

# PanGraphRNA User Manual

## (version 1.0)

- PanGraphRNA is an efficient, flexible and web-based Galaxy platform that can be easily used to construct graph pangenomes from genetic variations at individual, subpopulation, and population levels. It can assist researchers to select appropriate graph pangenomes using various performance metrics for both real and simulation experiments.
- Currently, PanGraphRNA is composed of four functional modules: **Graph Pangenome Preparation Module, Construction Module, Evaluation Module, and Application Moudule.**
- PanGraphRNA was powered with an advanced packaging technology, which enables compatibility and portability.
- PanGraphRNA project is hosted on <https://github.com/cma2015/PanGraphRNA>
- PanGraphRNA docker image is available at <https://hub.docker.com/r/malab/pangraphrna>

## Graph Pangenome Construction Module and Alignment

This module implements a fast, memory-efficient toolkit HISAT2 to construct graph pangenomes at the individual, subpopulation, or population level. Subsequently, it performs read-genome alignment.

Tools	Description	Input	Output	Time (test data)	Reference
<b>Individual Level Graph Pangenome</b>	Construct individual level graph pangenome and perform read-genome alignment	Reference genome in FASTQ format and variation information in VCF format	HISAT2 alignment report in TXT format and alignment result in BAM format	~10 mins	<a href="#">HISAT2</a>

Tools	Description	Input	Output	Time (test data)	Reference
<b>Subpopulation Level Graph Pangenome</b>	Construct subpopulation level graph pangenome and perform read-genome alignment	Reference genome in FASTQ format and variation information in VCF format	HISAT2 alignment report in TXT format and alignment result in BAM format	~10 mins	<a href="#">HISAT2</a>
<b>Population Level Graph Pangenome</b>	Construct population level graph pangenome and perform read-genome alignment	Reference genome in FASTQ format and variation information in VCF format	HISAT2 alignment report in TXT format and alignment result in BAM format	~10 mins	<a href="#">HISAT2</a>

For large genomes like maize, graph construction and indexing require memory proportional to genome size and variant density. Incorporating known splicing sites during indexing constrains read boundaries is a strategy of reducing spurious alignments from genome complexity and improving mapping efficiency. In polyploid genomes, reporting additional multimapping loci may help evaluate alignment ambiguity caused by high subgenomic sequence similarity and its impact on downstream analyses. For highly heterozygous genomes, careful variant organization is critical: anchoring the graph on the most representative haplotype as the primary backbone and progressively integrating other variants enhances allelic diversity representation.

## Individual Level Graph Pangenome

In this function, an ultrafast and memory-efficient tool **HISAT2** (Kim, D., *et al.*, 2019) is integrated for constructing individual level graph pangenomes and aligning sequencing reads. See <https://daehwankimlab.github.io/hisat2/manual> for details.

### Input

- **Input reference genome file:** Input reference genome file for primary path of graph pangenome in FASTA format
- **Input VCF file:** Input VCF file containing variant information to be integrated into the primary path of graph pangenome in VCF format
- **Input FASTQ file:** Cleaned single-end or paired-end RNA-seq reads in FASTQ format

## Parameters

- **Accession name:** Input accession name available in the VCF to specify the variant data (Default: 628)
- **Threads:** The number of threads used for parallel computation (Default: 10)
- **VCF prefix:** The prefix of variant records (e.g., var\_1125, "var" is the vcf prefix) (Default: var)

## Output

- **HISAT2 alignment report in TXT format**
- **HISAT2 alignment result in BAM format**

The screenshot shows the 'Individual Level Graph Pangenome' workflow in the AnVIL Galaxy environment. The interface includes a left sidebar with tool categories, a central workflow form, and a right sidebar with a history of datasets. Red boxes and arrows highlight the following steps:

- Step 1:** select a reference genome (points to the 'Input reference genome file for primary path of graph pang genome' field with value '6: tair10.fasta')
- Step 2:** select a VCF file (points to the 'Input VCF file containing variant information to be integrated into the primary path of graph pang genome' field with value '7: 1001all\_test.vcf')
- Step 3:** input an accession name (points to the 'Accession name' field with value '628')
- Step 4:** choose single-end or paired-end (points to the 'Single-end or paired-end reads?' dropdown menu with value 'single-end')
- Step 5:** select clean reads (points to the 'Input FASTQ file (SE type)' field with value '5: SRR1234567.fastq\_clean\_reads.fastq')
- Step 6:** click here to run this function (points to the 'Execute' button)

Additional visible parameters include 'Strand-specific reads?' set to 'non-strand-specific', 'Threads' set to 10, and 'Extract uniquely mapped reads?' set to 'Yes'. The 'History' panel on the right lists several datasets, including '1001all\_test.vcf', 'tair10.fasta', and various SRR1234567 datasets.

## Subpopulation Level Graph Pangenome

In this function, an ultrafast and memory-efficient tool **HISAT2** (Kim, D., *et al.*, 2019) is integrated for constructing subpopulation level graph pang genomes and aligning sequencing reads. See <https://daehwankimlab.github.io/hisat2/manual> for details.

## Input

- **Input reference genome file:** Input reference genome file for primary path of graph pang genome in FASTA format
- **Input VCF file:** Input VCF file containing variant information to be integrated into the primary path of graph pang genome in VCF format

- **Input accession name list:** Input accession name list (TXT file) available in the VCF file to specify the variant data
- **Input FASTQ file:** Cleaned single-end or paired-end RNA-seq reads in FASTQ format

## Parameters

- **Threads:** The number of threads used for parallel computation (Default: 10)
- **VCF prefix:** The prefix of variant records (e.g., var\_1125, "var" is the vcf prefix) (Default: var)

## Output

- **HISAT2 alignment report in TXT format**
- **HISAT2 alignment result in BAM format**

The screenshot displays the AnVIL web interface for the 'Subpopulation Level Graph Pangenome' tool (Galaxy Version 1.0.0). The interface includes a left sidebar with navigation links, a central tool configuration panel, and a right sidebar showing a history of datasets. Red boxes and arrows highlight the following steps:

- Step 1: select a reference genome**: Input reference genome file for primary path of graph pangenome (6: tair10.fasta).
- Step 2: select a VCF file**: Input VCF file containing variant information to be integrated into the primary path of graph pangenome (7: 1001all\_test.vcf).
- Step 3: input an accession list**: Input accession name list (TXT file) available in the VCF file to specify the variant data (10: accession\_list).
- Step 4: choose single-end or paired-end**: Single-end or paired-end reads? (single-end).
- Step 5: select clean reads**: Input FASTQ file (SE type) (5: SRR1234567.fastq\_clean\_reads.fastq).
- Step 6: click here to run this function**: Execute button.

The right sidebar shows a history of datasets, including:

- 10 : accession\_list
- 9 : hisat2 alignment individual summary.txt (on SRR1234567.fastq\_clean\_reads.fastq)
- 8 : hisat2 alignment individual summary.txt (on SRR1234567.fastq\_clean\_reads.fastq)
- 7 : 1001all\_test.vcf
- 6 : tair10.fasta
- 5 : SRR1234567.fastq\_clean\_reads.fastq
- 4 : SRR1234567.fastq\_read\_s\_quality\_report.html
- 3 : SRR1234567.fastq
- 2 : SRR1234567.sra
- 1 : SRR1234567.sra

## Population Level Graph Pangenome

In this function, an ultrafast and memory-efficient tool **HISAT2** (Kim, D., *et al.*, 2019) is integrated for constructing population level graph pangenomes and aligning sequencing reads. See <https://daehwankimlab.github.io/hisat2/manual> for details.

## Input

- **Input reference genome file:** Input reference genome file for primary path of graph pangenome in FASTA format

- **Input VCF file:** Input VCF file containing variant information to be integrated into the primary path of graph pangenome in VCF format
- **Input FASTQ file:** Cleaned single-end or paired-end RNA-seq reads in FASTQ format

## Parameters

- **Threads:** The number of threads used for parallel computation (Default: 10)
- **VCF prefix:** The prefix of variant records (e.g., var\_1125, "var" is the vcf prefix) (Default: var)

## Output

- **HISAT2 alignment report in TXT format**
- **HISAT2 alignment result in BAM format**

The screenshot displays the AnVIL Galaxy interface for the 'Population Level Graph Pangenome' tool (Galaxy Version 1.0.0). The interface includes a left sidebar with navigation options like 'Tools', 'Workflows', and 'History'. The main workspace shows the tool's configuration form with five steps highlighted by red boxes and arrows:

- Step 1: select a reference genome** - Points to the 'Input reference genome file for primary path of graph pangenome' field, which has '6: tair10.fasta' selected.
- Step 2: select a VCF file** - Points to the 'Input VCF file containing variant information to be integrated into the primary path of graph pangenome' field, which has '7: 1001all\_test.vcf' selected.
- Step 3: choose single-end or paired-end** - Points to the 'Single-end or paired-end reads?' dropdown menu, which is set to 'single-end'.
- Step 4: select clean reads** - Points to the 'Input FASTQ file (SE type)' field, which has '5: SRR1234567.fastq\_clean\_reads.fastq' selected.
- Step 5: click here to run this function** - Points to the 'Execute' button at the bottom of the configuration form.

Below the configuration form, there is a section titled 'What it does' describing the tool's function: 'In this function, an ultrafast and memory-efficient tool HISAT2 (Kim, D., et al., 2019) is integrated for constructing graph pangenomes and aligning sequencing reads. See <https://daehwankimlab.github.io/hisat2/manual> for details.'

The right sidebar shows a 'History' panel with a list of datasets, including '12: hisat2\_alignment\_subpopulation.bam', '11: hisat2\_alignment\_subpopulation\_summary.txt', '10: accession\_list', '9: hisat2\_alignment\_individual.bam', '8: hisat2\_alignment\_individual\_summary.txt', '7: 1001all\_test.vcf', '6: tair10.fasta', '5: SRR1234567.fastq\_clean\_reads.fastq', and '4: SRR1234567.fastq\_read'.