

# WGD\_Tracker

## Summary

Introduction .....	1
Organizing folders and files .....	2
Installation .....	4
Usage .....	5
RBBH Pipeline .....	6
Inputs .....	6
Configuration file .....	6
Outputs .....	9
Ks Pipeline .....	11
Inputs .....	11
Configuration file .....	11
Outputs .....	13
Synteny Pipeline .....	14
Inputs .....	14
Configuration file .....	14
Outputs .....	15
Dotplot Pipeline .....	16
Inputs .....	16
Configuration file .....	16
Outputs .....	17
Karyotype Pipeline .....	18
Inputs .....	18
Configuration file .....	18
Outputs .....	19

# Introduction

WGD\_Tracker is a tool designed to ease (intra- and inter-)genomic comparisons for detecting whole genome duplication events and to provide dating estimations by Ks analyses. This tool is fully customizable, allowing users to select stringent or flexible parameters depending on their specific analytical objectives.

This tool comprises 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 can 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 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 graphical representation of the two analyzed genomes karyotypes, illustrating the detected syntenic blocks 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 your 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'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 (e.g. Osativa\_cds.fasta, Osativa\_cds.gff, Osativa\_cds.cds).

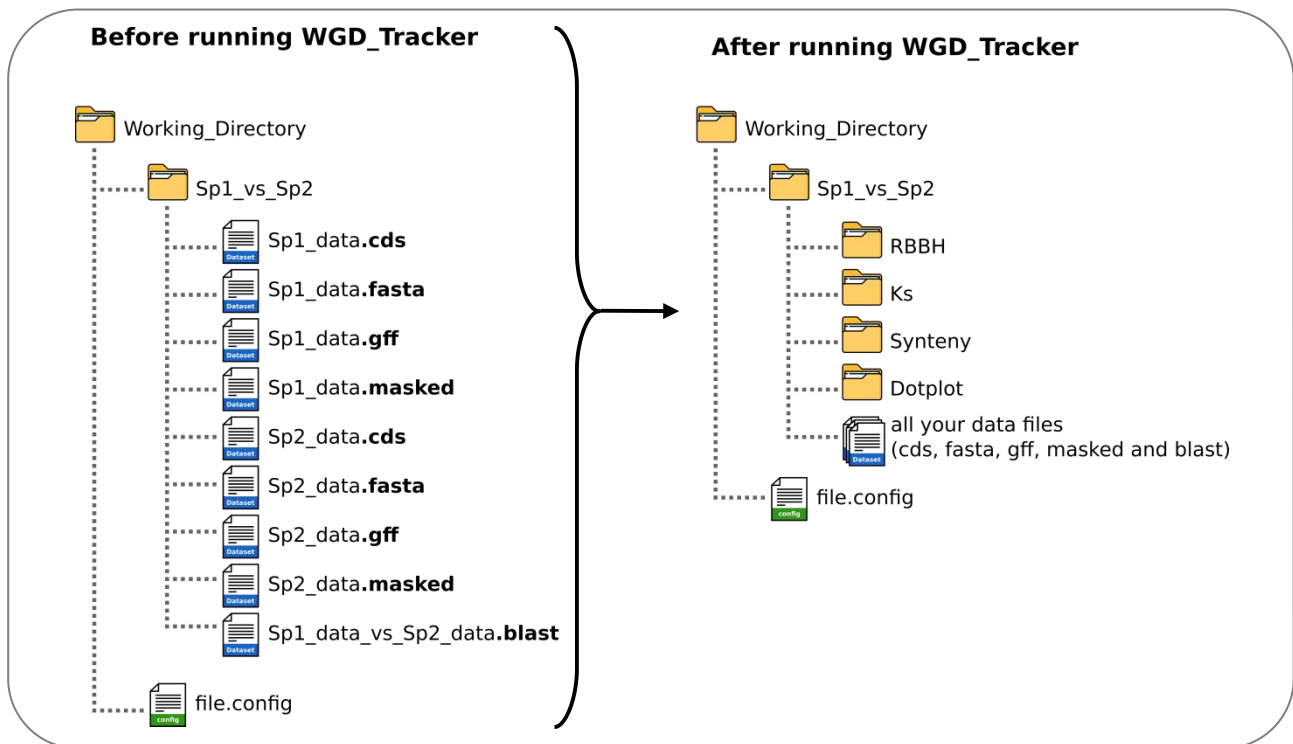
The extensions to use are:

- **.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

```
##gff-version 3
##annot-version v7.0
##species Oryza sativa
Chr1    phytozomev11    gene    2903    10817    .    +    .    Name=LOC_Os01g01010
Chr1    phytozomev11    mRNA    2903    10817    .    +    .    Name=LOC_Os01g01010.1
Chr1    phytozomev11    mRNA    2984    10562    .    +    .    Name=LOC_Os01g01010.2
Chr1    phytozomev11    gene    11218   12435    .    +    .    Name=LOC_Os01g01019
Chr1    phytozomev11    mRNA    11218   12435    .    +    .    Name=LOC_Os01g01019.1
Chr1    phytozomev11    gene    12648   15915    .    +    .    Name=LOC_Os01g01030
Chr1    phytozomev11    mRNA    12648   15915    .    +    .    Name=LOC_Os01g01030.1
Chr1    phytozomev11    gene    16292   20323    .    +    .    Name=LOC_Os01g01040
Chr1    phytozomev11    mRNA    16292   20323    .    +    .    Name=LOC_Os01g01040.1
Chr1    phytozomev11    mRNA    16321   20323    .    +    .    Name=LOC_Os01g01040.2
Chr1    phytozomev11    mRNA    16321   20323    .    +    .    Name=LOC_Os01g01040.3
Chr1    phytozomev11    mRNA    16292   18304    .    +    .    Name=LOC_Os01g01040.4
```

↑  
make sure that the first column of the gff  
has the same name as your ".fasta" file

↑  
The last column of the gff must have  
only one information as presented here



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 does not require any installation; a simple git clone is enough. However, WGD\_Tracker does **require dependencies**: Python3, Java, PAML, MACSE, R and Parallel. However, if you only want to use some of the tool's pipelines, not all dependencies are necessary for them to work properly. The following table shows the recommended dependencies version and which dependencies are required for each pipeline:

	Recommended version	RBBH	Ks	Synteny	Dotplot	Karyotype
Python	v3.9	X	X	X	X	X
Java	v1.8.0		X			
PAML	v4.9		X			
MACSE	v2.05		X			
R	v4.4		X			
Parallel	20190122	X	X			

## Retrieve WGD\_Tracker

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

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

```
$ cd WGD_Tracker/WGD_Tracker/
$ conda env create -p ./dependencies_conda_env -f wgd_tracker_dependencies.yml
$ conda activate dependencies_conda_env
$ R
$ install.packages("tinytex")
$ tinytex::install_tinytex()
$ q()
$ conda deactivate
```

# 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

1<sup>st</sup> value corresponds to the minimum percentage alignment requested (integer expected: within 0 and 100)

2<sup>nd</sup> 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 <i>versus</i> Species2 Genomic	len_ratio="[%alignment, 'simple']"
Species1 CDS <i>versus</i> Species2 Transcript	len_ratio="[%alignment, 'double']"
Species1 CDS <i>versus</i> Species2 CDS	len_ratio="[%alignment, 'double']"
Species1 CDS <i>versus</i> Species1 CDS ( <i>i.e.</i> intragenomic)	len_ratio="[%alignment, 'double']"

## corr\_intra

**Optional** - List of two values expected (default: not used)

1<sup>st</sup> value corresponds to a pattern to target (*e.g.* "." or "-")

2<sup>nd</sup> 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)

1<sup>st</sup> value corresponds to the Species name (*e.g.* Osativa)



2<sup>nd</sup> value corresponds to the targeted motif of the third column in the gff file (e.g. “gene”, “mRNA”)  
 3<sup>rd</sup> 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 <i>versus</i> Species2 CDS	coding="True" coding_type="simple" SP1_coding_infos="['Species1','mRNA',60]"
Species1 Genomic <i>versus</i> Species2 Transcript	
Species1 Genomic <i>versus</i> Species2 Genomic	coding="True" coding_type="double" SP1_coding_infos="['Species1','mRNA',60]" SP2_coding_infos="['Species2','mRNA',60]"  <b>If you only want to check if Species2 alignment are in coding region:</b> coding="True" coding_type="simple" SP1_coding_infos="['Species2','mRNA',60]"  <b>If Intragenomic:</b> 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)

1<sup>st</sup> value corresponds to the Species name (e.g. Osativa)

2<sup>nd</sup> value corresponds to the targeted motif of the third column in the gff file (e.g. “gene”, “mRNA”)

3<sup>rd</sup> 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

## Inputs

Several input files must be provided for the RBBH 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 splitted into different folders to avoid serveur 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 RBBH 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")

### 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 RBBH 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 informations 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 correspond to the part of the chromosome name to remove in SP1 (1<sup>st</sup> value) and SP2 (2<sup>nd</sup> value)

**out**

**Optional** - string expected (default: "dotplot")

Name of the figure output (no extension 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 RBBH pipeline 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 correspond to the part of the chromosome name to remove in SP1 (1<sup>st</sup> value) and SP2 (2<sup>nd</sup> value)

### corr\_SB

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

1<sup>st</sup> and 4<sup>th</sup> values corresponds to species (SP1 or SP2)

2<sup>nd</sup> and 5<sup>th</sup> values corresponds a pattern to target (e.g. "." or "-")

3<sup>rd</sup> and 6<sup>th</sup> values corresponds 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 extension needed).

## Outputs

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