Tutorial on Organelle Genome Recombination Detection and Recombinant Map Construction

——A Guide to Using ReHRI and ReHRV Software

1 Install

1.1 Download

The URL is: https://github.com/wlqg1983/ReHRI_ReHRV_1.0

1.2 Unzip the compressed file and enter the folder

```
unzip ReHRI_ReHRV_1.0-main.zip
cd ReHRI_ReHRV_1.0-main
```

1.3 Install ReHRI and ReHRV

Currently, ReHRI and ReHRV can only run on Linux (v20.0.4) system and require conda (24.5.0) as a dependency. The user must activate the conda environment to use ReHRI and ReHRV. Below are the installation commands. If the software download speed is slow during installation, the user can manually modify the channels in the ReHRI_ReHRV_1.0.yml file.

```
conda env create -f ReHRI_ReHRV_1.0.yml
conda activate ReHRI_ReHRV_1.0
```

The "ReHRI_ReHRV_1.0" is a user-defined conda environment name. The environment name must start with a letter and only contain letters, numbers, underscores, or periods (no spaces or other special characters).

Update plasmidrender：

```
cp -r bin/plasmidrender $(conda info --base)/envs/ReHRI_ReHRV_1.0/lib/python3.12/site-packages/
```
(Entered as a single line)

```
chmod +x bin/*
```

1.5 Verify the installation results

Testing ReHRI.py help documentation

```
python bin/ReHRI.py -h
```

usage: ReHRI.py [-h] -c CONFIG [-redo] [-resume] [-v]

ReHRI: A tool to check spanning reads for supporting subconfig of your organelle genome.

Options:

-h, --help show this help message and exit
-c, CONFIG Path to external configuration file.
-redo Delete all previous results and start calculation anew.
-resume Resume from a previous project.
-v, --version Show the version number and exit.
Testing ReHRI.py version
python bin/ReHRI.py -v
ReHRI version=1.0

Testing ReHRV.py help documentation
python bin/ReHRV.py -h
usage: ReHRV.py [-h] -c CONFIG [-redo] [-v]
ReHRV: A tool to map the conFig. of your organelle genome.
Options:
-h, --help show this help message and exit
-c, CONFIG Path to external configuration file.
-redo Delete all previous results and start calculation anew.
-v, --version Show the version number and exit.
Testing ReHRV.py version
python bin/ReHRV.py -v
ReHRV version=1.0

2 Software operating principle

2.1 ReHRI operating principle

ReHRI (Repeat-mediated Homologous Recombination Identification) is based on the fundamental principle that homologous recombination between inverted repeats (IRs) results in inversion of the intervening genomic sequence (Fig. 2-1A), whereas recombination events involving direct repeats (DRs) generate two distinct subgenomic molecules (Fig. 2-1B).

Trimmed reference sequences (TRSs) are extracted from both the major configuration (main configuration) and minor configuration (sub configuration), centered around the repeat sequences (Fig. 2-2).

(1) Major Configuration:

TRSs are extracted around paired repeats RUa and RUb, generating TRS_{LR} (from LR)

and TRS_{UD} (from UD).

(2) Minor Configurations:

IR-mediated configuration (Fig. 1-1A): TRS_{LU} and TRS_{RD} are truncated, centered around RUb and RUa, respectively.

DR-mediated configuration (Fig. 2-2B): TRS_{LD} and TRS_{UR} are truncated, centered around RUa and RUb, respectively.

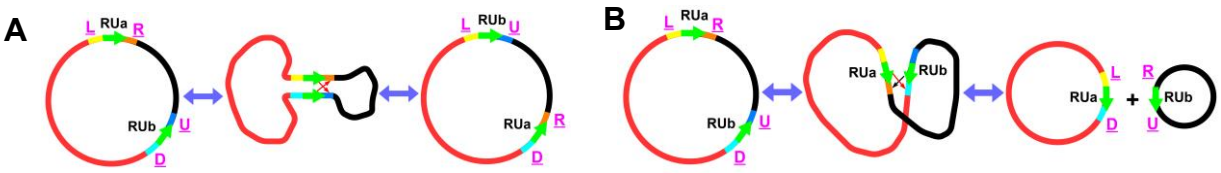

Fig. 2-1 Schematic diagram of repeat-mediated circular genome recombination

A: Recombination mediated by inverted repeats (IRs); B: Recombination mediated by direct repeats (DRs).

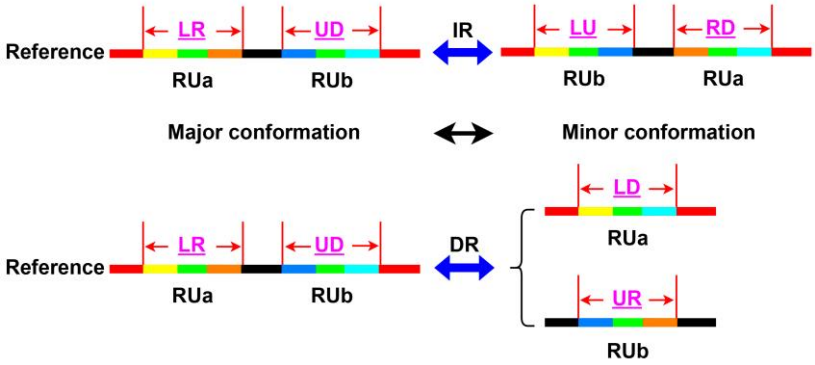

Fig. 2-2 Schematic diagram of TRS interception

Inverted repeat sequences (IRs) mediated recombination causes inversion of the intermediate sequence of (A) paired repeat units, while directed repeat sequences (DRs) mediated recombination causes (B) a chromosome to produce a pair of subtyping chromosomes. LR, UD, LD, UR, LU, and RD represent the truncated TRSs. LR and UD come from the main configuration. LD and UR are derived from DRs mediated minor configurations. LU and RD are derived from IRs mediated minor configurations.

As shown in Fig. 2-3, the analysis pipeline involves the following steps for both major and minor configurations:

(1) TRS Construction: *L* base pairs (bp) are extracted from both flanks of each repeat sequence to generate the trimmed reference sequence (TRS).

(2) Read Alignment: Sequencing reads are mapped to the constructed TRS using ReHRI's alignment algorithm.

(3) Spanning Read Identification: Successfully mapped reads are analyzed for their ability to span the repeat sequence ($I > 0$).

(4) Configuration Detection: The presence of any spanning reads ($M > 0$) confirms the existence of the genomic configuration corresponding to that specific TRS.

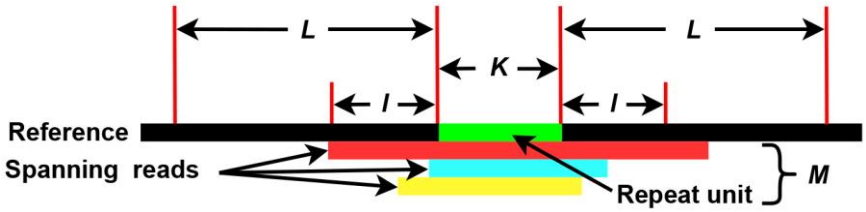

Fig. 2-3 Schematic diagram of mapping reads to a TRS

L: Indicates the length of a sequence intercepted on both sides of a repeat unit. The default value is 1000 base pairs (bp).

K: The length of a repeat unit. The default value is 5 bp.

I: Represents the length of a read across the left and right sides of a repeat unit. When *I* ≥ 1 base pair (bp), the read is considered to span a repeat unit.

M: The number of reads that span a repeat unit and *I* ≥ 1 bp.

2.2 Definition and calculation of recombination ratio

For mitogenome recombination probability calculations, we employ the following read counting methodology:

(1) Main Configuration Analysis:

TRS_{LR}: M_{LR} = Number of reads spanning RUa in main configuration

TRS_{UD}: M_{UD} = Number of reads spanning RUb in main configuration

(2) Minor Configuration Analysis:

TRS_{LU}: M_{LU} = Number of reads spanning RUb in DRs mediated minor configuration

TRS_{RD}: M_{RD} = Number of reads spanning RUa in DRs mediated minor configuration

TRS_{LD}: M_{LD} = Number of reads spanning RUa in IRs mediated minor configuration

TRS_{UR}: M_{UR} = Number of reads spanning RUb in IRs mediated minor configuration

The ratio of repeat unit RUa mediated recombination in IRs mediated genome recombination is:

$$p = \frac{M_{RD}^{-}}{M_{LR}^{+}} \times 100\% \quad (2-1) \qquad p = \frac{M_{RD}^{+} + M_{RD}^{-}}{M_{LR}^{-} + M_{LR}^{+}} \times 100\% \quad (2-2)$$

The above is the calculation formula for the recombination ratio considering two cases: single chain (formula 2-1) and double chain (formula 2-2) (+/- represents plus and minus chains, the same below).

The ratio of RUb mediated recombination in IRs mediated genome recombination is:

$$p = \frac{M_{LU}^{+}}{M_{UD}^{-}} \times 100\% \quad (2-3) \qquad p = \frac{M_{LU}^{+} + M_{LU}^{-}}{M_{UD}^{-} + M_{UD}^{+}} \times 100\% \quad (2-4)$$

The above is the calculation formula for the recombination ratio considering two cases: single chain (formula 2-3) and double chain (formula 2-4).

The ratio of RUa mediated recombination in DRs mediated genome recombination is:

$$p = \frac{M_{\mathrm{LD}}^{+}}{M_{\mathrm{LR}}^{+}} \times 100\% \quad (2-5) \qquad p = \frac{M_{\mathrm{LD}}^{+} + M_{\mathrm{LD}}^{-}}{M_{\mathrm{LR}}^{+} + M_{\mathrm{LR}}^{-}} \times 100\% \quad (2-6)$$

The above is the calculation formula for the recombination ratio considering two cases: single chain (formula 2-5) and double chain (formula 2-6).

The ratio of RUb mediated recombination in DRs mediated genome recombination is:

$$p = \frac{M_{\mathrm{RU}}^{+}}{M_{\mathrm{UD}}^{+}} \times 100\% \quad (2-7) \qquad p = \frac{M_{\mathrm{RU}}^{+} + M_{\mathrm{RU}}^{-}}{M_{\mathrm{UD}}^{+} + M_{\mathrm{UD}}^{-}} \times 100\% \quad (2-8)$$

The above is the calculation formula for the recombination ratio considering two cases: single chain (formula 2-7) and double chain (formula 2-8).

3 Running ReHRI software

```
python bin/ReHRI.py -c ReHRI.config.ini
python bin/ReHRI.py -c ReHRI.config.ini -redo
python bin/ReHRI.py -c ReHRI.config.ini -resume
```

ReHRI relies on a configuration file (ReHRI.config.ini) to set runtime parameters. While most parameters can be left at their default values, users only need to specify a few key settings.

In case of unexpected program interruption, ReHRI provides two recovery options:

(1) **-redo:** Deletes all intermediate results and restarts the computation from scratch.

(2) **-resume:** Resumes calculations from incomplete results, allowing the program to proceed until final outputs are generated.

This ensures flexibility in handling computational interruptions while optimizing runtime efficiency.

3.1 Operating mode I of ReHRI

In Mode I, ReHRI performs the following operations:

(1) Truncated TRSs within the provided genome.

(2) Identifies repeat sequence pairs capable of mediating genome recombination.

Required User Inputs (Fig. 3-1):

(1) Genome sequence file (inputfasta): Must be in FASTA format.

(2) Genome type (genome_type): Specify whether the genome is linear (L) or circular (C).

Notes for Multi-Chromosome Genomes:

If the genome contains multiple chromosomes, include all chromosomes in a single FASTA file.

```
[general]
;;·Parameters·for·general·set.
project_id·=·RI
mode·=·A
inputfasta·=·ReHRI_TestData/plastome_mitogenome.fasta
genome_type·=
complementary_chain·=·Yes
redundant_intermediate_results·=·D·
```

Fig. 3-1 Parameters in [general]

When searching for duplicate sequences, the length parameter (*K* value in Fig. 2-2) can be conFig.d flexibly (Fig. 3-2):

(1) Range Specification (using colons):

Bounded range: 50:1000 (50 bp ≤ length ≤ 1000 bp)

Minimum length only: 50: (length ≥ 50 bp)

Maximum length only: :100 (length ≤ 100 bp)

(2) Discrete Values (using commas):

Example: 50,100 (searches for lengths of 50 bp or 100 bp)

```
[ROUSFinder]
;; Parameters of ROUSFinder for finding repeats.
repeat_length = 32, 507:508
reward = 1
penalty = 20
```

Fig. 3-2 Length of repeat units in [ROUSFinder]

For sequencing data analysis, ReHRI supports both next-generation (NGS) and third-generation (TGS) sequencing data. However, the system currently processes either NGS or TGS data in a single run, as illustrated in Fig. 3-3. For paired-end NGS data, users should separate the forward and reverse read files with a space. When submitting TGS data, platform specification is required. Users must indicate whether the data was generated using Oxford Nanopore (ont) or PacBio (pacbio) technologies (Fig. 3-3).

```
[sequencing_depth]
;;·Parameters·for·mapping·reads·to·TRS·from·mainconfiguration·and·subconfigurat
alignment_software·=·minimap2
evalue·=·1e-5
NGS_single_end·=·
NGS_pair_ends·=·
TGS·=·ReHRI_TestData/Pacbio.CRR302668.1000.fastq
TGS_type·=·pacbio
filter_reads·=·Y
threads·=·
```

Fig. 3-3 Alignment softwares and sequencing data in [sequencing_depth]

ReHRI offers three alignment software options for mapping reads to TRSs: minimap2, bwa, and BLAST, with minimap2 set as the default. The applicable scenarios of minimap2, bwa, and blast are shown in Table 3-1.

Fig. 3-1 Comparison of Applicable Scenarios for Minimap2, BWA, and Blast

Tool	Best Application Scenario	Possible Missed Detection Reasons
minimap2	Long reads, fast alignment	Short reads, high-repeat regions, default parameters
BWA	Short reads, variant detection	Low efficiency with long reads
BLAST	Homology search, cross-species comparison	Time-consuming, not suitable for large-scale alignment

Detection of minor configurations is primarily governed by two key parameters: spanning_dead_flanking_depeat_length and spanning_dead_numbers (Fig. 3-4). These parameters are essential for identifying minor structural variations, with both defaulting to a value of 1. The two parameters can be modified in re-filtering mode (refiltermode=Y) to enable iterative screening of genome recombination events mediated by the queried repetitive sequences. Unless otherwise specified, all other parameters should retain their default values.

```
[check_spanning_reads]
;; Parameters for checking repeat-spanning reads.
spanning_read_flanking_repeat_length = 1
spanning_read_number = 1
```

Fig. 3-4 Detection parameters of minor configurations in [check_stpanning_deads]

Spanning_dead_flanking_depeat_length is the length of the read after crossing the repeated sequence, which is the value of *l* in Fig. 2-3. Spanning-read_number is the number of reads with a length ≥ *l* after crossing duplicate units, which is the *M* value in Fig. 2-3. It is suggested that both parameter values be set to 1, and the obtained results can be re filtered in Mode 3.

3.2 Operating mode II of ReHRI

The task of querying duplicate sequences is extremely challenging. The results of finding duplicate sequences by different algorithms are also different. Different duplicate sequence results have a significant impact on the results of mediating genome recombination. So, ReHRI has set up an interface that accepts users to provide duplicate sequence information, allowing users to provide duplicate sequences themselves (Fig. 3-5). At this point, it is necessary to set mode=C (Fig. 3-6). The repeat information file provided by the user is in .tsv format (Fig. 3-7).

When mode=C, ReHRI identifies duplicate sequence pairs by matching their shared

fragment_id. The system automatically detects all possible pairwise combinations of repeat units bearing identical fragment_id identifiers. For genomes containing a single chromosome, both chromosome and paired_chromosome columns must be excluded when providing paired repeat unit information.

To investigate specific recombination mediation effects, users may optionally supply a paired repeat unit information file (format specification shown in Fig. 3-8) to target particular repeat unit pairs of interest.

```
[manually_calibrated_repeat_info]
;; Parameters for calibrating results.
calibrated_repeat_file = ReHRI_TestData/Two_manual_rep.txt
```

Fig. 3-5 User-provided repeat information in [manually_calibrated_peat_info]

```
[general]
;; Parameters for general set.
project_id = AR
mode = C
```

Fig. 3-6 Parameters for ReHRI operating mode II

fragment_id	length	start	end	direction	chromosome
Repeat_10	16	63176	63161	minus	chr1
Repeat_10	16	66320	66305	minus	chr1
Repeat_2	17	4690	4706	plus	chr1
Repeat_2	17	50595	50611	plus	chr1
Repeat_15	87	332788	332874	plus	chr2
Repeat_15	87	359155	359069	minus	chr2

Fig. 3-7 Example of repeat information provided by users (tsv format)

chr1, chr2, chr3, ...: For chromosome numbering, this format must be used, and the numbering order represents the arrangement order of chromosomes in the input fasta file in [general]. The fragment_id of different repeat units of the same repeat units must be the same.

fragment_id	length	start	end	direction	chromosome	paired_id	paired_length	paired_start	paired_end	paired_direction	paired_chromosome
RU1a	17	4690	4706	plus	chr1	RU1b	17	50595	50611	plus	chr1
RU2a	16	63176	63161	minus	chr1	RU2b	16	66320	66305	minus	chr1
RU3a	87	332788	332874	plus	chr2	RU3b	87	359155	359069	minus	chr2

Fig. 3-8 Example of paired repeat information provided by users (tsv format)

chr1, chr2, chr3, ...: For chromosome numbering, this format must be used, and the numbering order must be the same as the arrangement order of chromosomes in the input fasta file in [general].

3.3 Operating mode III of ReHRI

If the initial filtering results are unsatisfactory, users can enable re-filtering by setting refilter_mode=Y (Fig. 3-9). This allows for adjustment of the key parameters spanning_dead_flanking_depeat_length and spanning_dead_numbers, followed by reprocessing of the query results to achieve improved outcomes.

```
[refilter_params]
;; Parameters for aggregrating the final results manually after the entire proce
refilter_mode = Y
refilter_id = FLT
spanning_read_flanking_repeat_length = 5
spanning_read_number = 5
redundant_intermediate_results = D
```

Fig. 3-9 Refilter parameters in [refilter_params]

3.4 Example data explanation of ReHRI

The various data examples and their usage scenarios used by ReHRI are shown in Table 3-2.

Table 3-2 Example data used for ReHRI

File name	File feature	Applicable scenarios
CRR302670_f1.10000.fasta	Forward reads: FASTQ or FASTA formats.	Evaluate NGS reads supporting genome recombination.
CRR302670_f1.10000.fastq		
CRR302670_r2.10000.fasta	Reverse reads: FASTQ or FASTA formats.	
CRR302670_r2.10000.fasta		
Pacbio.CRR302668.1000.fasta	TGS reads: FASTQ or FASTA formats.	Evaluate TGS reads supporting genome recombination.
Pacbio.CRR302668.1000.fastq		
NC_000932.1.fasta	*Arabidopsis thaliana* chloroplast genome	Testing recombination in the chloroplast genome.
NC_037304.1.fasta	*Arabidopsis thaliana* mitogenome	Testing recombination in the mitogenome.
plastome_mitogenome.fasta	*Arabidopsis thaliana* two organelle genomes	Testing recombination between multiple chromosomes.
One_manual_rep.tsv	Repeat sequences within a single chromosome	All paired repeat units were tested for genome recombination mediation activity.
One_manual_rep_corr.tsv	Paired repeat units within a single chromosome	Testing genome recombination mediation by specified repeat unit pairs.
Two_manual_rep.tsv	Intra- and inter-chromosomal repeat units	Genome-wide analysis of paired repeat units for recombination mediation capacity (intra- and inter-chromosomal).
Two_manual_rep_corr.tsv	Intra- and inter-chromosomal paired repeat units	Assessment of recombination mediation by user-defined repeat unit pairs (intra- and inter-chromosomal).

4 Interpretation of ReHRI core results

The running results of ReHRI are stored in the paired\peats-recomb-supporting_iratio.tsv file located in the folder {project_id}/final-repeat-spaning-results_{project_id}.

The result is a 20-column tsv file, as shown in Fig. 4-1.

1	2	3	4	5	6	7	8	9	10
fragment_id	length	start	end	direction	chromosome	plus_ratio(s/m)	minus_ratio(s/m)	combined_ratio	type
RU1a	17	4690	4706	plus	chr1	38/34	38/34	1.117647	direct_repeat
RU2a	16	63176	63161	minus	chr1	60/42	60/42	1.428571	direct_repeat
RU3a	87	332788	332874	plus	chr2	6/6	6/6	1	inverted_repeat
11	12	13	14	15	16	17	18	19	20
paired_id	paired_length	paired_start	paired_end	paired_direction	paired_chromosome	paired_plus_ratio(s/m)	paired_minus_ratio(s/m)	paired_combined_ratio	spanning_read_mcfg
RU1b	17	50595	50611	plus	chr1	50/16	50/16	3.125	sufficient
RU2b	16	66320	66305	minus	chr1	47/28	47/28	1.678571	sufficient
RU3b	87	359155	359069	minus	chr2	6/6	6/6	1	sufficient

Fig. 4-1 Repeat sequence-mediated recombination predicted by ReHRI

The meaning of each column is as follows:

① fragment_id: The ID of the repeat unit.

② length: The length of the repeat unit.

③ start: The start position of the repeat unit in the genome.

④ end: The end position of the repeat unit in the genome.

⑤ direction: The strand location (plus or minus).

⑥ chromosome: The chromosome ID in the genome.

⑦ plus_ratio: The ratio of reads spanning the repeat sequence in the minor vs. major configuration on the plus strand.

⑧ minus_ratio: The ratio of reads spanning the repeat sequence in the minor vs. major configuration on the minus strand.

⑨ combined_ratio: The overall ratio of repeat-mediated genomic recombination across both DNA strands.

⑩ type: The type of repeat sequence (direct or inverted repeat sequences).

⑪ spanning_read_mcfg: Whether the number of reads spanning the repeat units in the major configuration meets the user-defined threshold.

⑫ The "paired_*" notation denotes the complementary repeat unit involved in mediating genomic recombination events.

5 Detailed explanations of ReHRI configuration file

Users can explore the performance of ReHRI by setting the. ini configuration file in more details. Table 5-1 provides a detailed interpretation of the parameters in the. ini configuration file.

Table 5-1 Detailed explanation of parameters in the configuration file

Parameter Category	Parameter	Values and Descriptions
[general]	project_id (required)	Project ID, composed of letters, numbers, and underscores.
	mode (default: A)	Software operation mode, values: N/A/R/C (case-insensitive). N: Program does not run; A: Program runs automatically; R: Only runs ROUSFinder to identify repeat

		sequences; C: Identifies reads spanning repeat sequences from user-provided repeats. In this case, "calibrated_repeat_file" under the [calibrate_ROUSFinder_results] category must be provided.
	inputfasta (required)	Organelle genome sequence file, which may contain multiple chromosomes.
	genome_type (default: C)	Sets the genome as linear (L) or circular (C).
	complementary_chain (default: Y)	When identifying reads spanning repeat sequences, considers both DNA strands (Y) or not (N).
	redundant_intermediate_results (default: D)	Deletes intermediate results (D) or keeps them (K) during software operation.
[ROUSFinder]	repeat_length (default: 50:)	Must be ⩾5bp; ⩽50bp→50:; ⩽100bp→:100; 100-200bp→100:200; =30bp→30
	reward (default: 1)	ROUSFinder parameter: reward value for sequence alignment when identifying repeat sequences.
	penalty (default: 20)	ROUSFinder parameter: penalty value for sequence alignment when identifying repeat sequences.
[manually_calibrated_repeat_info]	calibrated_repeat_file	Input file location for manually calibrated repeat sequence results. Required when mode=C.
[mainconfiguration]	flanked_sequence_length (default: 1000bp)	In main configuration, the length of sequences extracted from both sides of the repeat unit.
[subconfiguration]	flanked_sequence_length (default: 1000bp)	In sub-configuration, the length of sequences extracted from both sides of the repeat unit.
[sequencing_depth]	alignment software (default: minimap2)	Alignment software options: minimap2, bwa, or blast.
	evalue (default: 1e-5)	Blast parameter: evaluates the significance of matching results.
	NGS_single_end	Second-generation single-end sequencing data (fastq or fasta format).
	NGS_pair_ends	Second-generation paired-end sequencing data (fastq or fasta format).
	TGS	Third-generation sequencing data (fastq or fasta format).
	TGS_type	Sets the third-generation sequencing platform type (pacbio or ont).
	filter_reads (default: Y)	Read filtering option: Enable to accelerate processing speed.
	Threads (Default value: 90% physical threads)	Number of threads. Uses default value if left empty.
[check_spanning_reads]	spanning_read_flanking_repeat_length (default: 1bp)	Length of the read spanning the repeat unit, must be a natural number.
	spanning_read_number (default: 1 bp)	Number of reads that meet the criteria for spanning the repeat unit.
[refilter_params]	refilter_mode (default: N)	Whether to re-filter reads spanning the repeat unit.
	refilter_id	Project ID for re-filtering reads spanning the repeat unit.
	spanning_read_flanking_repeat_length (default: 5 bp)	Re-filtering parameters for repeat-spanning reads. It must be a natural number.
	spanning_read_number (default: 5)	Re-filtering criteria for repeat-spanning reads. It must be a natural number.

6 Detailed explanations of ReHRI operation results

The folder named {project_id} stores all the results of ReHRI after running.

The folder final_depeat-spanning_desults_{project_id} stores all the information about duplicate sequences found in the query:

① one_chain_without_sufficient_spanning_reads.tsv
② one_repeat_unit_without_spanning_reads.tsv
③ paired_repeats_for_mapping.tsv
④ paired_repeats_recomb-supporting_ratio.tsv
⑤ repeat_sequences_{project_id}_chr1.fasta
⑥ repeat_sequences_{project_id}_chr2.fasta

File ① (Unpaired Repeat Sequences)

Contains repeat sequences from a single DNA strand (either positive or negative)

Content: Sequences lacking spanning reads for crossover detection

File ② (Non-Crossable Repeats)

Stores repeat sequences present on both DNA strands

Feature: All included sequences lack spanning reads for crossover events

File ③ (Visualization Subset)

Curated subset extracted from File ④

Application: Serves as input for genome recombination visualization in ReHRV

File ④ (Core Results)

Primary output containing all significant recombination events

Reference: Detailed interpretation provided in Section 5

Files ⑤ & ⑥ (Chromosomal Repeat Libraries)

Contain chromosome-specific repeat sequences in FASTA format

Format: Standard FASTA with chromosomal origin annotations.

The folder named subconfig_repeat-spanned_results_{project_id} stores the intermediate results of read mapping to the TRS under the secondary configuration. Similarly, the folder mainconfig_repeat-spanned_results_{project_id} contains the intermediate results of read mapping to the TRS under the main configuration. Each of these folders corresponds to a specific TRS mapping result, with their naming conventions detailed as follows:

DR_LD_RUxxa_RUxxb_plus_1000_results
DR_LD_RUxxa_RUxxb_minus_1000_results
DR_UR_RUxxa_RUxxe_plus_1000_results
DR_UR_RUxxa_RUxxe_minus_1000_results
IR_LU_RUxxb_RUxxc_plus_1000_results
IR_LU_RUxxb_RUxxc_minus_1000_results

IR_RD_RUxxa_RUxxb_plus_1000_results
IR_RD_RUxxa_RUxx1b_minus_1000_results
DR_LR_RUxxa_RUxxb_plus_1000_results
DR_LR_RUxxa_RUxxb_minus_1000_results
DR_UD_RUxxa_RUxxe_plus_1000_results
DR_UD_RUxxa_RUxxe_minus_1000_results
IR_LR_RUxxa_RUxxb_plus_1000_results
IR_LR_RUxxa_RUxxb_minus_1000_results
IR_UD_RUxxa_RUxxe_plus_1000_results
IR_UD_RUxxa_RUxxe_minus_1000_results

The naming rules for each part of the folder name are shown in Table 6-1:

Table 6-1 Meaning of characters in result folder names

Symbol	Meaning of the Symbol
DR	Direct Repeat sequence
IR	Inverted Repeat sequence
LU	TRSs from minor configurations, using IR unit RUb as the genomic anchor point (Fig. 2-3).
RD	TRSs from minor configurations, using IR unit RUa as the genomic anchor point (Fig. 2-3).
LD	TRSs from minor configurations, using DR unit RUa as the genomic anchor point (Fig. 2-3).
UR	TRSs from minor configurations, using DR unit RUb as the genomic anchor point (Fig. 2-3).
LR	TRSs from major configurations, using RUa as the genomic anchor point (Fig. 2-3).
UD	TRSs from major configurations, using RUb as the genomic anchor point (Fig. 2-3).
RU	Repeat Unit
xx	Number, a natural number
a/b/c...	Different repeat units of the same repeat sequence
plus	Represents the plus strand of genome
minus	Represents the minus strand of genome
1000	Length of the sequence extracted from both sides of RU, i.e., *L* in Fig. 2-3
results	Suffix for folder names

Each folder contains TRS sequences (in fasta format), and reads across repeated sequences in TRS are mapped to the sequencing depth of TRS, and reads are mapped to the BAM document of TRS (Fig. 6-1).

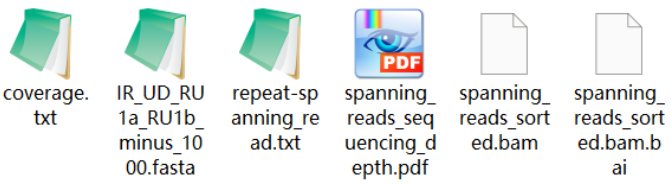

Fig. 6-1 Various results after mapping read to each TRS

The read mapping across repeated sequences in TRS to the sequencing depth of TRS is shown in Fig. 6-2, and the sequencing depth values are saved in coverage.txt. The BAM document can be visualized using software such as Tablet to map the actual situation of the read to TRS (Fig. 6-3).

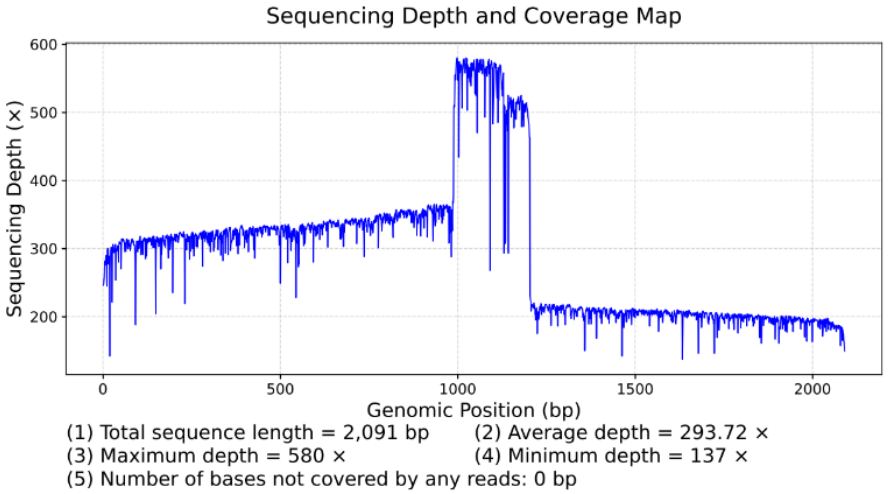

Fig. 6 Sequencing depth of reads spanning RU when mapped to TRS

Fig. 6-3 Visualization results of reads spanning RU when mapped to TRS

When ReHRI filters the results of the initial screening again, that is, after ReHRI executes mode III, the results are stored in the folder named {project_id} in the refiltered_repeat-spanning_results_{FLT} folder. The interpretation of the results is based on the interpretation of the results in the folder final_depeat-spanning-results_{project_id}.

7 ReHRV generates genomic maps of repeat sequences.

The software ReHRV (Repeat mediated Homologous Recombination Visualization) can draw a schematic diagram of circular genome recombination based on the results of ReHRI software, to display the genomic maps of various subtypes of mitogenomes mediated by repeats. The graph is represented in a circular pattern, with arrows indicating the direction of the DNA molecule's positive strand before and after recombination. Repeat units (RU) are represented by colored blocks, with the same-colored blocks representing different units of

the same repeat sequence, and different colors representing different repeat sequences. The “ctg” represents the region between two adjacent repeat units (Fig. 7-1A). The map of linear chromosomes is a circular map with notches (Fig. 7-1B). The size of the graph radius represents the length of the genome sequence.

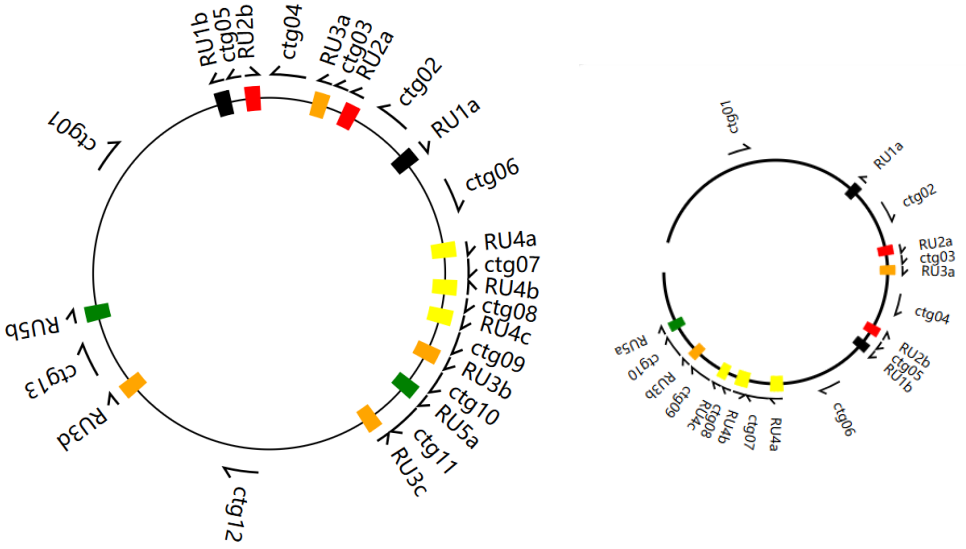

Fig. 7-1 Schematic diagram of the genome map drawn by ReHRV

7.1 Operation of ReHRV

The command line to run ReHRV is as follows:

```
python bin/ReHRV.py -c ReHRV.config.ini
python bin/ReHRV.py -c ReHRV.config.ini -redo
```

The parameters of ReHRV are provided in the form of an .ini configuration file. The vast majority of these parameters come with default values, and only a small number require user input. Specifically, the **'-redo'** parameter enables users to redraw the repeat sequence gene map in the event of an unexpected program interruption. It should be noted that this process will delete the intermediate results generated from the previous run.

7.2 Configuration file of ReHRV

ReHRV is capable of generating genome maps for three modes: the main configuration ([mainconfiguration] mode; Fig. 7-2A), the inverted repeat (IR)-mediated secondary configuration ([IR-mediated_deverse_recover] mode; Fig. 7-2B), and the direct repeat (DR)-mediated secondary configuration ([DR-mediated_decomb_1to2] mode; Fig. 7-2C). For map generation, users are required to provide information including the positions of repeated sequences, the genome sequence (in FASTA format), genome length, and genome type (i.e., linear or circular structure).

Each mode is conFig.d with an auto_map parameter to control whether the corresponding mode is executed (with options: Y/N/M). Specifically:

(1) When auto_map=N, ReHRV will not run the corresponding mode.

(2) When auto_map=Y, ReHRV will automatically generate all genome maps.

(3) When auto_map=M, ReHRV will generate user-specified genome maps under interactive guidance.

In cases where two or more chromosomes merge into a single chromosome under DR mediation, ReHRV only allows users to input two chromosomes at a time. Therefore, users need to run ReHRV repeatedly in the [DR-mediated_decomb_2to1] mode to merge multiple chromosomes into one. When two chromosomes recombine into a single chromosome, one of them must have a circular structure (C); the parameter settings for this scenario are shown in Fig. 7-2D (chr1_type, chr2_type).

If both chromosomes are linear (L), ReHRV can only perform cross-sequence recombination between the two chromosomes in the [DR-mediated_decomb_2to2] mode, resulting in two new linear chromosomes. The relevant parameter settings are illustrated in Fig. 7-2E.

A

```
[mainconfiguration]
;; Parameters for mapping genome mainconfiguration.
auto_map = Y
inputfile = ReHRV_TestData/paired_repeats_for_mapping_virtual.tsv
genome_length = 605764
genome_type = C
```

B

```
[IR_mediated_reverse_recomb]
;; Parameters for drawing maps of Inverted Repeat (IR) mediated genome recombina
auto_map = Y
inputfile = ReHRV_TestData/paired_repeats_for_mapping_virtual.tsv
inputfasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2.fasta
genome_type = C
```

C

```
[DR_mediated_recomb_1to2]
;; Parameters for drawing maps of organelle genome recombination mediated by dir
auto_map = y
inputfile = ReHRV_TestData/paired_repeats_for_mapping_virtual.tsv
inputfasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2.fasta
genome_type = C
```

D

```
[DR_mediated_recomb_2to1]
;; Parameters for drawing maps of organelle genome recombination mediated by dir
auto_map = y
flip_chain = Y
chr1_file = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2_map.tsv
chr1_fasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2.fasta
chr1_type = C
chr2_file = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr2_1to2_map.tsv
chr2_fasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr2_1to2.fasta
chr2_type = C
```

E
```
[DR_mediated_recomb_2to2]
;; Parameters for drawing maps of organelle genome recombination mediated by dir
auto_map = Y
flip_chain = Y
chr1_file = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2_map.tsv
chr1_fasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr1_1to2.fasta
chr2_file = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr2_1to2_map.tsv
chr2_fasta = ReHRV_TestData/ReHRV_DR_RU5a_RU5b_chr2_1to2.fasta
```

Fig. 7-2 Parameters required for ReHRV to draw various genome maps

In the [DR-mediated-recomb_2to2] and [DR-mediated-recomb_2to1] modes, due to the free rotation of the two chromosomes, all repeat units capable of mediating genome recombination can drive the fusion of the two chromosomes into one via positive repetition. Thus, when the parameter flipuchain=Y, it enables all such recombination-mediating repeat units to facilitate the formation of a single chromosome through positive repetition of the two chromosomes.

Across all mapping modes, the input files (inputfile, chr1_file, and chr2_file) follow the format of 8-column TSV lists, as depicted in Fig. 7-3. Each line corresponds to a pair of repeat sequences, and paired repeat sequences may be omitted. Notably, the header names and repeat sequence designations must strictly match those illustrated in Fig. 7-3.

fragment_id	start	end	direction	paired_id	paired_start	paired_end	paired_direction
RU3c	386117	385989	minus				
RU1a	153164	152647	minus	RU1b	231395.0	231912.0	plus
RU2a	193682	193809	plus	RU2b	231329.0	231202.0	minus
RU3a	196488	196753	plus	RU3d	556153.0	556418.0	plus
RU3d	556153	556418	plus	RU3b	343152.0	343280.0	plus
RU4a	317468	310009	minus	RU4b	317468.0	310009.0	minus
RU4a	317468	310009	minus	RU4c	322009.0	327009.0	plus
RU5a	359772	363702	plus	RU5b	601834.0	605764.0	plus

Fig. 7-3 Schematic diagram of 8 column table for ReHRV (tsv format)

7.3 Parameter for each element in the genome map

The [mapper_config] option is used to conFig. the attribute parameters of each element in the generated genome map. The specific parameter values are shown in Fig. 7-4:

```
[mapper_config]
;; Parameters for the properties of various elements in the genome map.
picture_box = 450
radius = 150
arrow_radius = 170
arrow_size = 10
arrow_thickness = 2
font_size = 18
tag_height = 20
tag_line_width = 1
output_svg_file = mainconfig
```

Fig. 7-4 Parameters of each element in the genome map in [mapper_config]

The default values and meanings of each parameter are shown in Table 7-1:

Table 7-1 Default values and descriptions of [mapper_config] options

Parameters	Parameter Values
picture_box	Size of the output image (length of one side of the square), default=280
radius	Radius of the genomic map, determines the image size, default=150
arrow_radius	Radius of the circle where the arrow is located, default=170
arrow_size	Size of the arrow, default=10
arrow_thickness	Thickness of the arrow line, default=2
font_size	Font size, default=18
tag_height	Height of the tag (annular sector), default=20
tag_line_width	Outline width of the tag (annular sector), default=1

7.4 Color parameters of repeat sequences

The [color_library] option is used to set the colors of repeat sequences in the genome map. The available color representations include 60 built-in color names (in English), as well as RGB values and hexadecimal color codes.

The 60 built-in color options are derived from Python's webcolors library, specifically: black, red, orange, yellow, green, cyan, blue, purple, brown, gray, darkslategray, dimgray, navy, indigo, darkgreen, darkred, firebrick, crimson, chocolate, olive, yellowgreen, lawngreen, limegreen, greenyellow, lightseagreen, seagreen, darkseagreen, lightgreen, forestgreen, darkcyan, mediumturquoise, turquoise, aquamarine, mediumaquamarine, aqua, deepskyblue, skyblue, steelblue, cadetblue, royalblue, mediumblue, darkviolet, plum, deeppink, hotpink, pink, palevioletred, mediumvioletred, coral, orangered, darkorange, goldenrod, gold, khaki, darkkhaki, wheat, lightgrey, lightslategray, slategray, darkgray.

The color values for RGB are set as follows: 0.0.0 (black), 255.0.0 (red), 255.165.0 (orange), 255.255.0 (yellow), 0.255.0 (green), 0.255.255 (cyan), 0.0.255 (blue), 128.0.128 (purple). The hexadecimal values representing colors are: # 000000 (black), # FF0000 (red), # FFA500 (orange), # FFFF00 (yellow), # 00FF00 (green), # 00FFFF (blue), # 800080 (purple). ReHRV allows up to 30 repeated units to be colored simultaneously, as shown in Fig. 7-5:

```
[color_library]
;; Colouring scheme for different repeats.
RU1 = black
RU2 = red
RU3 = orange
RU4 = yellow
RU5 = green
RU6 = cyan
RU7 = blue
RU8 = purple
RU9 = gray
RU10 = slategray
RU11 = forestgreen
RU12 = gold
RU13 = indigo
RU14 = darkgreen
RU15 = darkred
RU16 = firebrick
RU17 = limegreen
RU18 = chocolate
RU19 = olive
RU20 = yellowgreen
RU21 = turquoise
RU22 = plum
RU23 = greenyellow
RU24 = darkslategray
RU25 = aqua
RU26 = navy
RU27 = crimson
RU28 = skyblue
RU29 = coral
RU30 = brown
```

Fig. 7-5 Coloring scheme for repeat sequences within the genome map

7.5 Parameters for arranging multiple genomic maps

The [Arrange_map] option is used to conFig. parameters for multi-graph layouts, where graphs are arranged in a 3x3 grid format (see Fig. 7-6A).

Arrange: Specifies whether multiple genome maps should be formatted. "Yes/Y" enables typesetting, while "No/N" disables it.

Font_size: Sets the font size of labels.

Image_dpi: Controls the resolution of the formatted image (default values are available for use).

Each genome map can have a simple label placed at its center (Fig. 7-6B), which may consist of uppercase and lowercase letters, numbers, and underscores. Additionally, each genome map can be independently positioned within the formatted image (Fig. 7-6C).

A

```
[Arrange_map]
;;·Arrange·images·into·a·grid·of·nine·squares.
;·General·set.
arrange·=·Y
font_size·=·20
image_dpi·=·600
```

B

```
;·Remarks·info·in·the·circle·map.
center_font·=·Main
left_middle_font·=·RU1
right_middle_font·=·RU2
top_middle_font·=·RU3
bottom_middle_font·=·RU4
top_left_font·=·RU5
top_right_font·=·RU6
bottom_left_font·=·RU7
bottom_right_font·=·RU8
;·*NOTE*:·Set·the·text·content·as·desired,·the·content·can·be·left·blank.
```

C

```
;·Path·to·up·to·nine·maps.
center_path·=·ReHRV_TestData/mainconfig_ReHRV_map.svg
left_middle_path·=·ReHRV_TestData/ReHRV_DR_RU3a_RU3d_chr1_1to2_map.svg
top_left_path··=·ReHRV_TestData/ReHRV_DR_RU3a_RU3d_chr2_1to2_map.svg
top_middle_path·=·ReHRV_TestData/ReHRV_DR_RU3a_RU3d_chr1_1to2_map.svg
top_right_path·=·ReHRV_TestData/ReHRV_DR_RU3a_RU3d_chr2_1to2_map.svg
right_middle_path·=·ReHRV_TestData/ReHRV_IR_RU2a_RU2b_map.svg
bottom_right_path·=·ReHRV_TestData/ReHRV_IR_RU4a_RU4b_map.svg
bottom_middle_path·=·ReHRV_TestData/ReHRV_DR_RU3a_RU3d_chr1_1to2_map.svg
bottom_left_path·=·ReHRV_TestData/ReHRV_IR_RU1a_RU1b_map.svg·
;·*NOTE*:·Only·accept·images·in·svg·format.·Set·the·path·as·desired,·or·leave·it
```

Fig. 7-6 Parameters for arranging multiple genomic maps

The positions of genomic maps within the 3×3 grid are shown in Fig. 7-7 below:

Top Left	Top Middle	Top Right
Left Middle	Center	Right Middle
Bottom Left	Bottom Middle	Bottom Right

Fig. 7-7 Schematic diagram of the genome map's position within the 3×3 grid.

The effect of the layout of the graph in the nine grid is shown in Fig. 7-8:

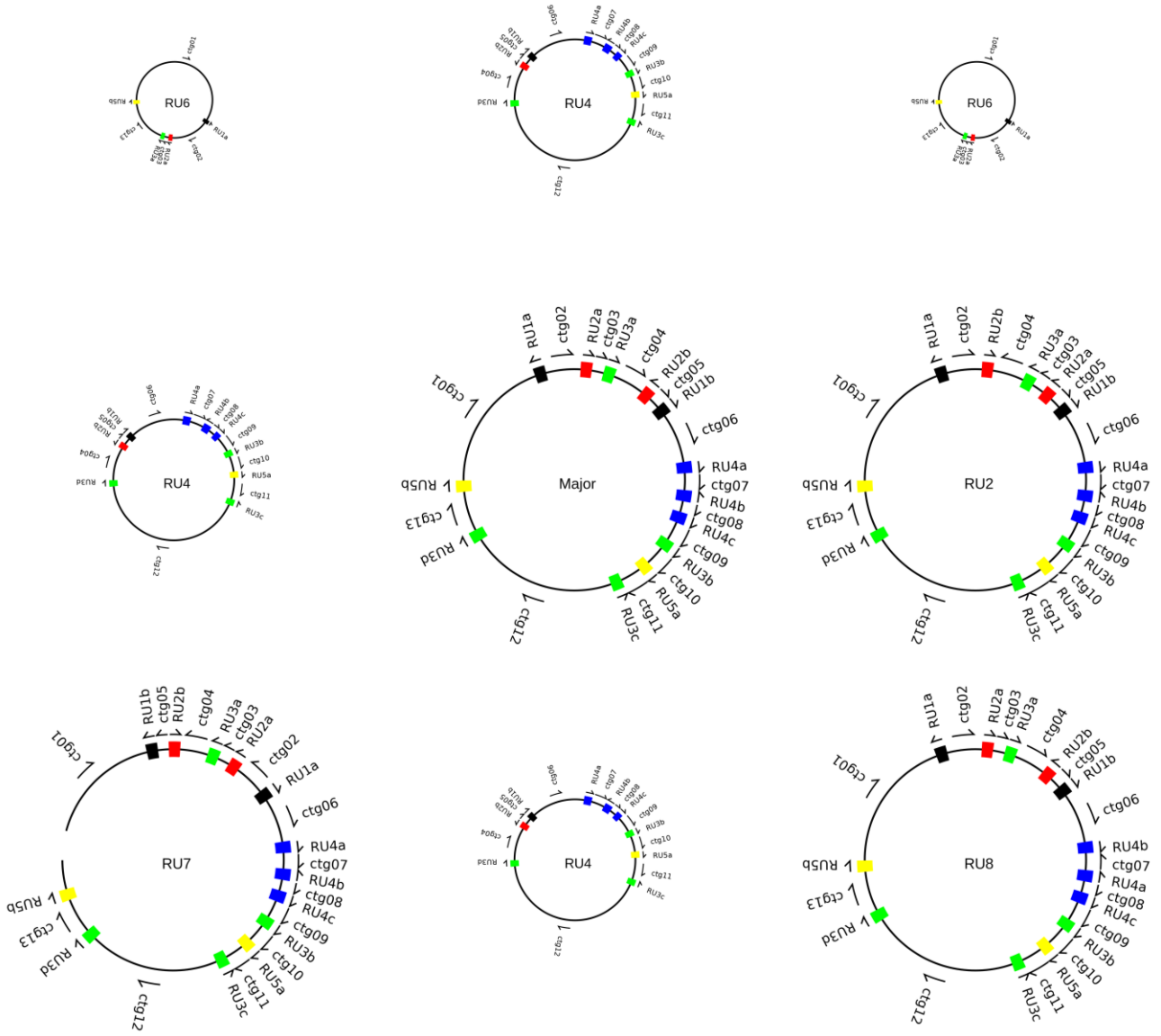

Fig. 7-8 The effect diagram after layout in the3×3 grid.

7.6 Interpretation of drawing results in each mode

The results are stored in a folder named {project_id}, which are generated by running the modes of [mainconfiguration], [IR_mediated_reverse_recomb], [DR_mediated_recomb_1to2], [DR_mediated_recomb_2to1], [DR_mediated_recomb_2to2], and [Arrange_map].

[mainconfiguration]：mainconfig_{project_id}
[IR_mediated_reverse_recomb]：Inv_Rev_{project_id}
[DR_mediated_recomb_1to2]：DR_1to2_{project_id}
[DR_mediated_recomb_2to1]：DR_2to1_{project_id}
[DR_mediated_recomb_2to2]：DR_2to2_{project_id}
[Arrange_map]：map_nine_squares_{project_id}

For example, two types of results are derived from the [DR-mediated_decomb_1to2]

mode:

A genome map of repeat sequences under the main configuration. This map is generated based on the 8-column lists input by the user and saved in SVG format, which can be opened using web browsers or software such as Adobe Illustrator CS6. Relevant information for the two generated chromosomes (chr1, chr2). Each chromosome corresponds to three files: a FASTA-format sequence file, an SVG-format repeat genome map, and the 8-column lists associated with the genome map (Fig. 7-9).

The output structure of other modes is similar to that of [DR-mediated_decomb_1to2]. Specifically, in the [DR-mediated_decomb_2to1] mode, when merging three or more sequences into a single chromosome, users can utilize the 8-column list (TSV format) and FASTA-format files generated from a previous run as input for subsequent runs. This iterative process enables the recombination of multiple chromosomes into one.

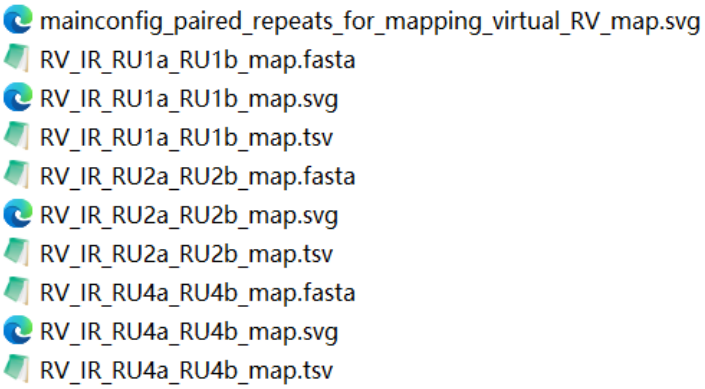

Fig. 7-9 Schematic diagram of the output results for each mode.