Purpose of scaffremodler

Scaffremodler regroup a severals programs which principal aims is to detect use paired reads to link different genomic regions. These main programs are accompanied with a severals others programs that can be used in complement to improve scaffold assemblies or to detect large structural variations between a reference sequence and a re-sequenced genome.

Installation

All proposed tools described here are written in python and work on linux system
To install the tools:
1- unzip the folder
2- go to the bin directory and open the loca_programs.conf file
3- set the path to each programs required (listed below)

To run fully all programs are needed:
1- bowtie can be found at http://bowtie-bio.sourceforge.net/index.shtml
2- bowtie2 can be found at http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
3- SortSam.jar, MarkDuplicates.jar and FilterSamReads.jar belong to Picard Tools and can be found at http://broadinstitute.github.io/picard/
4- bwa can be found at http://bio-bwa.sourceforge.net
5- samtools can be found at http://www.htslib.org
6- circos-0.67-7 is required and can be found at http://circos.ca/software/download/circos/
perl, python and java are required. Biopython is also required.
7- bamgrepreads can be found at https://code.google.com/p/variationtoolkit/wiki/BamGrepReads

How to cite

Please cite either:

Martin G, Baurens F-C, Droc G, Rouard M, Cenci A, Kilian A, Hastie A, Doležel J, Aury J-M, Alberti A, et al. 2016. Improvement of the banana “Musa acuminata” reference sequence using NGS data and semi-automated bioinformatics methods. BMC Genomics 17:1–12.

Guillaume Martin, Françoise Carreel, Olivier Coriton, Catherine Hervouet, Céline Cardi, Paco Derouault, Danièle Roques, Frédéric Salmon, Mathieu Rouard, Julie Sardos, Karine Labadie, Franc-Christophe Baurens, Angélique D’Hont. Evolution of the banana genome (Musa acuminata) is impacted by large chromosomal translocations**.** *Molecular Biology and Evolution*, 2017; DOI: 10.1093/molbev/msx164

Scaffremodler descriptions

The package provided comprise 21 programs listed here:
1_create_conf.py
2_map.py
3_filter_single_pair.py
4_filter_sam.py
5_calc_stat.py
6_parse_discord.py
7_select_on_cov.py
8_ident_SV.py
scaffremodler_wrapper.py
SplitOnX.py
conf4circos.py
contig_scaff.py
convert2X.py
draw_circos.py
fusion_scaff.py

group4contig.py
look4fusion.py
reEstimateN.py
verif_fusion.py
filter_common_zones.py
drawDestZones.py

All 21 programs run using the following command: python program-name <--options-name value>

Tools descriptions

- **1_create_conf.py**

This program takes in input a multifasta file and output a tabulated file recording sequence length informations. Each line corresponds to a sequence in the multicast file and contain two columns: the column correspond to sequence name and the second the sequence size.
In addition this program generate a configuration file that will be used by all other programs that begin with a number. These programs can run without this configuration file but as some options are common between these programs, generating the configuration allows passing options only once. These options and their utility are described in their respective programs. When a configuration file is passed to other programs, these programs add informations in it.

In addition to these parameters the following options should be filled:
--ref : A multi-fasta containing the reference sequence.
--chr : The name of the output file that will contain the informations on sequence length.
output file description : tabulated file containing two columns and one line for each sequences contained in the multi-fasts file. Column 1 contains sequence name, column 2 contains sequence length.
--output : The name of the output configuration file that will be generated.
--restimate : in 6_parse_discord.py paired reads are parsed based on their mapping orientation and insert size. The minimal and maximal correct insert size is re-estimated in 5_calc_stat.py. Minimal and maximal insert size are calculated by adding and subtracting, respectively, X*standard deviation to the median insert size calculated on first identified well mapped reads. As median insert size, insert size standard deviation is estimated on identified well mapped reads. The X value is provided in the **msd** option.
y : use re-estimated minimal and maximal insert size for parsing of well mapped reads
n : don't use re-estimated minimal and maximal insert size for parsing of well mapped reads (default)
If the insert size has not a normal like distribution, it is not recommended to re-estimate minimal and maximal insert size.
--msd : multiplicator of standard deviation to re-estimate minimal and maximal insert size to identify well mapped reads. This parameter is used if **restimate** option is set to 'y'. (Default: 3)

- **2_map.py**

This program aligns paired reads along sequences recorded in a multifasta file. Read pairs should be in fastq format with either phred33 or phred64 quality encoding.

Options:
--**tool** : This option select the mapping tool used to perform the alignment of paired reads. Depending on the tool used, alignment parameters are different. The possible argument are:
bowtie: align each mate of a pair independently using bowtie. Only single mapped reads are reported (correspond to -a -m 1 parameters for bowtie). At the end of mapping process read pairs are reconstructed.

bowtie2_single: align each mate of a pair independently using bowtie2 with the end-to-end and very-sensitive bowtie2 options. At the end of mapping process read pairs are reconstructed. (default)
bowtie2: align paired reads using bowtie2 with end-to-end and very-sensitive bowtie2 options.
bwa: align paired reads using bwa aln algorithm with default parameters.
bwa_mem: align paired reads using bwa mem algorithm with default parameters.
--ref : path to multifasta containing the reference sequence
--q1 : path to mate1 read fastq file
--q2 : path to mate2 read fastq file
--orient : expected read orientation
rf : reverse-forward reads (default)
fr : forward-reverse reads
--mini : minimal read insert size to consider read pair to be well mapped. (Default: 2500)
--maxi : maximal read insert size to consider read pair to be well mapped (default 7500)
--qual : quality encoding of the fast files
33 : phred33 quality encoding
64 : phred64 quality encoding
--index : This options decide if reference index should be built. The reference index is built in the folder that contain the reference. Index have the same name of the reference sequence with additional extensions.
y : build reference index (default)
n : don't build reference index
--rmindex :
y : remove reference index at the end of the process (default)
n : don't remove reference index at the end of the process
--thread : number of thread to use for mapping step. (Default: 1)
--out : output sam file name that will contain reads pairs aligned to the reference (default : mate.sam).
--config : configuration file generated by 1_create_conf.py. If a config file is passed, all other options except --out will be ignored.

- **3_filter_single_pair.py**

This program filter paired reads in a sam file sorted by query name. The filtering can be either done on the mapping quality and/or based on a threshold between AS/XS flags. This program output a filtered sam file sorted by query name. Unmapped and single end mapped pairs are not removed during this step.

Options:
--sam : path to the input sam file (The file should be sorted on query name).
--asxs : an integer corresponding to the minimal difference between the AS/XS flag value to keep a pair. A pair is kept if both mate pass threshold.
--qual : an integer corresponding to the minimal mapping quality. If both mate of a pair have a mapping quality superior or equal to the asxs parameter, the pair is kept.
--rminput : This options decide if input should be deleted after treatment.
y : remove input
n : don't remove input (default)
--out : output filtered sam file name (default : Single_hit_mapped.sam).
--config : configuration file generated by 1_create_conf.py. If a config file is passed, all other options except --out will be ignored.

- **4_filter_sam.py**

This program removes unmapped and single mapped paired reads in a sam/bam file.

Options:

--sam : path to the input sam/bam file.
--type : input type.
sam : it is a sam file (default)
bam : it is a bam file
--sort : sort order of the output file.
queryname : read are sorted based on read name in the output file
coordinate : read are sorted based on their coordinate in the output file (default)
--rminput : This options decide if input should be deleted after treatment.
y : remove input
n : don't remove input (default)
--out : output filtered sam/bam file name (default : rmdup_mapped.bam). The extension of the output file name (.sam or .bam) decides of the output file format.
--config : configuration file generated by 1_create_conf.py. If a config file is passed, all other options except --out will be ignored.

- **5_calc_stat.py**

This program calculate coverage for each covered sites of the reference sequences, estimate mean, median and 90 confidence interval coverage for the covered sites (uncovered sites are not taken in account). This program also re-estimate insert size by calculating median insert size of correctly mapped reads. It also calculates insert size standard deviation.

Options:
--sam : path to the input sam/bam.
--type : input type.
sam : it is a sam file (default)
bam : it is a bam file
--out : output coverage file name. (Default: coverage.cov)
output file description : The output file is a tabulated file composed of 3 columns. Column 1: sequence name, column 2: position, column 3: coverage at the position.
--stat : output file name where coverage and insert size will be recorded. (Default: stat.txt)
--outconf : **deprecated option**
--config : configuration file generated by 1_create_conf.py. If a config file is passed, all other options except output files will be ignored.
Minimal and maximal correct insert size are re-estimated and written in 'Calc_coverage' section of the configuration file under 'mini' and 'maxi' labels respectively. The mean, median and insert size standard deviation are also recorded in 'Calc_coverage' section under 'mean_insert', 'median_insert' and 'standard_deviation_insert' labels respectively. Mean and median coverage are also recorded in this section under 'mean_coverage' and 'median_coverage' labels respectively.

- **6_parse_discord.py**

This program takes in input a sam/bam file, identify discordant read pairs, calculate proportion of discordant reads on 1kb window size and parse the sam/bam file in 11 sub bam files corresponding to the different discordant types of mapped pairs :
1- correct orientation and insert size : reverse-forward or forward-reverse, depending on correct orientation (**out_rf** or **out_fr** options respectively)
2- correct orientation but insert size > expected (deletion type, **out_del** option)
3- correct orientation but insert size < expected (insertion type, **out_ins** option)
4- uncorrect orientation : reverse-forward or forward-reverse depending on correct orientation (**out_rf** or **out_fr** option respectively)
5- reverse-reverse mapped pairs on the same chromosome (**out_rr** option)
6- forward-forward mapped pairs on the same chromosome (**out_ff** options)
7- reverse-forward mapped pairs on distinct chromosomes (**out_chr_rf** option)

8- forward-reverse mapped pairs on distinct chromosomes (**out_chr_fr** option)
9- reverse-reverse mapped pairs on distinct chromosomes (**out_chr_rr** option)
10- forward-forward mapped pairs on distinct chromosomes (**out_chr_ff** option)
11- an additional bam file containing discarded reads. These discarded read pairs are incorrectly mapped reads that have an insert size lower than the minimal insert size passed in **mini_dis** options.

Options:

--sam : path to the input sam/bam.

--chr : path to a tabulated file containing reference sequence informations. The file is a two column file with sequence name in column 1 and sequence length in column 2. This is the file generated by 1_create_conf.py under **chr** option.

--sort : sort order of the input file.
queryname : read are sorted based on read name
coordinate : read are sorted based on their coordinate
unsorted : read are not sorted (default)

--type : input type.
sam : it is a sam file (default)
bam : it is a bam file

--rminput : This options decide if input should be deleted after treatment.
y : remove input
n : don't remove input (default)

--orient : expected read orientation
rf : reverse-forward reads (default)
fr : forward-reverse reads

--mini_dis : Minimal insert size to keep the discordant read pair for parsing (integer). Only discordant read pairs mapping on the same chromosome that are not of insertion type are filtered with this parameter. (default : 10000, integer)

--mini : minimal insert size to consider well mapped read pair (integer).

--maxi : maximal insert size to consider well mapped read pair (integer).

--out_ins : Output bam file name for insertion discordant read type (default : discord_ins.bam)

--out_del : Output bam file name for deletion discordant read type (default : discord_del.bam)

--out_fr : Output bam file name for forward-reverse mapped read pairs (default : discord_fr.bam)

--out_rf : Output bam file name for reverse-forward mapped read pairs (default : discord_rf.bam)

--out_ff : Output bam file name for forward-forward mapped read pairs (default : discord_ff.bam)

--out_rr : Output bam file name for reverse-reverse mapped read pairs (default : discord_rr.bam)

--out_chr_fr : Output bam file name for forward-reverse mapped read pairs mapping on different chromosomes (default : discord_chr_fr.bam)

--out_chr_rf : Output bam file name for reverse-forward mapped read pairs mapping on different chromosomes (default : discord_chr_rf.bam)

--out_chr_ff : Output bam file name for forward-forward mapped read pairs mapping on different chromosomes (default : discord_chr_fr.bam)

--out_chr_rr : Output bam file name for reverse-reverse mapped read pairs mapping on different chromosomes (default : discord_chr_rr.bam)

--out_discarded : Output bam file name for paired read discarded by **mini_dis** parameter (default : discarded.bam)

--liste_type : Output name of file listing parsed read pair. (Default: liste_type.txt)
output file description :

--discord_prop : Output name of discordant read proportion file. (Default: discord_prop.txt)
output file description :

--exclude_chrom : Exclude chromosomes from analysis: "no_exclude" or chromosomes names separated by "=". (Default: no_exclude)

--config : configuration file generated by 1_create_conf.py. If a config file is passed, all other options except output files will be ignored.

- **7_select_on_cov.py**

This program use discordant reads in a sam/bam file to detect non-contiguous linked zones. Sam/bam file provided should only contain reads pairs that are of the same discordance type. As repeat sequences and sequence divergence between the reference and the accession re-sequenced can cause the identification of false linkage additional filtering steps are performed.
The identification of non-contiguous linked zones can be described in 5 steps:

Step1: Genome coverage for one discordance type is calculated from the provided sam/bam file.

Step2: Discordant zones are identified and defined if this zone contain at least X covered sites with no more than Y contiguous covered sites. X and Y parameters are specified with **min_zone** and **min_gap** parameters respectively. The zone is filtered out if the median coverage of covered sites exceeds or is lower than specified threshold specified in **min_cov** and **max_cov** parameters respectively. Schematization of identification of discordant zones is found on figure 1.

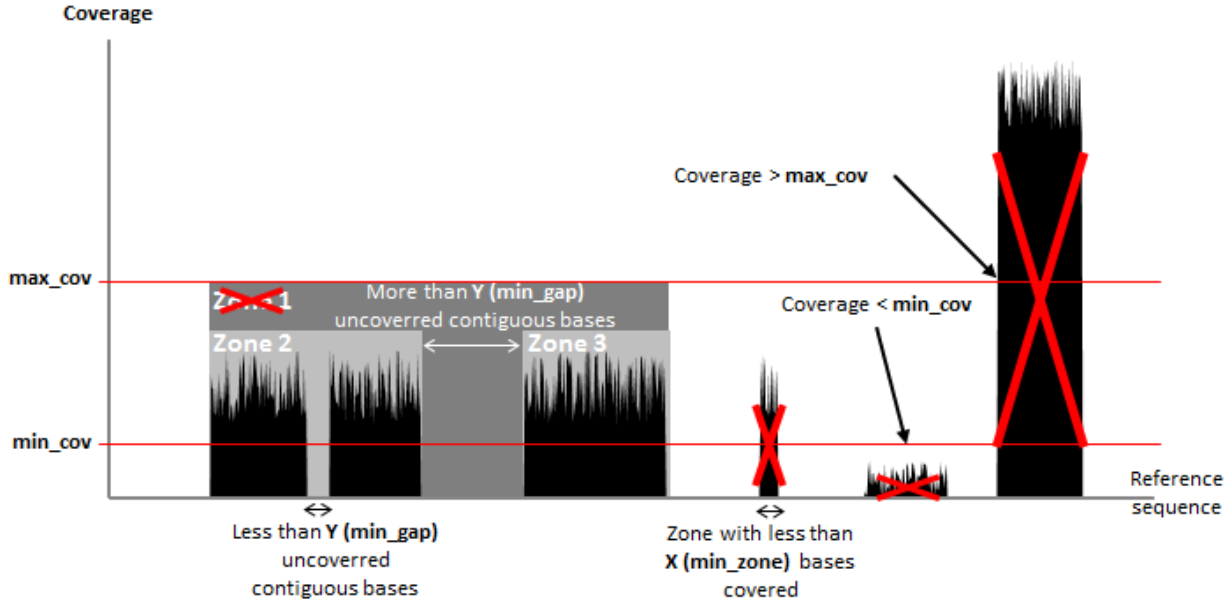

Figure 1 : Schematization of identification of discordant zones.

Step3: The **destination zone** of identified discordant zone in step2 is calculated. For each discordant zone the discordant paired reads locating in the zones are used to define the destination zone. As a discordant zone can have several destinations, the destinations are defined using an iterative process to attribute reads do each **destination zone**. The iterative step consist in taking a reads destination and adding another read if it locate in a zone around the destination of the first read plus or minus a defined value **W**. For second and following iteration reads are added if they locate in a zone around the median destination of already grouped read plus or minus a defined value **W**. The **W** value is equal to the size of the **discordant zone** plus value provided in **ecart** options. If the configuration file is provided this value is automatically calculated as the 3 * (estimated standard deviation of insert size). If no more reads can be attributed to the defined destination zone, discordant zone and corresponding destination zone coverages are recalculated and filtered with the **min_cov**, **max_cov** and **min_zone** parameters. The iterative step is repeated for each discordant zone identified in step 2 until all reads have been used. Schematization of identification of destination zones is found on figure 2.

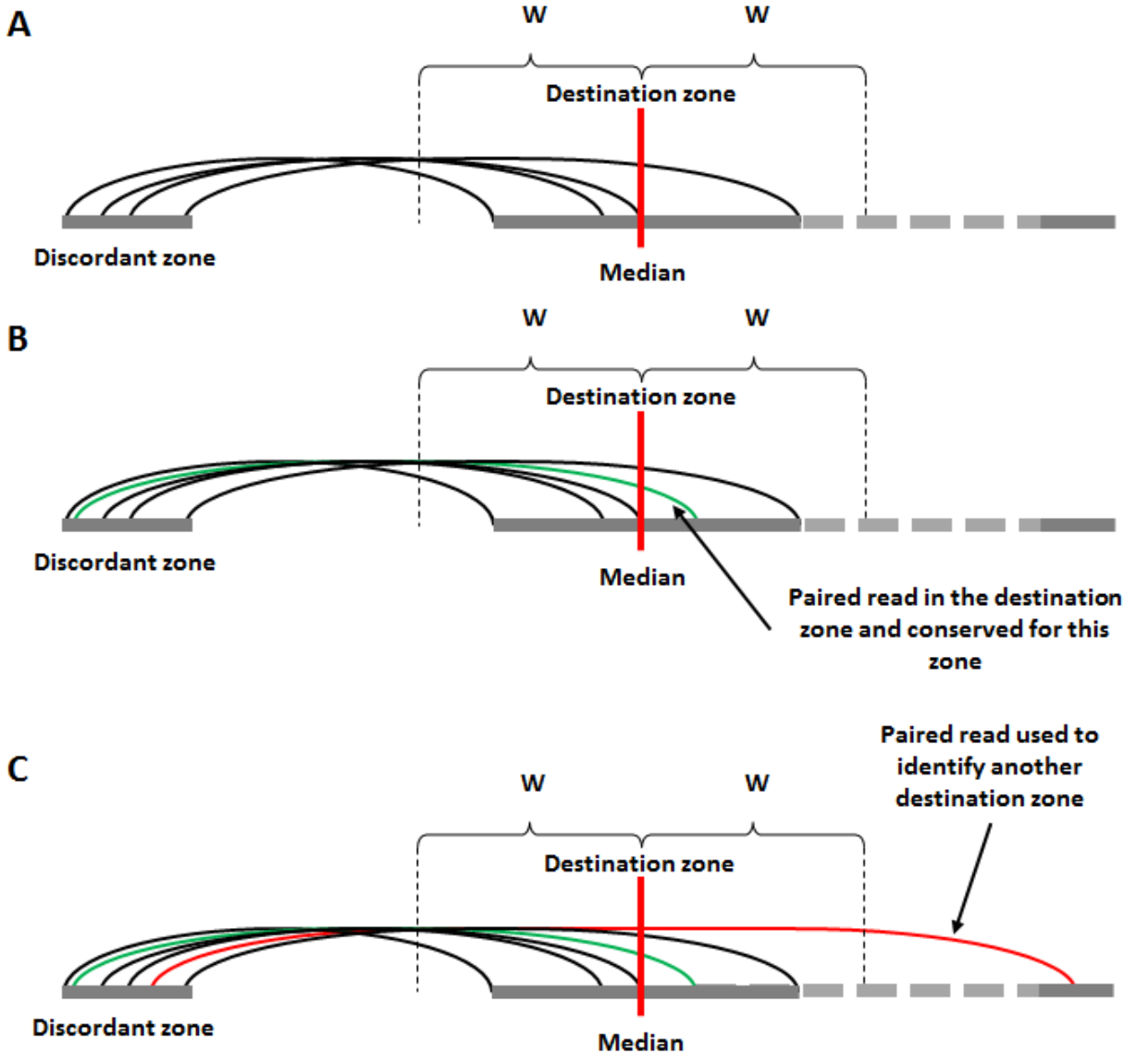

Figure 2 : Schematization of identification of destination zones. (**A)** Re-calculation of median position of destination zone. (**B**) Identification of paired read aligning in both discordant and destination zones and attribution of this reads to this destination zone. This read is used to re-estimate median destination zone for the following iteration. (**C**) Identification of paired read aligning in discordant zone but not destination zone. This read is attributed to this destination zone but will be used to look for another destination zone.

Step4: Discordant zones and there destination are merged if they are similar. Two discordant zones and destination are considered similar if both destinations and discordant zones are closer than a value specified in **max_dist_merge** parameter.

Step5: Identified linked zone are scored with the following formula:

$$Score = 100 * f(median\ coverage) * g(discordant\ zone\ size)$$

$$f(x) = \begin{cases} \frac{1}{a} * x & (si\ x < a) \\ 1 & (si\ x \geq a) \end{cases}$$

The score value is comprised between 0 and 100. Linked zones are written in a file provided in out option and linked zone having a score equal or higher to the value specified in min_score options are identified in the output file with the PASSED value in column 14.

This program should be launched for each discordant sam/bam file.

Options:

--ref : path to multifasta containing the reference sequence.
--sam : path to the input sam/bam.
--type : input type.
sam : it is a sam file (default)
bam : it is a bam file
--min_zone : Minimal number of covered sites in the zone. (Default: 500)
--maxcov : Maximal median coverage accepted in a zone. (Default: 300)
--mincov : minimal median coverage accepted in a zone. (Default: 0)
--min_gap : maximal distance of contiguous uncovered sites in a zone. (Default: 300)
--ecart : value that will be added or subtracted to keep a read as the same destination. Recommended 3*sd(insert). (Default: 2000)
--chr : path to a tabulated file containing reference sequence informations. The file is a two column file with sequence name in column 1 and sequence length in column 2. This is the file generated by 1_create_conf.py under **chr** option.
--max_dist_merge : Maximal distance between two discordant zone to merge. (Default: 1000)
--YiS : Y-intercept of the linear function for zone size that will give the first component of product giving the score. Do not put negative value. (Default: 0)
--MiS : Minimal zone size for which the first component of product giving the score will be maximal. (Default: 1000)
--YiC : Y-intercept of the linear function for coverage that will give the second component of product giving the score. Do not put negative value. (Default: 0)
--MiC : Minimal zone coverage for which the second component of product giving the score will be maximal. (Default: 25)
--min_score : Minimal score for a discordant zone to be identified as passed. (Default: 70)
--out : Output file listing linked discordant zones.
output file description : The output file is a tabulated file containing in each line a linked discordant zone. In column 1: chromosome discordant zone 1, column 2: start zone position, column 3: end zone position, column 4: zone size, column 5: zone coverage, column 6: chromosome discordant zone 2, column 7: start zone position, column 8: end zone position, column 9: zone size, column 10: zone coverage, column 11: misc, column 12: read number, column 13: score, column 14: status of discordant zone based on **min_score** option (PASSED or NOT_PASSED).
--config : Configuration file generated by 1_create_conf.py. If a config file is passed, all other options except **out** and **sam** options will be ignored.

- **8_ident_SV.py**

This program tries to identify among non-contiguous linked zones (identified with *7_select_on_cov.py*) signature of simple structural variations. Structural variations searched are insertions, deletions, inversions, duplications, reciprocal and non-reciprocal translocations in re-sequenced accession. For deletions and duplications, the coverage is also used to validate the structure.

Options:

--frf : Path to discordant 'forward-reverse' or 'reverse-forward' zones file depending on expected orientation.

--ff : Path to discordant 'forward-forward' zones file.

--rr : Path to discordant 'reverse-reverse' zones file.

--ins : Path to discordant 'insertion' zones file.
--delet : Path to discordant 'deletion' zones file.
--chr_rr : Path to discordant 'chromosome reverse-reverse' zones file.
--chr_fr : Path to discordant 'chromosome forward-reverse' zones file.
--chr_rf : Path to discordant 'chromosome reverse-forward' zones file.
--chr_ff : Path to discordant 'chromosome forward-forward' zones file.
--chr : Path to a tabulated file containing reference sequence informations. The file is a two column file with sequence name in column 1 and sequence length in column 2. This is the file generated by 1_create_conf.py under **chr** option.

--covf : Coverage file calculated in *5_calc_stat.py* corresponding to 'stat.txt' if default are set when running *5_calc_stat.py*.
--orient : Expected read orientation
rf : reverse-forward reads (default)
fr : forward-reverse reads
--insert : The expected insert size of paired reads (in number of bases). (Default: 5000)
--exp_cov : Float value corresponding to the expected coverage of the genome.
--ploid : This parameter helps identifying deletions and duplications depending on the ploidy level of the genome searched. (Default: 0.33)
Example: If homozygous duplication expected in diploid: expected = coverage + coverage*1.
(Ideally ploid value = 1 but set ploid value lower than 1)
If heterozygous duplication expected in diploid: expected = coverage + coverage*0.5. (Ideally ploid value = 0.5 but set ploid value lower than 0.5)
--thread : Number of thread to use (integer). Exceeding 9 will not improve calculation time (Default: 1)
--out : Output file containing structural variations detected. (Default: SV_detected.tab)
--config : Configuration file generated by 1_create_conf.py. If a config file is passed: **chr**, **covf**, **insert**, **exp_cov** and **ploid** options will be ignored.

- **scaffremodler_wrapper.py**

This program is a wrapper of the 8 precedent programs. Options are identical to previously described options.

Options:

--tool : This option select the mapping tool used to perform the alignment of paired reads. Depending on the tool used, alignment parameters are different. The possible arguments are:

bowtie: align each mate of a pair independently using bowtie. Only single mapped reads are reported (correspond to -a -m 1 parameters for bowtie). At the end of mapping process read pairs are reconstructed.

bowtie2_single: align each mate of a pair independently using bowtie2 with the end-to-end and very-sensitive bowtie2 options. At the end of mapping process read pairs are reconstructed. (default)

bowtie2: align paired reads using bowtie2 with end-to-end and very-sensitive bowtie2 options.

bwa: align paired reads using bwa aln algorithm with default parameters.
bwa_mem: align paired reads using bwa mem algorithm with default parameters.

--ref : Path to multifasta containing the reference sequence.

--q1 : Path to mate1 read fastq file

--q2 : Path to mate2 read fastq file

--orient : Expected read orientation

rf : reverse-forward reads (default)

fr : forward-reverse reads

--mini : Minimal read insert size to consider read pair to be well mapped. (Default: 2500)

--maxi : Maximal read insert size to consider read pair to be well mapped (default 7500)

--qual : Quality encoding of the fast files

33 : phred33 quality encoding

64 : phred64 quality encoding

--index : This options decide if reference index should be built. The reference index is built in the folder that contain the reference. Index have the same name of the reference sequence with additional extensions.

y : build reference index (default)

n : don't build reference index

--rmindex :

y : remove reference index at the end of the process (default)

n : don't remove reference index at the end of the process

--filter_multi : Filter reads with multiple locations .

y : yes (default)

n : no

--mini_dis : Minimal insert size to keep the discordant read pair for parsing (integer). Only discordant read pairs mapping on the same chromosome that are not of insertion type are filtered with this parameter. (Default: 10000, integer)

--mult_max_cov : Multiplicator of median coverage for maximal median coverage to keep a zone (float). The value of (**mult_max_cov** * mediane coverage) is equivalent to **maxcov** parameter in *7_select_on_cov.py* (Default: 10).

--mult_min_cov : Multiplicator of median coverage for minimal median coverage to keep a zone (float). The value of (**mult_min_cov** * mediane coverage) is equivalent to **mincov** parameter in *7_select_on_cov.py* (Default: 10).

--min_zone : Minimal number of covered sites in the zone. (Default: 500)

--min_gap : Maximal distance of contiguous uncovered sites in a zone. (Default: 300)

--thread : Number of thread to use for *2_map*, *7_select_on_cov*, *8_ident_SV*. (Default: 1)

--msd : multiplicator of standard deviation to re-estimate minimal and maximal insert size to identify well mapped reads. This parameter is used if **restimate** option is set to 'y'. (Default: 3)

--max_dist_merge : Maximal distance between two discordant zone to merge. (Default: 1000)

--YiS : Y-intercept of the linear function for zone size that will give the first component of product giving the score. Do not put negative value. (Default: 0)

--MiS : Minimal zone size for which the first component of product giving the score will be maximal. (Default: 1000)

--YiC : Y-intercept of the linear function for coverage that will give the second component of product giving the score. Do not put negative value. (Default: 0)

--MiC : Minimal zone coverage for which the second component of product giving the score will be maximal. (Default: 25)

--min_score : Minimal score for a discordant zone to be identified as passed. (Default: 70)

--ploid : This parameter helps identifying deletions and duplications depending on the ploidy level of the genome searched. (Default: 0.33)

Example: If homozygous duplication expected in diploid: expected = coverage + coverage*1.
(Ideally ploid value = 1 but set ploid value lower than 1)
If heterozygous duplication expected in diploid: expected = coverage + coverage*0.5. (Ideally ploid value = 0.5 but set ploid value lower than 0.5)

--restimate : In 6_parse_discord.py paired reads are parsed based on their mapping orientation and insert size. The minimal and maximal correct insert size is re-estimated in 5_calc_stat.py. Minimal and maximal insert size are calculated by adding and subtracting, respectively, X*standard

deviation to the median insert size calculated on first identified well mapped reads. As median insert size, insert size standard deviation is estimated on identified well mapped reads. The X value is provided in the **msd** option.

y : use re-estimated minimal and maximal insert size for parsing of well mapped reads

n : don't use re-estimated minimal and maximal insert size for parsing of well mapped reads (default)

If the insert size has not a normal like distribution, it is not recommended to re-estimate minimal and maximal insert size.

--rm_intermediate : Remove intermediate bam/sam.

y : yes (default)

n : no

--prefix : Prefix for all output files. (Default: apmap)

--exclude_chrom : Exclude chromosomes from analysis: "no_exclude" or chromosomes names separated by "=". (Default: no_exclude)

--step : Steps to perform. Concatenation of integers from 1 to 8 corresponding to the 8 programs presented above.

Example : --step 12 will run *1_create_conf* and *2_map*.

- **conf4circos.py**

This program takes in input all files needed to generate circos and output severals files that will be used to generate different circus pictures plus a config file.

Options:

--ref : A multi-fasta containing the reference sequence.

--chr : path to a tabulated file containing reference sequence informations. The file is a two column file with sequence name in column 1 and sequence length in column 2. This is the file generated by 1_create_conf.py under **chr** option.

--orient : expected read orientation

rf : reverse-forward reads (default)

fr : forward-reverse reads

--cov : Coverage file calculated in *5_calc_stat.py* corresponding to 'stat.txt' if default are set when running *5_calc_stat.py*.

--window : Window size (bases) for which mean coverage will be calculated and plotted in the circos representation. (Default: 1000)

--frf : Path to discordant 'forward-reverse' or 'reverse-forward' zones file depending on expected orientation.

--ff : Path to discordant 'forward-forward' zones file.

--rr : Path to discordant 'reverse-reverse' zones file.

--ins : Path to discordant 'insertion' zones file.

--delet : Path to discordant 'deletion' zones file.

--chr_rr : Path to discordant 'chromosome reverse-reverse' zones file.

--chr_rf : Path to discordant 'chromosome reverse-forward' zones file.

--chr_ff : Path to discordant 'chromosome forward-forward' zones file.

--chr_fr : Path to discordant 'chromosome forward-reverse' zones file.

--liste_read : Path to the .list (**--liste_type** option) file generated by *6_parse_discord.py*.

--dis_prop : Path to the .prop (**--discord_prop** option) file generated by *6_parse_discord.py*.

--agp : Path to an AGP file locating scaffolds along chromosomes.

--prefix : Prefix for all output files. If this options is passed, all others output options are ignored.

--out_kar : Karyotype output file name. (Default: circos_karyotype.txt)

--out_N : File name of text file locating N. (Default: circos_loc_N.txt)

--out_cov : Mean coverage output file. (Default: circos_mean.cov)

--out_frf : A link output file name corresponding to discordant ‘forward-reverse’ or ‘reverse-forward’ zones identified. (Default: circos_zone_frf.link)

--out_ff : A link output file name corresponding to discordant ‘forward-forward’ zones identified. (Default: circos_zone_ff.link)

--out_rr : A link output file name corresponding to discordant ‘reverse-reverse’ zones identified. (Default: circos_zone_rr.link)

--out_ins : A link output file name corresponding to discordant ‘insertion’ zones identified. (Default: circos_zone_ins.link)

--out_delet : A link output file name corresponding to discordant ‘deletion’ zones identified. (Default: circos_zone_delet.link)

--out_chr_rr : A link output file name corresponding to discordant ‘chromosome reverse-reverse’ zones identified. (Default: circos_zone_chr_rr.link)

--out_chr_rf : A link output file name corresponding to discordant ‘chromosome reverse-forward zones identified. (Default: circos_zone_chr_rf.link)

--out_chr_ff : A link output file name corresponding to discordant ‘chromosome forward-forward’ zones identified. (Default: circos_zone_chr_ff.link)

--out_chr_fr : A link output file name corresponding to discordant ‘chromosome ‘forward-reverse’ zones identified. (Default: circos_zone_chr_fr.link)

--Rout_rf : A link output file name corresponding to reverse-forward mapped reads. (Default: circos_read_rf.link)

--Rout_fr : A link output file name corresponding to forward-reverse mapped reads. (Default: circos_read_fr.link)

--Rout_ff : A link output file name corresponding to forward-forward mapped reads. (Default: circos_read_ff.link)

--Rout_rr : A link output file name corresponding to reverse-reverse mapped reads. (Default: circos_read_rr.link)

--Rout_ins : A link output file name corresponding to discordant of insertion mapped reads. (Default: circos_read_ins.link)

--Rout_delet : A link output file name corresponding to discordant of deletion mapped reads. (Default: circos_read_delet.link)

--Rout_chr_rr : A link output file name corresponding to chromosome reverse-reverse mapped reads. (Default: circos_read_chr_rr.link)

--Rout_chr_rf : A link output file name corresponding to chromosome reverse-forward mapped reads. (Default: circos_read_chr_rf.link)

--Rout_chr_ff : A link output file name corresponding to chromosome forward-forward mapped reads. (Default: circos_read_chr_ff.link)

--Rout_chr_fr : A link output file name corresponding to chromosome forward-reverse mapped reads. (Default: circos_read_chr_fr.link)

--out_scaff : A tile output file name corresponding to scaffolds located along the reference sequence. (Default: circos_scaffold.tile)

--output : Output name of the configuration file that will be passed to draw the circos picture. (Default: config_circos.conf)

--filterdraw : The filter to draw zone. Possible values : P : passed, NP : not_passed, N : New zone identified by reads only.

--removedZones : a file containing a list of score file. All the zones of these score file will be excluded of the analysis. (One line by score file : “discType = path/to/tho/score_file”)

--nbaccess : if you have run the filter_common_zones.py script, you can draw only zone present in a certain number of accession. Example : --nbaccess 6-8 to draw only zones presents in 6,7 or 8 accessions.

--filterzone : The filter to define a similar zone. Possible values : P : passed, NP : not_passed, N : New zone identified by reads only

- **draw_circos.py**

This program takes in input a configuration file and generates a circos picture (png format) representing several layers summarizing paired read mapping information.
All drawn links are color coded based on discordant type:

- Concordant read : grey
- Inter chromosomal discordance:

Reverse-forward : red
Forward-reverse : blue
Forward-forward : green
Reverse-reverse : purple

- Intra chromosomal discordance:

Insertion : red
Deletion : orange
Reverse-reverse : purple
Forward-forward : green

If expected orientation is reverse-forward :
Forward-reverse : blue
If expected orientation is forward-reverse:
Reverse-forward : blue

Options:

--config : The conf file generated by conf4circos.py
--draw : A list of chromosome position separated with = (ex:chr01=1000=20000). (Default: all)
--cov : Draw coverage plot.
y : yes (default)
n : no
--scaff : Draw scaffold position.
y : yes (default)
n : no
--discord : Draw discordant proportion plot.
y : yes (default)
n : no
--frf : Draw discordant 'forward-reverse' or 'reverse-forward' zones.
y : yes (default)
n : no
--ff : Draw discordant 'forward-forward' zones.
y : yes (default)
n : no
--rr : Draw discordant 'reverse-reverse' zones.
y : yes (default)
n : no
--ins : Draw discordant insertion zones.
y : yes (default)
n : no
--delet : Draw discordant deletion zones.
y : yes (default)
n : no
--chr_rr : Draw discordant chromosome 'reverse-reverse' zones.
y : yes (default)
n : no
--chr_rf : Draw discordant chromosome 'reverse-forward' zones.
y : yes (default)
n : no

--chr_fr : Draw discordant chromosome ‘forward-reverse’ zones.
y : yes (default)
n : no
--chr_ff : Draw discordant chromosome ‘forward-forward’ zones.
y : yes (default)
n : no
--read_fr : Draw links corresponding to forward-reverse mapped reads.
y : yes (default)
n : no
--read_rf : Draw links corresponding to reverse-forward mapped reads.
y : yes (default)
n : no
--read_ff : Draw read link corresponding to forward-forward mapped reads.
y : yes (default)
n : no
--read_rr : Draw read link corresponding to reverse-reverse mapped reads.
y : yes (default)
n : no
--read_ins : Draw read link corresponding to discordance of insertion mapped reads.
y : yes (default)
n : no
--read_delet : Draw read link corresponding to discordance of deletion mapped reads.
y : yes (default)
n : no
--read_chr_rr : Draw read link corresponding to chromosome reverse-reverse mapped reads.
y : yes (default)
n : no
--read_chr_rf : Draw read link corresponding to chromosome reverse-forward mapped reads.
y : yes (default)
n : no
--read_chr_fr : Draw read link corresponding to chromosome forward-reverse mapped reads.
y : yes (default)
n : no
--read_chr_ff : Draw read link corresponding to chromosome forward-forward mapped reads.
y : yes (default)
n : no
--text : Locate N regions (unknown regions with positions) in a text layer.
y : yes (default)
n : no
--out : The output name of the picture.

- **convert2X.py**

This program replaces specified regions in the provided table file by “X”. These “X” will be used to split scaffold using SplitOnX.py

Options:

--table : A table file with region to convert to “X”.

File description : A tabulated file with in column 1 : scaffold name, column 2 : start position and column 3 : end position if a zone is concerned and nothing if only a site.

--fasta : A multifasta file containing sequences.

--out : Output file name of the multifasta file containing specified regions replaced by “X”. (Default: X_converted.fasta)

- **SplitOnX.py**

This program split DNA sequence when “X” are found. All scaffolds are renamed by length.

Options:
--fasta : A multifasta file containing sequences to split.
--out : Output file name. (Default: Splitted.fasta)

- **look4fusion.py**

This script looks for possible scaffold fusions and junctions based on discordant zones detected and unknown regions in a reference genome sequence.

Options:
--config : The configuration file generated by *conf4circos.py*
--bound : Boundaries of scaffold to look for fusion and junction. Only scaffold extremities are searched for fusion and junction. This means that no partial scaffold fusion and junction are searched. (Default: 10 000)
--out : Output text file containing possible fusions and junctions. (Default: possible_fusion.txt)
--out_tar : Output name of a tar.gz file containing circos figures showing discordant zone leading to the detection of possible fusions and junctions. (Default: possible_fusion.tar)

- **group4contig.py**

This program takes scaffold name to join provided in a table file and group them by linkage. Scaffold groups are outputted in a table file. This file should be edited to be used by *contig_scaff.py* program.

Options:
--table : A tabulated input file having an identical structure of the tabulated file provided in *fusion_scaff.py* or the output file of *look4fusion.py*.
--out : Output file name containing scaffolds to group together. (Default: intermediate_junction.txt).
File description:

```
>scaffold1
        scaffold1   FWD   scaffold2   FWD
        scaffold1   FWD   scaffold3   REV
        scaffold2   FWD   scaffold3   REV
>scaffold 6
        scaffold6   REV   …
```

- **contig_scaff.py**

This program creates junctions between scaffolds using a tabulated file and output a multifasta file containing all sequences in the input fasta file, including joined scaffolds.

Options:
--table : A table file of scaffold to join.
File description:

```
>scaffold1"
scaffold1       FWD
scaffold2       FWD
scaffold3       REV
```

>scaffold6"
scaffold6 REV
…
--fasta : The multi-fasta file containing scaffolds.
--out : Output file name of the multi fasta file. (Default: super_contig.fasta)
--out_verif : Output file name that register the constitution of the newly formed scaffold. This file is used by *verif_fusion.py* to validate junctions performed. (Default: contig2verif.txt)

- **fusion_scaff.py**

This program merges scaffold sequences based on tabulated file.

Options:
--table : A tabulated file containing information of scaffold to merge.
File description: A tabulated file containing in column 1: scaffold_name; column 2: start; column 3: end; column 4: scaffold orientation (FWD or REV); column 5: scaffold_name; column 6: start; column 7: end; column 8: scaffold orientation; column 9: orientation of the inserted region.
The first 4 column collect information on scaffold to insert. Columns 5 to 9 collect information on the destination of the insertion. Column 4 and 8 are always “FWD”.
--fasta : A multifasta file containing sequences.
--out : Output file name of the multifasta file containing merged scaffolds. (Default: fusion.fasta)
--out_verif : Output file name that register the constitution of the newly formed scaffold. This file is used by *verif_fusion.py* to validate fusions performed. (Default: fusion2verif.txt)

- **verif_fusion.py**

This program verifies scaffold sequence fusions and junctions performed by fusion_scaff.py or contig_scaff.py. The verification is performed by drawing circos pictures representing paired reads overlapping scaffold junction/fusion performed.

Options:
--config : The configuration file generated by conf4circos.py.
--list : The file name passed in --out_verif option when running fusion_scaff.py or contig_scaff.py
--bound : Boundaries around junction to draw paired reads. Choose a value >= 2 fold library insert size. (Default: 10000)
--thread : Thread number used for circos drawing (integer). (Default: 1)
--out_tar : Output name of a tar.gz file containing circos pictures validating fusions and junctions performed. (Default: possible_fusion.tar)

- **reEstimateN.py**

This program re-estimate N present in DNA sequence. Re-estimated N are replaced by S. First paired read insert size is re-estimated and second correctly orientated paired reads overlapping unknown regions are used to re-estimate the size of these unknown regions.

Options:
--config : The configuration file generated in scaffremodler pipeline.
--exclude : Chromosome/scaffold names separated with "=" to exclude for the insert size estimation.
--min_read : The minimal read number requested to make the re-estimation of an unknown region. (Default: 30)
--out : Output file name of the fasta file containing re-estimated N regions. (Default: N_restimated.fasta)

--thread : Number of thread to use. (Default: 1)

- **Find_common_zones.py**

If you run scaffremodler on several accessions, this script find the common zones from the different accessions. The score files produced by this script can be used by the conf4Circos.py to draw only the desired zones.
When a zone is absent from an accession, the script look in the bam file and report a new zone found if the number of discordant reads. If this number is bigger than the minimal reads number of the config file, a new zone is reported with a specific flag.
The flags are:
0 : the zone is not found.
1 : this is a new zone identified only by reads.
2 : a zone identified by the pipeline but with a score value under the thresholds.
3 : a zone identified by the pipeline and a score value

You have to join a configuration file containing:

[Discordant_type]
Accession_name = path/to/the/discordant_score_file path/to/the/discordant_bam_file Minimum_Number_of_reads insert_size maximum_merge_distance

Example :

```
[chr_ff]
Accession1 = path/to/Accession1_chr_ff.score path/to/Accession1_chr_ff.bam 5 5000 2500
Accession2 = path/to/Accession2_chr_ff.score path/to/Accession2_chr_ff.bam 5 5000 2500
Accession3 = path/to/Accession3_chr_ff.score path/to/Accession3_chr_ff.bam 5 5000 2500

[chr_rr]
Accession1 = path/to/Accession1_chr_rr.score path/to/Accession1_chr_rr.bam 5 5000 2500
Accession2 = path/to/Accession2_chr_rr.score path/to/Accession2_chr_rr.bam 5 5000 2500
Accession3 = path/to/Accession3_chr_rr.score path/to/Accession3_chr_rr.bam 5 5000 2500

Etc…
```

Options:
--conf : The configuration file
--nump : The number of threads to use. (Default: 1)

- **drawDestZones.py**

From a zone defined by the user, this script draw all the zones of the genomes linked by reads presents in the first zone. A minimum number of reads can be define.

--zone : Coordinates of the zone (ex : chr01:10000:20000)
--minreads : Minimum reads to make a cluster. (Default: 5)
--confSc : Path to the configuration file of scaffremodler.
--confCi : Path to the configuration file of conf4circos.
--locaPrograms : Path to the file containing the path of the programs.
--nump : Number of processor to use.
--maxgap : Maximum maxgap in pb to clusterize two reads. (Default: 5000)