

## SPAdes de novo genome assembly

**Citation:** If you use SPAdes in your research, cite [this paper](#).

- If you use other pipelines, cite the following papers:
  - metaSPAdes: [Nurk et al., 2017](#)
  - plasmidSPAdes: [Antipov et al., 2016](#)
  - metaplasidSPAdes / plasmidVerify: [Antipov et al., 2019](#)
  - metaviralSPAdes / viralVerify: [Antipov et al., 2020](#)
  - rnaSPAdes: [Bushmanova et al., 2019](#)
  - biosyntheticSPAdes: [Meleshko et al., 2019](#)
  - coronaSPAdes: [Meleshko et al., 2022](#)
- Other types of assembly citations found [here](#)

**About:** SPAdes is designed for assembly and analysis of sequencing data. SPAdes is primarily developed for Illumina sequencing data.

- The SPAdes package contains assembly pipelines for isolated and single-cell bacterial, as well as metagenomic and transcriptomic
  - Additional modes allow to discover bacterial plasmids and RNA viruses, as well as perform HMM-guided assembly
- SPAdes package includes supplementary tools for efficient k-mer counting and k-mer-based read filtering, assembly graph construction and simplification, sequence-to-graph alignment and metagenomic binning refinement

**Quick start:** this is not a complete [user guide](#), just for convenience

- SPAdes supports assembly for second-gen sequencing data (Illumina)
- Download SPAdes binaries for Linux or MacOS [here](#), or compile SPAdes from [source](#)
- SPAdes only requires Python 3.8+ to be installed
- Test your SPAdes installation: `bin/spades.py --test`
- For a single paired-end library: `bin/spades.py -1 <left.fastq.gz> -2 <right.fastq.gz> -o <output_folder>`
- Assemble an RNA viral genome: `bin/spades.py --rnaviral -1 <left.fastq.gz> -2 <right.fastq.gz> -o output_folder`
  - Available assembly modes: `--isolate`, `--sc`, `--plasmid`, `--meta`, `--metaplasid`, `--metaviral`, `--rna`, `--rnaviral`, `--bio`, `--corona`, `--sewage`
- [Reported errors / debugging](#)
- Standalone SPAdes tools
  - k-mer counting: `spades-kmer-counting`
  - Read filtering using k-mer coverage: `spades-read-filter`
  - Estimating number of unique k-mers: `spades-kmer-estimating`
  - Assembly graph construction: `spades-gbuilder`
  - Assembly graph simplification: `spades-gsimplifier`
  - Splitting assembly graph into components: `spades-gfa-split`
  - Alignment of long reads to assembly graph: `spalgnr`

- Specific alignment of long reads to assembly graph used in hybrid assembly pipeline: `spades-gmapper`
- Refinement of metagenome-assembled genomes: `binspreader`
- Alignment of profile HMMs to assembly graph: `pathracer`

**Installation:** SPAdes requires a 64-bit Linux system and Python (3.8 or higher). You can either download binaries or download source code and compile it yourself.

- When properly installed, the following files will be placed in the `bin` directory
  - `spades.py` (main executable script)
  - `metaspades.py` (main executable script for metaSPAdes)
  - `plasmidspades.py` (main executable script for plasmidSPAdes)
  - `metaplasmidspades.py` (main executable script for metaplasmidSPAdes)
  - `metaviralspades.py` (main executable script for metaviralSPAdes)
  - `rnaspades.py` (main executable script for rnaSPAdes)
  - `rnaviralspades.py` (main executable script for rnaviralSPAdes)
  - `coronaspades.py` (wrapper script for coronaSPAdes mode)
  - `spades-core` (assembly module)
  - `spades-gbuilder` (standalone graph builder application)
  - `spades-gmapper` (standalone long read to graph aligner)
  - `spades-kmercount` (standalone k-mer counting application)
  - `spades-hammer` (read error correcting module for Illumina reads)
  - `spades-ionhammer` (read error correcting module for IonTorrent reads)
  - `spades-bwa` (BWA alignment module which is required for mismatch correction)
  - `spades-corrector-core` (mismatch correction module)
- Downloading SPAdes Linux binaries: go to the directory in which you wish SPAdes to be installed
  - `wget https://github.com/ablab/spades/releases/download/v4.0.0/SPAdes-4.0.0-Linux.tar.gz`
  - `tar -xzf SPAdes-4.0.0-Linux.tar.gz`
  - `cd SPAdes-4.0.0-Linux/bin/`
  - In this case you do not need to run any installation scripts - SPAdes is ready to use. We also suggest adding SPAdes installation directory to the `PATH` variable.
- Verifying your installation: for testing purposes, SPAdes comes with a toy data set. To try SPAdes on this data set, run:
 

```
<spades installation dir>/bin/spades.py --test
```

# if you added `bin` folder from SPAdes installation directory to the `PATH` variable, you can run

```
spades.py --test
```

If the installation is successful, you will find the following information at the end of the log

```
===== Assembling finished. Used k-mer sizes: 21, 33, 55
```

```
* Corrected reads are in spades_test/corrected/
* Assembled contigs are in spades_test/contigs.fasta
* Assembled scaffolds are in spades_test/scaffolds.fasta
* Assembly graph is in spades_test/assembly_graph.fastg
* Assembly graph in GFA format is in
spades_test/assembly_graph_with_scaffolds.gfa
* Paths in the assembly graph corresponding to the contigs are
in
  spades_test/contigs.paths
* Paths in the assembly graph corresponding to the scaffolds
are
  in spades_test/scaffolds.paths

===== SPAdes pipeline finished.

===== TEST PASSED CORRECTLY.

SPAdes log can be found here: spades_test/spades.log

Thank you for using SPAdes!
```