calculates completely rearranged breakpoints ratio and obtains estimation accuracy by accuracy estimation function.

**Parameters for statisticsAdjacency:**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Meaning** | |
| ancestor\_file | | block sequence file for inferred ancestor |
| ancestor\_copy\_number | | target copy number of ancestor |
| ancetsor\_name | | ancestor 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 |
| cutcycle | | parameter for whether cut circular chromosomes |
| getCRBratio | | parameter for whether calculated CRB ratio and estimation accuracy |

**Example usage:**

./scenarios/Gramineae.py (Ancestor 4)

**Important output:**

Calculating ancestral adjacencies support table, for example:

./outputdata/Gramineae/Ancestor4/ev\_Ancestor4.xls

Processing circular chromosomes, for example:

./outputdata/Gramineae/Ancestor4/Ancestor4.cutcycle.block

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). If the copy number of species 1 is not equal to species 2 because of WGDs, block sequence of species 2 should be amplified to species 1. Then, IAGS used BMO matching both species and transformed to adjacencies. The adjacencies absent in species 2 are fusions and absent in species 1 are fissions.

**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). IAGS used BMO matching both species and then plots chromosomes painting.

**Parameters for calculateFissionAndFussions:**

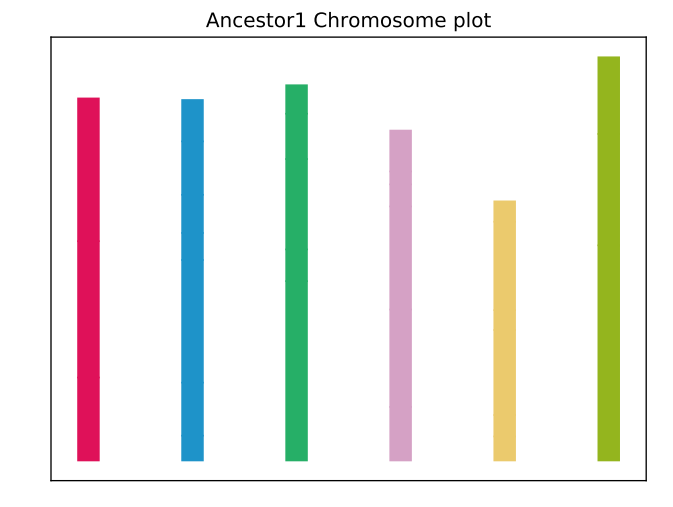
|  |  |  |
| --- | --- | --- |
| **Parameters** | **Meaning** | |
| block\_length\_file | | a table recorded each block length |
| rearranged\_species\_block\_file | | rearranged species block sequence file |
| rearranged\_species\_name | | name of rearranged species |
| rearranged\_species\_copy\_number | | 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:**

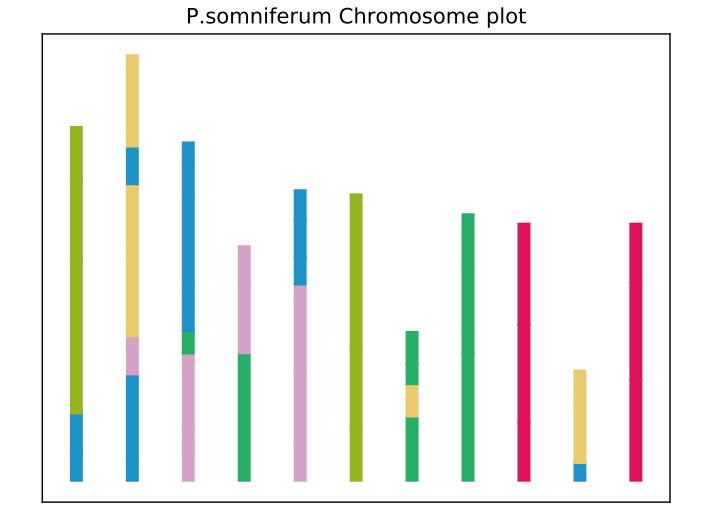
./scenarios/PapaverChromosomePainting.py

**Important output:**

./outputdata/Papaver/plot/



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



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