WGD_Tracker

Summary

Introduction

WGD_Tracker is a tool designed to ease (intra- and inter-) genomic comparisons for detecting whole genome duplication events and to provide WGD dating by Ks analyses. This tool is fully customizable, allowing users to select stringent or flexible parameters, detailed in this document, depending on their specific goals.

This tool includes five pipelines:

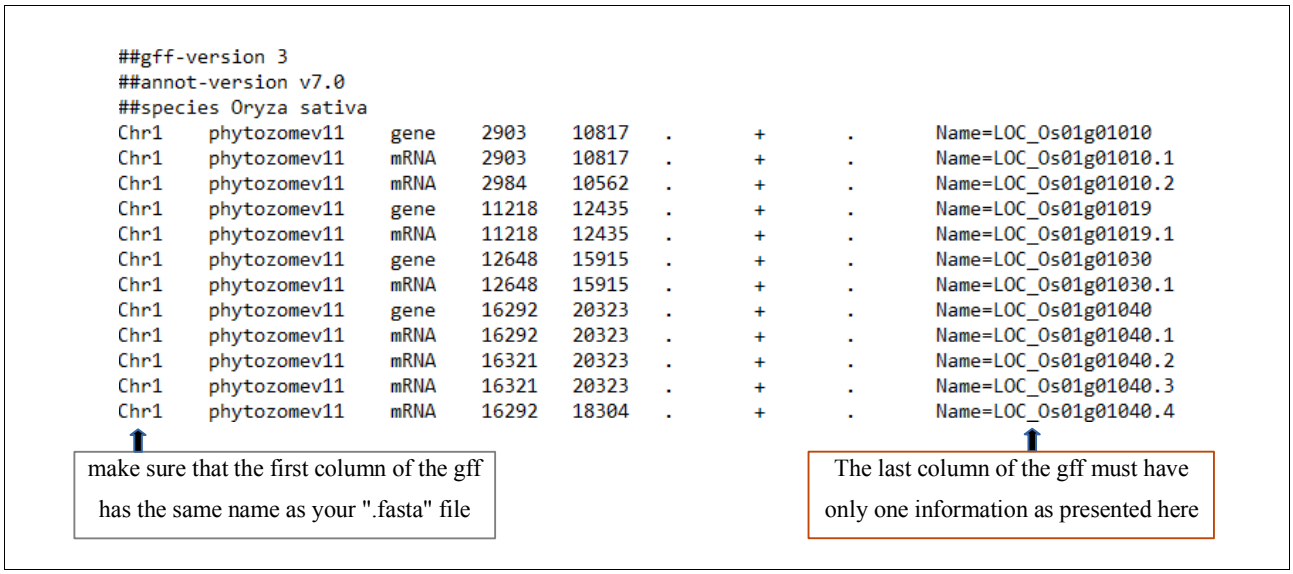
1) The **RBBH Pipeline** identifies homologous gene pairs via reciprocal BLAST best hit (RBBH) analysis from a BLAST output file. Because polyploid genomes contain multiple copies of the same genes, this pipeline can also search for reciprocal blast best hits (RBH) and not just the best hits in order to highlight all putative duplicated gene copies.
2) The **Ks Pipeline** allows the calculation of synonymous substitution rates (Nei & Gojobori model) between gene pairs.
3) The **Synteny Pipeline** accurately identifies syntenic blocks even when dispersed.
4) The **Dotplot Pipeline** generates graphical plots from the outputs of the three previous Pipelines (RBBH; Ks; or Synteny).
5) **Karyotype Pipeline** generates a graphical representation of conserved syntenic blocks between two genomes as karyotypes, for each chromosome across both genomes.

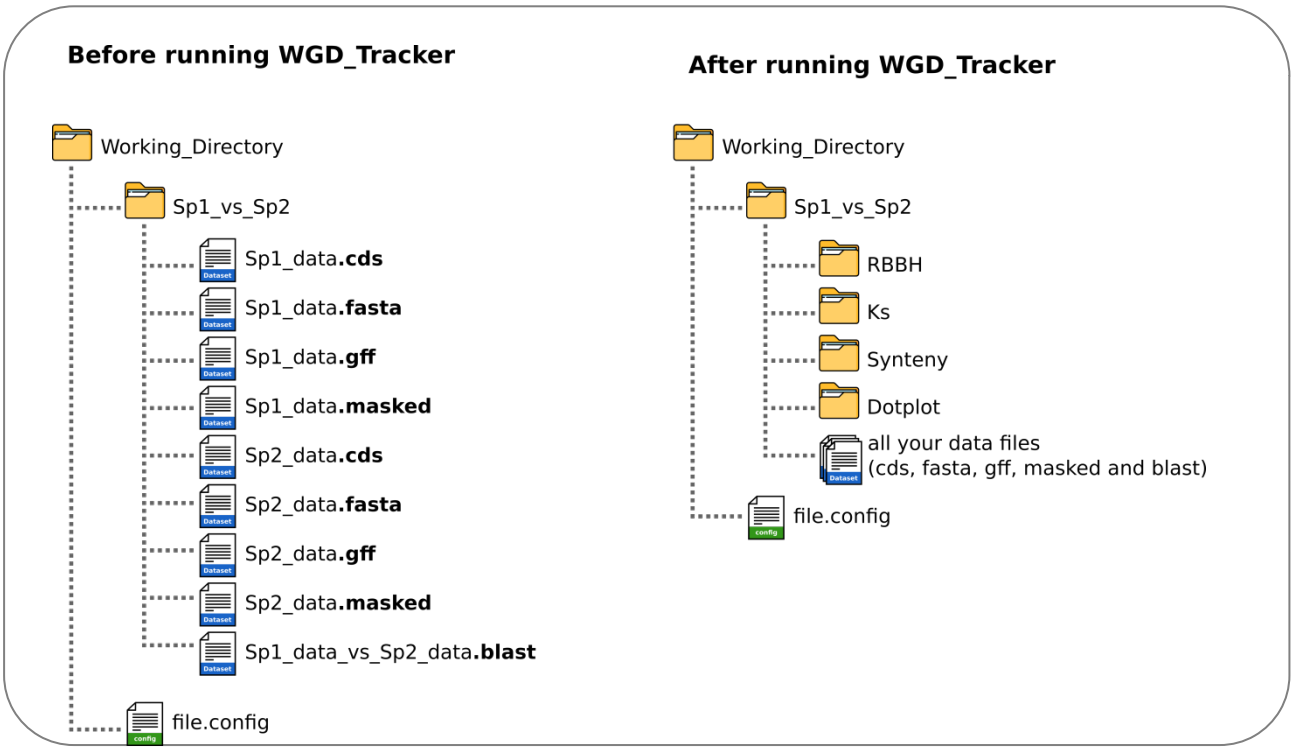
Organizing folders and files

The working directory must contain (1) the configuration file (**file.config**), (2) your working directory containing all data files and (3) the WGD_Tracker directory and dependencies. The folder name of your working directory must be specified in the configuration file (*e.g.* **data_dir**="Sp1_vs_Sp2"). Unlike the configuration file and the working directory, which can be named as you wish, the data files must be specifically named. In the configuration file, you need to specify the names of the compared genomes (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You will need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension (*e.g.* Osativa_cds.fasta, Osativa_cds.gff, Osativa_cds.cds).

The extensions to use are:

- **.fasta** for your genome assembly in fasta format
- **.cds** for your fasta file containing all your cds sequences (check for transcript)
- **.masked** for your masked genome assembly
- .**gff** for your gff3 files (note that some formatting is required, see below)
- **.blast** for the output file from the BLAST analysis





Once the analyses are complete, you will find the results in the same folder that contains your data file. Each pipeline creates a specific folder containing all output files.

Installation

WGD_Tracker installation can be performed with a git clone. However, WGD_Tracker does **require dependencies**. The following table describes the dependencies required for each pipeline:

	RBBH	Ks	Synteny	Dotplot	Karyotype
Python 3	X	X	X	X	X
Java		X			
PAML		X			
MACSE		X			
R		X			
Singularity		X			
Parallel	X	X			

Please find the recommended dependencies version, which was used when developing the tool: Python v3.9, Parallel 20190122, Java v1.8.0, PAML v4.9, MACSE v2.05, R v4.1.0, Singularity v3.8.0

Retrieve WGD_Tracker

```
$ git clone https://github.com/MorganeMilin/WGD_Tracker.git
```

Conda environment creation: (Please note that you must have access to conda)

```
$ conda env create -p ./dependencies_conda_env -f ./WGD_Tracker/wgd_tracker_dependencies.yml
```

Image sif for R dependencies: (Please note that you must have access to singularity and the generated .sif image must be located in the WGD_Tracker folder)

```
$ cd WGD_Tracker
$ singularity build ./rmarkdown.sif ./wgd_tracker_R.def
```

Usage

To run the RBBH Pipeline:

```
$ sbatch --cpus-per-task=2 ./WGD_Tracker/RBBH_Pipeline.txt ./file.config
```

To run the Ks Pipeline:

```
$ sbatch --cpus-per-task=20 ./WGD_Tracker/Ks_Pipeline.txt ./file.config
```

To run the Synteny Pipeline:

```
$ sbatch ./WGD_Tracker/Synteny_Pipeline.txt ./file.config
```

To run the Dotplot Pipeline:

```
$ sbatch ./WGD_Tracker/Dotplot_Pipeline.txt ./file.config
```

To run the Karyotype Pipeline:

```
$ sbatch ./WGD_Tracker/Karyotype_Pipeline.txt ./file.config
```

RBBH Pipeline

Inputs

Several input files must be provided for the RBBH pipeline to run properly:

1. The ".fasta" and/or ".cds" file(s) used to generate the BLAST analysis
2. The BLAST output file (tabular output format 6 without header, **"-outfmt 6"** option)
3. Optional - The masked fasta file, if you want to remove hits located in repeated regions
4. The gff file(s), properly formatted as described on page 2
5. The configuration file

Configuration file

It is important to fill in the configuration file according to your dataset and specific analytic purpose. Only a few parameters need to be specified in the configuration file (***i.e.* data_dir, tool_dir, Nb_CPU, intragenomic, SP1 and SP2**), the other parameters do not need to be specified unless you want to use different settings than defined by default.

data_dir
Mandatory - String expected
The absolute path to the Working Directory containing all data files (".fasta", ".gff", ".blast", etc.)

tool_dir
Mandatory - String expected
The absolute path to the WGD_Tracker tool folder

Nb_CPU
Mandatory - Integer expected (default: 2)
The number of cores to use for the analysis. Note that the value specified in the configuration file must be the same as the value specified in the sbatch line of code that executes the pipeline

intragenomic
Mandatory - Boolean expected: "True" or "False"
Specify whether this is an intragenomic (*e.g.* "True") or intergenomic (*e.g.* "False") analysis

SP1 and **SP2**
Mandatory - String expected
The name(s) of your compared genome(s) must be specified here (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You'll need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension

identity
Optional - Integer expected (default: 70)
Minimum identity (in %) value that will be kept in the dataset

len_align

Optional - Integer expected (default: 60)
Minimum alignment length (in nucleotide) that will be kept in the dataset

len_ratio

Optional - List of two values expected (default: not used)
only if at least one dataset used in BLAST are CDS and/or Transcripts
1st value corresponds to the minimum percentage alignment requested (integer expected: within 0 and 100)
2nd value corresponds to the number of CDS and/or transcript files used (string expected: "single" or "double")

Example:

Genomic Comparisons	Settings that must be used
Species1 CDS *versus* Species2 Genomic	len_ratio="[%alignment, 'simple']"
Species1 CDS *versus* Species2 Transcript	len_ratio="[%alignment, 'double']"
Species1 CDS *versus* Species2 CDS	len_ratio="[%alignment, 'double']"
Species1 CDS *versus* Species1 CDS (*i.e.* intragenomic)	len_ratio="[%alignment, 'double']"

corr_intra

Optional - List of two values expected (default: not used)
1st value corresponds to a pattern to target (*e.g.* "." or "-")
2nd value corresponds to the number of elements to be deleted

Example:

Sequence name	Settings	Results
sequence1.1 sequence1.2 sequence1.3	corr_intra="['.', -1]"	sequence1 sequence1 sequence1
sequence1.1.2 sequence1.1.25 sequence1.1.256	corr_intra="['.', -2]"	sequence1 sequence1 sequence1
sequence1-1 sequence1-2 sequence1-1.2 sequence1-1-2	corr_intra="['-', -1]"	sequence1 sequence1 sequence1 sequence1-1

Useful for intragenomic analysis only. Multiple conformations can be found for a given sequence (*e.g.* LOC_Os01g01010.1, LOC_Os01g01010.2, etc). This parameter ensures that alignments against themselves are removed, including alignments of identical sequences with different conformations

coding

Optional - Boolean expected: "True" or "False" (default: False)
coding="True" allows to keep only alignments located in coding region (only if dataset used in BLAST are genome assembly)

coding_type

Optional - String expected: "simple" or "double" (default: not used)
Specify how many genomes you wish to filter

SP1_coding_infos and **SP2_coding_infos**

Optional - List of three values expected (default: not used)
1st value corresponds to the Species name (*e.g.* Osativa)
2nd value corresponds to the targeted motif of the third column in the gff file (*e.g.* "gene", "mRNA")

3rd value corresponds to the minimum length (in nucleotides) necessary to confirm that the alignment is in a coding region

Example:

Genomic Comparisons	Example of the settings that must be used
Species1 Genomic *versus* Species2 CDS	coding="True" coding_type="simple" SP1_coding_infos="['Species1','mRNA',60]"
Species1 Genomic *versus* Species2 Transcript	
Species1 Genomic *versus* Species2 Genomic	coding="True" coding_type="double" SP1_coding_infos="['Species1','mRNA',60]" SP2_coding_infos="['Species2','mRNA',60]" **If you only want to check if Species2 alignment are in coding region:** coding="True" coding_type="simple" SP1_coding_infos="['Species2','mRNA',60]" **If Intragenomic:** intragenomic="True" coding="True" coding_type="double" SP1_coding_infos="['Species1','mRNA',60]"

TErm

Optional - Boolean expected: "True" or "False" (default: False)
TErm="True" allows to remove alignments present in regions containing repeated elements

TErm_type

Optional - String expected: "simple" or "double" (default: not used)
Specify how many genomes you wish to filter

SP1_TErm and **SP2_TErm**

Optional - List of three values expected (default: not used)
1st value corresponds to the Species name (*e.g.* Osativa)
2nd value corresponds to the targeted motif of the third column in the gff file (*e.g.* "gene", "mRNA")
3rd value corresponds to the percentage of repeat authorized in the alignment

Example:

Genomic Comparisons	Example of the settings that must be used
if you want to check both Species1 and Species2 alignment	TErm="True" TErm_type="double" SP1_TErm="['Species1','mRNA',25]" SP2_TErm="['Species2','mRNA',25]"
if you only want to check Species1 alignment	TErm="True" TErm_type="simple" SP1_TErm="['Species1','mRNA',25]"
if you only want to check Species2 alignment	TErm="True" TErm_type="simple" SP1_TErm="['Species2','mRNA',25]"

if intragenomic analysis	intragenomic="True" TErm="True" TErm_type="double" SP1_TErm="['Species1','mRNA',25]"

If the dataset is a genome assembly, the percentage of repeats on the alignment is checked, whereas if the dataset is a CDS sequences, the percentage of repeats on the entire CDS sequence is verified and not just the part of the sequence aligned by blast

BH_limit

Optional - Integer expected (default: 1)

Number of best hit to keep for (1) each CDS or Transcript (if CDS and/or Transcript used in BLAST) or (2) for each non-overlapping windows (if genome assembly in BLAST)

interval

Optional - Integer expected (default: 50000)

Non-overlapping windows length (in nucleotides) to be used to analyzed each chromosome. Required only if the analysis was done on a genomic assembly

Outputs

The output files of this pipeline will be located in the RBBH folder:

- Several output files from filtration steps
    - pFlt_*.txt is the filter output file for identity and alignment length values
    - cds_*.txt and non_coding_*.txt are the output files separating hits found in coding and non-coding regions
    - TErm_*.txt corresponds to the output file after deletion of hits found in repeated regions
- **BH_*.txt** corresponds to the output file retrieving the best hit(s) for each sequence
- **RBBH_*.txt** is the final output file providing the Reciprocal Blast Hits

Output file(s) format is the same as BLAST output, plus a few columns in BH_*.txt and RBBH_*.txt file(s):

1. query or source (gene) sequence id
2. subject or target (reference genome) sequence id
3. percentage of identical positions
4. alignment length (sequence overlap)
5. number of mismatches
6. number of gap openings
7. start of alignment in query
8. end of alignment in query
9. start alignment in subject

10. end alignment in subject
11. expect value
12. bit score
13. best hit number

If the dataset used are genome assemblies:

14. length of the reciprocal alignment in the query sequence
15. start of the reciprocal alignment in the query sequence
16. end of the reciprocal alignment in the query sequence
17. length of the reciprocal alignment in the subject sequence
18. start of the reciprocal alignment in the subject sequence
19. end of the reciprocal alignment in the subject sequence

Ks Pipeline

Synonymous substitutions (Ks) are calculated here using the PAML program (v. 4.9; Yang 2007), with the Nei and Gojobori model (1986).

Inputs

Several input files must be provided for the Ks pipeline to run properly:

1. The ".cds" file(s)
2. The RBBH output file (the format is the same as a BLAST output file)
3. The configuration file

Configuration file

The Ks computation step generating many small files, the process is parallelized and analyses are split into different folders to avoid server saturation. Since this Ks calculation step can be parallelized, the creation of several folders with a smaller number of fasta files would help at analyzing a larger number of folders in parallel.

data_dir

Mandatory - String expected
The absolute path to the Working Directory containing all data files (".fasta", ".gff", ".blast", etc.).

tool_dir

Mandatory - String expected
The absolute path to the WGD_Tracker tool folder.

Nb_CPU

Mandatory - Integer expected (default: 2)
The number of cores to use for the analysis. Note that the value specified in the configuration file must be the same as the value specified in the sbatch line of code that executes the pipeline.
It is recommended that you take advantage of the parallelization of this pipeline to reduce the analysis time. To do so, simply increase this parameter according to the size of your dataset and the performance of your computer.

intragenomic

Mandatory - Boolean expected: "True" or "False"
Specify whether this is an intragenomic (*e.g.* "True") or intergenomic (*e.g.* "False") analysis.

SP1 and **SP2**

Mandatory - String expected
The name(s) of your compared genome(s) must be specified here (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You'll need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension.

Ks_begin

Optional - String expected (default: "fasta_Extract")
Specify the analysis step to start with. Either step 1 **"fasta_Extract",** which generates all fasta files to be analyzed; or step 2 **"Ks_calculation"**, which calculates the Ka and Ks for each pair of genes, or the last step **"Ks_distribution"**, which generates a Ks distribution and estimate the mode of the peak(s).

Ks_folder_limit

Optional - Integer expected (default: 10000)
Maximum number of folders.

Ks_file_limit

Optional - Integer expected (default: 2000)
Number of gene pairs to analyze per folder.

mxt_ksmin

Optional - Float or integer expected (default: 0.01)
Minimum Ks value kept for the analysis. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

mxt_ksmax

Optional - Float or integer expected (default: 3)
Maximum Ks value kept for the analysis. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

mxt_kmin

Optional - Integer expected (default: 2)
Minimum number of peaks expected. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

mxt_kmax

Optional - Integer expected (default: 4)
Maximum number of peaks expected (Warning: analysis time increases with k). Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

mxt_boots

Optional - Integer expected (default: 1000)
Bootstrapping effort during search for optimal number of peaks. (Warning: this is time consuming. Recommended value is 1000). Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

mxt_epsilon

Optional - Integer expected (default: 1e-3)
Convergence criterion; heuristics are stopped when loglik is improved by less than epsilon. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s)

mxt_breaks

Optional - Integer expected (default: 300)
Number of breaks on the histogram. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s).

Outputs

The output files of this pipeline will be located in the Ks folder:

- **Res_compil_NG_Ks_total.txt** - Output file containing the Ka, Ks and ratio values for each pair of genes compared.
- **Ks_distribution_NG.pdf** - Distribution of the Ks values and estimates the mode of the peak(s) using the R mixtools package.
- **Ks_distribution_NG_log_transformed.pdf** - Distribution of the Ks values (with a dataset logarithmic transformation) and estimates the mode of the peak(s) using the R mixtools package.
- **Ks_distribution_NG_sqrt_transformed.pdf** - Distribution of the Ks values (with a dataset square root transformation) and estimates the mode of the peak(s) using the R mixtools package.

Res_compil_NG_Ks_total.txt format:

1. gene1 - gene name used in the gene pair comparison
2. gene2 - second gene name used in the gene pair comparison
3. Ka - number of non-synonymous substitutions (altering) per non-synonymous site
4. Ks - number of synonymous substitutions per synonymous site
5. Ka/Ks - ratio used to assess selection pressure on coding regions

Synteny Pipeline

Inputs

Several input files must be provided for the Synteny pipeline to run properly:

1. The gff file(s), properly formatted as described on page 2
2. The configuration file

Configuration file

data_dir

Mandatory - String expected
The absolute path to the Working Directory containing all data files (".fasta", ".gff", ".blast", etc.)

tool_dir

Mandatory - String expected
The absolute path to the WGD_Tracker tool folder

intragenomic

Mandatory - Boolean expected: "True" or "False"
Specify whether this is an intragenomic (*e.g.* "True") or intergenomic (*e.g.* "False") analysis

SP1 and **SP2**

Mandatory - String expected
The name(s) of your compared genome(s) must be specified here (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You'll need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension

sp1_motif and **sp2_motif**

Mandatory - String expected
For each species specify the targeted motif of the third column in the gff file (*e.g.* "gene", "mRNA")

corr_SB

Optional - List of six values expected: "[SP1, string, integer, SP2, string, integer]"
1st and 4th values corresponds to species (SP1 or SP2)
2nd and 5th values corresponds a pattern to target (*e.g.* "." or "-")
3rd and 6th values corresponds to the number of elements to be deleted

gap

Optional - Integer expected (default: 100)
Maximum number of genes with no hits tolerated between two consecutive hits of the same syntenic block

gene_nb

Optional - Integer expected (default: 5)
Minimum number of hits required to define a syntenic block

Ks_min

Optional - Float or integer expected (default: 0.01)
Minimum Ks value kept for the analysis. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s)

Ks_max

Optional - Float or integer expected (default: 3)
Maximum Ks value kept for the analysis. Parameter needed to generate a graphical representation of the data and estimate the mode of the peak(s)

Outputs

The output files of this pipeline will be located in the Synteny folder. There are several output files corresponding to intermediate steps. There are two important files to keep, which correspond to syntenic hit results, presented in two different formats:

Res_compil_NG_Syntenic_blocks_STEP_3.txt format:

1. gene1 - gene name used in the gene pair comparison
2. gene2 - second gene name used in the gene pair comparison
3. Ka - number of non-synonymous substitutions (altering) per non-synonymous site
4. Ks - number of synonymous substitutions per synonymous site
5. Ka/Ks - ratio used to assess selection pressure on coding regions

Syntenic_blocks_STEP_3.txt format:

1. Reference id
2. Species1 Chromosome
3. Species2 Chromosome
4. Syntenic Block number
5. orientation
6. gene number in the block
7. Species1 gene rank list
8. Species1 gene name list
9. Species2 gene rank list
10. Species2 gene name list
11. Ks

Dotplot Pipeline

Inputs

Several input files must be provided for the Dotplot pipeline to run properly:

1. The fasta and gff file(s), properly formatted as described on page 2
2. The configuration file
3. The dotplot control file which contain the axis information as following:

#Axis	Species	Axis_label	Chromosomes_order
x axis	Osativa_cds	Oryza sativa	Chr1,Chr2,Chr3,Chr4
y axis	Sbicolor_cds	Sorghum bicolor	Chr1,Chr2,Chr3,Chr4

Configuration file

data_dir

Mandatory - String expected

The absolute path to the Working Directory containing all data files (".fasta", ".gff", ".blast", etc.)

tool_dir

Mandatory - String expected

The absolute path to the WGD_Tracker tool folder

intragenomic

Mandatory - Boolean expected: "True" or "False"

Specify whether this is an intragenomic (*e.g.* "True") or intergenomic (*e.g.* "False") analysis

SP1 and **SP2**

Mandatory - String expected

The name(s) of your compared genome(s) must be specified here (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You'll need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension

sp1_motif and **sp2_motif**

Mandatory - String expected

For each species specify the targeted motif of the third column in the gff file (*e.g.* "gene", "mRNA")

dotplot_control_file

Mandatory - String expected

Name of the file

dotplot_data

Mandatory - string expected: "RBBH", "Ks" or "Synteny"

This corresponds to the input file used to generate the dotplot, which can be the output file from the RBBH, the Ks or the synteny analyses.

name_corr

Optional - List of two values expected: "[string, string]"

Each string corresponds to the part of the chromosome id to remove in SP1 (1st value) and SP2 (2nd value)

out

Optional - string expected (default: “dotplot”)
Name of the figure output (no extention needed)

Outputs

The output files of this pipeline will be located in the Dotplot folder. There are three output files: **dotplot_dataset_file.txt** and **dotplot_formatting_file.txt** correspond to the dataset and the various plot formatting parameters used to generate the figure **dotplot.png**.

Karyotype Pipeline

Inputs

Several input files must be provided for the Karyotype pipeline (dedicated to interspecific comparative analyses) to run properly:

1. The fasta and gff file(s), properly formatted as described on page 2
2. The configuration file
3. The karyotype control file:

#Species	**infos**	**Chromosomes_order**
Osativa_cds	order	[Chr1,Chr2,Chr3,Chr4]
Sbicolor_cds	order	[Chr1,Chr2,Chr3,Chr4]
#Which species chromosomes are used to color the karyotype		
target_color	Osativa_cds	

Configuration file

data_dir

Mandatory - String expected

The absolute path to the Working Directory containing all data files (".fasta", ".gff", ".blast", etc.)

tool_dir

Mandatory - String expected

The absolute path to the WGD_Tracker tool folder

intragenomic

Mandatory - Boolean expected: "True" or "False"

Specify whether this is an intragenomic (*e.g.* "True") or intergenomic (*e.g.* "False") analysis

SP1 and **SP2**

Mandatory - String expected

The name(s) of your compared genome(s) must be specified here (*e.g.* **SP1**="Osativa_cds"; **SP2**="Sbicolor_cds"; first the short species name (*e.g.* "Osativa" and not "Oryza_sativa") and second the data type: cds, genomic or transcript). You'll need to use the same names for your data files. Thus, for each species, your different data files will be distinguished only by the extension

sp1_motif and **sp2_motif**

Mandatory - String expected

For each species specify the targeted motif of the third column in the gff file (*e.g.* "gene", "mRNA")

karyotype_control_file

Mandatory - String expected

Name of the file

name_corr

Optional - List of two values expected: "[string, string]"

Each string corresponds to the part of the chromosome name to remove in SP1 (1st value) and SP2 (2nd value)

corr_SB

Optional - List of six values expected: "[SP1, string, integer, SP2, string, integer]"

1st and 4th values correspond to species (SP1 or SP2)

2nd and 5th values correspond a pattern to target (*e.g.* "." or "-")
3rd and 6th values correspond to the number of elements to be deleted.

corr_size

Optional - Integer expected
This allows scaling down the chromosome sizes in the figure to fit within the defined dimensions of the output figure.

out

Optional - String expected (default: "Karyotype")
Name of the figure output (no extention needed).

Outputs

The output file of this pipeline is a .svg file (**Karyotype.svg**) that will be located in the Karyotype folder.