bio-101

# Metagenomic Protocol (From Quality Control to Mapping) for Metagenome-assembled Genomes Using Anvi'o

**Soumyadev Sarkar, Tanner Richie, and Sonny T. M. Lee***

Division of Biology, Kansas State University, Manhattan, Kansas, United States
*For correspondence: leet1@ksu.edu

## Abstract

Anvi'o (advanced analysis and visualization platform for 'omics data), an open-source software, provides a visualization platform for advanced analysis of 'omics data. Anvi'o is able to analyze a range of multi-'omics data including metagenomics, genomics, pangenomics, phylogenomics, metapangenomics, metatranscriptomics, and microbial population genetics. The modular architecture of Anvi'o is different from any other existing software, which ensures interactivity, flexibility, extensibility, and most importantly, reproducibility. Using Anvi'o, users can very easily avoid rigid workflows and deal with 'omics data. Snakemake is a tool for building computational workflows. The strategy here is to combine the Anvi'o with snakemake workflow, which yields better documentation and reproducibility. Combining the snakemake with Anvi'o allows users to easily establish the workflows in any type of computer system. Further, users can also set up parallelization of several independent analysis steps. The Anvi'o circumvents the requirements of complex computational demands and resources by directly converting raw input to data products, which can easily be analyzed through Anvi'o interactive interface. This lesson will enable users to perform every step demonstrated, reproduce the results, and improvise these steps for other projects.

**Keywords:** MAGs, Anvi'o, Metagenomics, Mapping, Quality control, Snakemake

## Background

Anvi'o (advanced analysis and visualization platform for 'omics data) is an easy-to-use interface that enables users to study metagenomics (Eren et al., 2021). It is a single platform to access several parameters such as number of contigs across samples, GC-content, N50, inferred taxonomy, genome completion, and redundancy scores. Anvi'o facilitates quality control, mapping, assembly, and manual refining of metagenomics data (Eren et al., 2015). Snakemake is a readable python-based workflow engine, and a powerful execution environment (Köster and Rahmann, 2012). It can both work in workstations with single core to clusters, with no need to modify the workflows. Snakemake workflows are able to work perfectly with other tools and/or web services with specific input and output file formats. That makes snakemake very flexible to work with. The rationale of this protocol is to combine the power of Anvi'o and snakemake, which will reduce the constraints of complex 'omics data analysis. The combination of Anvi'o and snakemake will guarantee that users can work with more flexibility, the main advantage over other existing protocols. Also, the interactive interface of Anvi'o can facilitate users to easily analyze the data. Users can customize their workflow with config files using snakemake-Anvi'o workflows (Mölder et al., 2021).

## Software

1. Anvi'o v7 (https://merenlab.org/software/anvio/)
2. Snakemake (https://snakemake.readthedocs.io/en/stable/)

Installing Anvi'o

(1) Conda setup: If the conda is not installed in the system, it is necessary to open a terminal such as iTerm.
*Command:*
*conda install*

To verify whether you already have conda installed, copy and paste the following command into your terminal:
*Command:*
*conda --version*

Always make sure that you work in an up-to-date conda environment by using the following command:
*Command:*
*conda update conda*

(2) Anvi'o environment setup
Create a new conda environment using the command:
*Command:*
*conda create -y --name anvio-7.1 python=3.6*

Then, activate it using the command:
*Command:*
*conda activate anvio-7.1*

(3) Installing Anvi'o
The first step is to download the python source package for the Anvi'o release using the following command:
*Command:*
*curl -L https://github.com/merenlab/anvio/releases/download/v7.1/anvio-7.1.tar.gz *
*--output anvio-7.1.tar.gz*

Then, use the following command to install Anvi'o:
*Command:*

*pip install anvio-7.1.tar.gz*

Users should note that the installation of Anvi'o is user friendly but may take a long time to finish and is computationally intensive.

## Data availability

1. The data can be accessed at https://figshare.com/s/35ea294e2671d75f1d5c and https://github.com/Bio-protocol/bioprotocol_2104071.

## Case study

### A. Input data

Anvi'o workflows help users to:

1. Streamlining standard repetitive steps of 'omics data analysis like assembly, mapping, mapping results profiling, annotation of functions/taxonomy, and generating Anvi'o databases in a scalable form.
2. Ask biological questions about the data.
3. Describe the data and the results easily to the scientific community.

Anvi'o uses the program *anvi-run-workflow* to run the workflows. For a particular workflow, the program will help users to prepare the config file.

The following code asks the program what workflows it knows:

*Command*
*anvi-run-workflow --list-workflows*
*Available workflows ....: contigs, metagenomics, pangenomics, phylogenomics, trnaseq*

After you have decided the process you wish to use, the config file allows you to change the parameters and order of steps associated with that process. Even if you are satisfied with all the default parameters, the config file is required for all workflows. Users should ensure that the config files are proper, and preparing a config file for a particular workflow could be challenging. The following code will help users to generate a config file:

*Command*
*anvi-run-workflow -w WORKFLOW-NAME *
*--get-default-config OUTPUT-FILE-NAME*

The --get-default-config will generate a default config file for a workflow that you can modify. Configurable flags and parameters will be contained in this file. You can either leave as is any parameters that you do not intend to change, or you can remove those that you do want to change from your config file to make it shorter and cleaner.

There are three configurations in the config file:

1. **General workflow parameters:** You will need a name for your project and the workflow mode you want to employ.

2. **Parameters:** Parameters that are exclusively applicable to a single rule, such as the Anvi'o profiling steps' minimum contig length. Each program has unique parameters. Users should make sure to use parameters that appear in the config file that would be identical to the names used in the particular program. For instance, if there are multiple ways to use adjustable parameters or arguments, users should use the longer one. As an example, *anvi-run-hmms* are able to accept with *-H* or *--hmm-profile-dir* parameters that specify the directory path of HMM profile. However, users are only allowed to use *--hmm-profile-dir* in the config file.
3. **Names of the output directory:** This regards how Anvi'o will deal with output directories and files.

## B. Samples.txt

The samples.txt file is for associating sample names with raw sequencing reads. There should be three or four columns (plus the optional groups column) in the samples.txt, with each column separated from the others by a TAB character. The following column names should be included in the header:

1. **Sample:** A name for each of your metagenomic samples.
2. **r1 and r2**: These two columns include the path (which could be relative or absolute; absolute paths are always preferred) to the FASTQ files corresponding to the sample.

It may additionally include the following column as an option:

3. **Group:** While binning genomes from metagenomic assemblies, one of the strategies is to combine numerous samples. This column's function is to specify which samples will be co-assembled. This is an optional column; if it is not present in the samples.txt file, each sample will be assembled independently. Only the samples utilized for the co-assembly would be mapped to the resultant assembly by default. You can co-assemble groups of samples, but you must then map all samples to each assembly.

## C. Workflow

Raw paired-end sequencing reads for shotgun metagenomes are the **default entry point** into the metagenomics workflow. The workflow's **default endpoint** is a merged profile database ready for bin refinement, as well as an annotated Anvi'o contigs database. The steps in the workflow are as follows:

1. Using illumina-utils, quality-check metagenomic short reads and generate a thorough report for the outcomes of this step: quality-check to be performed by removing the low quality reads according to the criteria mentioned in Minoche et al. (2011), the combination of B-tail trimming and passed chastity filter to be used, and reads to be removed that contained uncalled bases.
2. Select programs for generating taxonomic profiles of short reads. These profiles are imported to individual databases of profiles that are available in the merged profile database of Anvi'o.
3. Using megahit and/or idba_ud and/or metaspades, assemble quality-filtered metagenomic reads.
4. Using anvi-gen-contigs-database, create an Anvi'o contigs database using assembled contigs. The contigs database should have annotations of the functions, taxonomy, and HMMs.
5. Using bowtie2, map short reads from metagenomes to contigs and then generate indexed and sorted BAM files.
6. Using anvi-profile to produce single Anvi'o profiles from individual BAM files.
7. Using anvi-merge, merge to generate single Anvi'o profiles.

You will only need a samples.txt file and a few FASTQ files. We will go through a mock example with three small metagenomes in this section. A limited number of reads were selected to create these metagenomes. The following samples.txt file can be found in your working directory:

| *sample* | *group* | *r1* | *r2* |
|---|---|---|---|
| *P1* | *G01* | *M-1_S21_L001_R1_001.fastq.gz* | *M-1_S21_L001_R2_001.fastq.gz* |
| *P2* | *G02* | *M-2_S22_L001_R1_001.fastq.gz* | *M-2_S22_L001_R2_001.fastq.gz* |
| *P3* | *G03* | *M-3_S23_L001_R1_001.fastq.gz* | *M-3_S23_L001_R2_001.fastq.gz* |

This file details the raw paired-end reads locations for the samples and 'groups'. The default name is samples.txt for the samples_txt file, but you can change it in the config file.

Let's have a look at the config file config-megahit.json in the working directory.

```
{
    "workflow_name": "metagenomics",
    "config_version": "2",
    "samples_txt": "samples.txt",
 "megahit": {
        "--min-contig-len": 1000,
        "--memory": 0.4,
        "threads": 7,
        "run": true,
    }
}
```

Every customizable parameter will be given a default value. We normally start with a default config file and delete every line that we do not want to keep. We have everything now to start. Let's generate a workflow graph at this stage. The following code will generate a workflow graph:

*Command*

```
anvi-run-workflow -w metagenomics \
                  -c config-megahit.json \
                  --save-workflow-graph
```

We can now start the workflow:

*Command*

```
anvi-run-workflow -w metagenomics \
                  -c config-megahit.json
```

After completing all the steps in this pipeline, users will be able to utilize these generated profiles for downstream work like manual refining and functional analyses.

## Result interpretation

This workflow explained the steps from quality control to mapping for generation of metagenome-assembled genomes. A general introduction was provided about the config and samples.txt files to connect raw sequencing reads with sample details. Anvi'o was coupled with snakemake workflows to generate profiles that could be used for downstream analyses.

## Discussion

The concept of using snakemake with Anvi'o was to have better documentation and reproducibility of the entire work. Snakemake also allows the Anvi'o workflow to be more specific using config files. There are opportunities to repurpose this existing workflow to user's own projects.

## Acknowledgments

Soumyadev Sarkar acknowledges the National Science Foundation EPSCoR for his research grant. This protocol is based on using Anvi'o (Eren et al., 2015 and 2021). This material is based upon work supported by the National Science Foundation under Award No. OIA-1656006 and matching support from the State of Kansas through the Kansas Board of Regents.

## Competing interests

The authors declare no competing interests.

## References

Eren, A. M., Esen, O. C., Quince, C., Vineis, J. H., Morrison, H. G., Sogin, M. L. and Delmont, T. O. (2015). Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ* 3: e1319.

Eren, A. M., Kiefl, E., Shaiber, A., Veseli, I., Miller, S. E., Schechter, M. S., Fink, I., Pan, J. N., Yousef, M., Fogarty, E. C., et al. (2021). Community-led, integrated, reproducible multi-omics with anvi'o. *Nat Microbiol* 6(1): 3-6.

Köster, J. and Rahmann, S. (2012). Snakemake--a scalable bioinformatics workflow engine. *Bioinformatics* 28(19): 2520-2522.

Minoche, A. E., Dohm, J. C. and Himmelbauer, H. (2011). Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome Biol* 12(11): R112.

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., et al. (2021). Sustainable data analysis with Snakemake. *F1000Res* 10: 33.