HHS Public Access

Author manuscript

Curr Protoc. Author manuscript; available in PMC 2023 February 08.

Published in final edited form as:

Curr Protoc. 2022 December ; 2(12): e604. doi:10.1002/cpz1.604.

Genotyping Microbial Communities with MIDAS2: From Metagenomic Reads to Allele Tables

Chunyu Zhao[1,2,5], **Miriam Goldman**[2,3,5], **Byron J. Smith**[2,4], **Katherine S. Pollard**[1,2,4,6]

[1]Data Science, Chan Zuckerberg Biohub, San Francisco, California

[2]Data Science and Biotechnology, Gladstone Institutes, San Francisco, California

[3]Biomedical Informatics, University of California San Francisco, San Francisco, California

[4]Epidemiology and Biostatistics, University of California San Francisco, San Francisco, California

[5]These authors contributed equally to this work.

Abstract

The Metagenomic Intra-Species Diversity Analysis System 2 (MIDAS2) is a scalable pipeline that identifies single nucleotide variants and gene copy number variants in metagenomes using comprehensive reference databases built from public microbial genome collections (metagenotyping). MIDAS2 is the first metagenotyping tool with functionality to control metagenomic read mapping filters and to customize the reference database to the microbial community, features that improve the precision and recall of detected variants. In this article we present four basic protocols for the most common use cases of MIDAS2, along with supporting protocols for installation and use. In addition, we provide in-depth guidance on adjusting command line parameters, editing the reference database, optimizing hardware utilization, and understanding the metagenotyping results. All the steps of metagenotyping, from raw sequencing reads to population genetic analysis, are demonstrated with example data in two downloadable sequencing libraries of single-end metagenomic reads representing a mixture of multiple bacterial species. This set of protocols empowers users to accurately genotype hundreds of species in thousands of samples, providing rich genetic data for studying the evolution and strain-level ecology of microbial communities. © 2022 The Authors. Current Protocols published by Wiley Periodicals LLC.

Basic Protocol 1: Species prescreening

Basic Protocol 2: Download MIDAS reference database

[6]Corresponding author: katherine.pollard@gladstone.ucsf.edu.

Author Contributions

Chunyu Zhao: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, writing original draft, writing review and editing. **Miriam Goldman**: Data curation, formal analysis, investigation, methodology, validation, writing original draft. **Byron Smith**: Conceptualization, investigation, methodology, writing original draft, writing review and editing. **Katherine Pollard**: Conceptualization, funding acquisition, investigation, project administration, resources, supervision, writing original draft, writing review and editing.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

copy number variant; genotyping; microbiome; single nucleotide variant; strain; MIDAS2

INTRODUCTION

This protocol describes how to use the Metagenomic Intra-Species Diversity Analysis System 2 (MIDAS2; Zhao, Dimitrov, Goldman, Nayfach, & Pollard, 2022a) to genotype the species present in microbial communities using shotgun metagenomic data, a bioinformatics procedure known as metagenotyping. Most microbial species harbor immense intraspecific genetic variation, in the form of single nucleotide variants (SNVs), gene copy number variants (CNVs), and other structural variants (Garud & Pollard, 2020; Shoemaker, Chen, & Garud, 2022; Van Rossum, Ferretti, Maistrenko, & Bork, 2020). These variants are detectable in metagenomic data, which is comprised of DNA sequencing reads sampled from the pool of genomes in a microbial community. With accumulating evidence that genetic differences influence strain ecology and function, metagenotyping has gained popularity (Ghazi, Munch, Chen, Jensen, & Huttenhower, 2022). MIDAS2 is a software tool and accompanying databases for metagenotyping a set of samples and merging the results across samples to infer the set of SNVs and CNVs present in the population.

Most metagenotyping pipelines identify variants based on aligning reads to reference databases of whole genomes and/or gene sequences. It is important to use a customized yet comprehensive reference database. For a non-comprehensive reference database, species in the sample but missing from the reference database cannot be genotyped, causing false negative results. Conversely, for a non-representative reference genome, closely related species in the reference database but not in the sample may compete for reads, reducing read alignment uniqueness and even generating “phantom” metagenotypes when reads from another species are incorrectly aligned (Zhao, Zhou, & Pollard, 2022b). MIDAS2 combats these problems by selecting genomes from a comprehensive reference database to build a sample-specific reference database customized to species that are present in the samples (adjustable to scientific objectives), and by tuning alignment and filtering parameters.

MIDAS2 performs metagenotyping of either SNVs (Basic Protocol 3) or CNVs (Basic Protocol 4). Before running either of these modules, species that are detectable in the metagenomic data are first determined (Basic Protocol 1) and a MIDAS Reference Database (MIDASDB) is downloaded (Basic Protocol 2). Users also have the option

of building this reference database locally from a custom genome collection (Support Protocol 3). Subsequently, the SNV and CNV module uses the selected species to build a customized Bowtie2 (Langmead & Salzberg, 2012) index. This index is composed of representative genomes (rep-genome) for the SNV module or pangenomes (pan-genome) for the CNV module. Following reference database customization, each module consists of two sequential steps: Single-sample and across-sample analysis. In the first step, metagenomic reads from each sample are aligned to the rep-genome Bowtie2 index to call alleles in the read pileups (SNV module) or to the pan-genome Bowtie2 index to estimate gene copy numbers (CNV module). In the second step, results are merged across samples and population variants are called. Support Protocol 4 describes how pipeline parameters can be adjusted to mitigate alignment and genotyping errors and accomplish specific scientific objectives.

Details about the output files generated by these workflows are included in each protocol as well as in the Guidelines for Understanding Results. The MIDAS2 installation process and command line interface are described in Support Protocols 1 and 2, respectively. Collectively, the information in this article empowers users to generate accurate genotypes for many microbial species in a set of samples in parallel. The analysis process starts from high-quality shotgun metagenomic reads and ends with tables of population SNV and/or CNV genotypes for each abundant and prevalent species.

Example input files used in this article can be found at Zenodo (https://zenodo.org/record/6774633/). Example custom genome collections can be found at Zenodo (https://zenodo.org/record/6774976/). The complete example output files can be found at Zenodo (https://zenodo.org/record/6775263/).

STRATEGIC PLANNING

Hardware and Software

MIDAS2 is a command-line tool that requires a 64-bit Linux system with at least 16 GB of RAM. Support Protocol 1 details the steps for installing the MIDAS2 software. Hardware resources—CPU, RAM, and disk—are all limiting factors for whole-genome read mapping based metagenotyping, depending on the scale and microbiome complexity of the input dataset. Metagenotyping hundreds of samples could easily utilize hundreds of gigabytes of disk space and RAM as well if processed in parallel across many CPUs. In a study with 200 samples, for example, we recommend a compute environment with 64 vCPUs, 256 GB RAM (e.g., EC2 instance m5.16xlarge) and 2 TB disk space. The most computationally intensive parts of the single-sample step of the SNV or CNV module are building the customized Bowtie2 index and read alignment (together, ~75% of runtime). The subsequent calling of population SNVs across samples is compute intensive. But this step scales linearly with increasing numbers of CPUs (Zhao et al., 2022a) so we recommend using as many CPUs as feasible.

Paired-End Sequencing Reads

The most common inputs for MIDAS are paired-end, short-read (e.g., Illumina), high-quality metagenomics shotgun sequencing read files in the FASTQ-format, compressed with Gzip. Paired-end reads are sequenced from both ends of a DNA fragment (insert). Each Illumina sequencing run produces paired-end reads in two files: *_1.fastq.gz contains the forward orientation reads, and *_2.fastq.gz contains the reverse orientation reads. Make sure for each forward read from *_1.fastq.gz, the corresponding paired read from *_2.fasta.gz is placed in the corresponding paired file on the same line. During preprocessing and quality control (e.g., adapter trimming) performed prior to running MIDAS2, it is important to maintain this correspondence and read orientation.

Unpaired Sequencing Reads

Users may alternatively provide single-end, high-quality sequencing reads to MIDAS2.

Quality Control

Quality control (QC) of input sequencing data is recommended. A typical QC pipeline includes multiple steps that remove adapter sequences and filter out low-quality bases, low complexity reads, and contamination (e.g., host and PhiX) reads (Clarke et al., 2019).

BASIC PROTOCOL 1: SPECIES PRESCREENING

Reference-based metagenotyping depends crucially on the choice and customization of reference database. Therefore, a typical MIDAS2 workflow starts with a species prescreening step for each metagenome, which enables customization of the reference database to match the species in the sample. This protocol describes the species selection step: Estimating species coverage per sample, merging the single-sample profiling results, and generating a list of species confidently detected in at least one sample. MIDAS2 estimates species coverage per sample by aligning reads to a database of sequences of fifteen universal, single-copy genes (SCGs) and using the median (or mean) coverage of each species' SCGs. The goal of species pre-screening is to determine which species are abundant and prevalent enough to be metagenotyped. Users can adjust parameters in order to be stricter or more inclusive. The default values are based on precision and recall in metagenomic simulations. Including rarer species may reduce metagenotype accuracy but is justified in some applications (e.g., with a defined community).

Necessary Resources

Hardware—64-bit Linux system with at least 16 GB of RAM (RAM and disk consumption depend largely on the size of the input data; installation instructions provided in Support Protocol 1)

Software—Metagenomic Intra-Species Diversity Analysis System 2 (MIDAS2; see Support Protocol 1 for installation)

AWK command

Input Files—Paired-end or single-end metagenomic sequencing data in FASTQ format, optionally compressed with Gzip [see Strategic Planning; to demonstrate, we have deposited two single-end HMP mock community samples (Truong, Tett, Pasolli, Huttenhower, & Segata, 2017) on Zenodo: https://zenodo.org/record/6774633]

1. Install MIDAS2 as described in Support Protocol 1.

2. Create a work directory containing the FASTQ files (here example input files are downloaded from Zenodo):

```
mkdir midas2_protocol
cd midas2_protocol
wget https://zenodo.org/record/6774633/files/reads.zip
unzip reads.zip
```

   We use the same example input files for all the protocols in this article and all the analyses are generated relative to the `midas2_protocol` root directory.

3. Initialize a local copy of a MIDASDB-UHGG. Here the SCG data is downloaded:

```
midas2 database --init --midasdb_name uhgg \
;--midasdb_dir midasdb_uhgg
```

   The above command downloads the SCG marker gene databases needed for species profiling analysis in this protocol to the local MIDASDB directory `midasdb_uhgg/`. The SCGs from the MIDASDB-GTDB can be downloaded instead of those from UHGG by adjusting the above command to point to that database with `--midasdb_name gtdb` and a different choice of directory to store the database, such as `--midasdb_dir midasdb_gtdb`. The download files take up 1.2 GB.

   During the construction of a MIDASDB, six-digit numeric species identifiers (`species_id`) are randomly assigned. The taxonomic assignment of these `species_id` is stored in the `metadata.tsv file`. More details about downloading other components of a MIDASDB are in Basic Protocol 2. In addition, Support Protocol 3 contains the steps for creating a MIDASDB from a user-provided collection of genomes.

   We will use the same work directory `midas2_protocol`/ and database subdirectory `midasdb_uhgg`/ for all analyses.

4. Run the single-sample species analysis to identify confidently detectable (i.e., relatively abundant) species in each sample, looping through samples.

```
for sample_name in SRR172902 SRR172903
```

```
do
midas2 run_species --sample_name ${sample_name} \
−1 reads/${sample_name}.fastq.gz \
--midasdb_name uhgg --midasdb_dir midasdb_uhgg \
--num_cores 4 midas2_output
done
```

Here four CPU cores are used for each sample. Note that more or fewer CPUs can be used by specifying the number with `--num_cores`. Also note that each sample could be run simultaneously.

Output files are created automatically in `midas2_output/SRR172902/species/`and `midas2_output/SRR172903/species/`.

5. Confirm `run_species` has finished successfully. Once the species profiling step for all the samples is complete without any reported error, check for the output files:

   - `species_profile.tsv`: A six-column, tab-separated file, describing the coverage of each species' marker genes in the sample. The columns are unique species identifier (`species_id`), total aligned read counts (`marker_read_counts`), median marker coverage (`median_marker_coverage`), mean marker coverage (`marker_coverage`), estimated relative abundance based on marker genes (`marker_relative_abundance`), and fraction of uniquely aligned markers (`unique_fraction_covered`). There is one row per species in the MIDASDB that is covered by at least two reads.

   For example, the `median_marker_coverage` of `species_id=102344` for sample SRR172902 is 10.819588.

   Both the SNV and CNV module of MIDAS2 will use the `species_profile.tsv` file to select species to be genotyped. We can look at the list of relatively low abundance species with relatively high confidence in each sample (`median_marker_coverage` > `0` and `unique_fraction_covered` > `0.6`) with the following commands:

```
$awk ‘$3>0 && $6>0.6’ midas2_output/SRR172902/species/
species_profile.tsv | grep -v species_id | wc -l
18
$awk ‘$3>0 && $6>0.6’ midas2_output/SRR172903/species/
species_profile.tsv | grep -v species_id | wc -l
8
```

There are eighteen species for SRR172902 and eight species for SRR172903 meeting our selection criterion.

6. Prepare the sample manifest file for the purpose of merging metagenotyping results across samples in the SNV and CNV modules. This file has two, tab-delimited columns. The first column specifies the sample identifier (`sample_name`) and the second column provides the output directory provided to `run_species` (`midas2_output` as in step 5).

   The following is one way to generate the sample manifest file for SRR172902 and SRR172903.

```
echo -e “sample_name\tmidas_outdir” > list_of_samples.tsv
ls reads | awk -F ‘.’ ‘{print $1}’ | awk -v OFS=‘\t’ ‘{print $1,
“midas2_output”}’ >> list_of_samples.tsv
```

   Out sample manifest file, `list_of_samples.tsv`, now looks like:

```
$cat list_of_samples.tsv
sample_name midas_outdir
SRR172902 midas2_output
SRR172903 midas2_output
```

   Based on this file, the `merge_species` command expects to locate the `midas2_output/SRR172902/species/species_profile.tsv` file generated by the `run_species` command for SRR172902, and similarly for SRR172903.

7. Merge species profiling results for the samples listed in the `list_of_samples.tsv`.

```
midas2 merge_species --samples_list list_of_samples.tsv \
--min_cov 0.01 midas2_output/merge
```

   The `--min_cov` flag defines the minimum `median_marker_coverage` for estimating species prevalence: Present if `median_marker_coverage` >=`min_cov`. Here we want to generate the full list of species with positive SCG profiling results. The minimal reported positive `median_marker_coverage` is 0.164444 (SRR172902). Therefore, we set `--min_cov` to 0.01. Users can also set this to a smaller number (e.g., 0.0000001) to include lower abundance species, with the caveat that some are likely to be false positives.

   The output files are created automatically under the directory `midas2_output/merge/species/`.

8. The species merging steps have finished successfully if the output files are created and MIDAS2 reports no errors. The primary output files are:

- `species_prevalence.tsv`: A six-column, tab-delimited file, summarizing the species profiling results. The columns are species identifier, median marker abundance, mean marker abundance, median marker coverage, mean marker coverage, and number of samples with the species present. There is one row per species.
- Five species-by-sample matrices in the same output directory.

The list of all the generated files can be found in Guidelines for Understanding Results.

9. It is worth noting that the species module is designed to select species with sufficient coverage for metagenotyping, rather than performing a comprehensive taxonomic profiling. To this end, we collect the list of species confidently detected in at least one sample:

```
awk '$6>0 {print $1}' midas2_output/merge/species/
species_prevalence.tsv | grep -v species_id > list_of_species.tsv
```

With `list_of_species.tsv` in hand, we can now download a subset of the MIDASDB in Basic Protocol 2.

BASIC PROTOCOL 2: DOWNLOAD MIDAS REFERENCE DATABASE

This protocol describes how to download all or part of a MIDASDB, a set of custom files constructed from microbial genome sequences and containing all the information needed to metagenotype the species detected in a set of shotgun-metagenomic samples. MIDAS2 provides two prebuilt MIDASDBs sourced from large, public microbial genome collections: MIDASDB-UHGG (4644 species; 286,997 genomes) based on the Unified Human Gastrointestinal Genome catalog (v1; Almeida et al., 2021) and MIDASDB-GTDB (47,893 species; 258,405 genomes) based on the Genome Taxonomy Database (v202; Parks et al., 2021). MIDASDB-UHGG is only suitable for human gut metagenomics samples, while MIASDB-GTDB can be used for metagenomic samples from various environments. Support Protocol 3 describes how to build a new MIDASDB locally from a custom genome collection. This is particularly useful if users plan to run an assembly pipeline and use the assembled contigs/scaffolds (i.e., metagenome assembled genomes or MAGs) as the reference database for MIDAS2. A MIDASDB should be downloaded or built before any other MIDAS2 protocols can be run.

There are three components in an MIDASDB: SCGs, rep-genome, and pan-genome. Each species contributes sequences to all three components. By preloading the MIDASDB, individual calls to MIDAS2 commands do not need to automatically download the necessary files. As a result, with a preloaded MIDASDB, per-sample analyses can be run in parallel without a risk of processes interfering with one another.

Necessary Resources

Hardware—A 64-bit Linux system with at least 16 GB of RAM (installation instructions are provided in Support Protocol 1)

Software—MIDAS2 (see Support Protocol 1 for installation)

Input Files—Optional list of species for download

1. Optional: List pre-built MIDASDBs.

```
$midas2 database --list
uhgg 286997 genomes from 4644 species version 1.0
gtdb 258405 genomes from 47893 species version r202
```

   For this protocol, we will use the MIDASDB-UHGG as an example. A prebuilt GTDB database is also available.

2. Initialize a local copy of MIDASDB-UHGG. This is a required first step for downloading any MIDASDB. This is the same command as step 3 in Basic Protocol 1. Skip this step if you already ran Basic Protocol 1.

```
midas2 database --init --midasdb_name uhgg --midasdb_dir
midasdb_uhgg
```

   This command creates the local directory `midasdb_uhgg/` if it doesn't exist and downloads the following files and/or directories:

   - `genomes.tsv`: The table-of-contents file assigning genomes to species and denoting the representative genome for each species.
   - `metadata.tsv`: Tab-delimited table specifying the six-digit numeric species identifiers (`species_id`) with taxonomic assignments.
   - `md5sum.json`: md5sum cache for database files; used internally by MIDAS2 during downloading.
   - `markers/`: SCG data needed for species prescreening.
   - `markers_models/`: SCG profile hidden Markov model.
   - `chunks/`: design cache for parallelizing the SNV module over partitions of the representative genome ("chunks").

3. Optional: Download the entire MIDASDB.

   This requires a large amount of data transfer and storage: 93 GB for MIDASDB-UHGG and 539 GB for MIDASDB-GTDB. Because MIDAS2 only uses database information for detected species (Basic Protocol 1), it is usually

unnecessary to download the entire MIDASDB. We recommend against this option in most cases.

```
midas2 database --download --midasdb_name uhgg --midasdb_dir
midasdb_uhgg --species all
```

4. Customized MIDASDB downloading:

   Users can take advantage of the MIDAS2 species-level database structure to download and decompress only the necessary portions of a MIDASDB. For example, in Basic Protocol 1, we collected the list of 22 species present in at least one sample (`list_of_species.tsv`). Now we can download database components (both rep-genome and pan-genome) only for these 22 species.

   ```
   midas2 database --download --midasdb_name uhgg --midasdb_dir
   midasdb_uhgg --species_list list_of_species.tsv
   ```

   The downloaded MIDASDB-UHGG files sufficient for the analysis of samples containing these 22 species are only 4.2 GB.

5. The download has completed successfully when the command `midas2 database --download` finishes and no error is reported.

BASIC PROTOCOL 3: POPULATION SINGLE NUCLEOTIDE VARIANT CALLING

This protocol describes the SNV module of MIDAS2, which takes as input metagenomic sequencing reads from a set of samples and generates files with SNV genotypes for each sample for all detected species. The SNV module has two steps: (1) single-sample allele tallying with the `midas2 run_snps` command and (2) population SNV calling with the `midas2 merge_snps` command. Basic Protocols 1 (Species) and 2 (MIDASDB) should be run before this protocol. These SNV outputs can be used for downstream analyses with other tools or user-supplied scripts, including for tracking and transmission studies, strain deconvolution, and evolutionary analyses. Users can modify default parameters in order to make post-alignment filtering stricter or more inclusive, which alters which sites and samples are genotyped for each species. They may also alter which species are genotyped and how the module is parallelized.

Necessary Resources

Hardware—A 64-bit Linux system with at least 16 GB of RAM (installation instructions are provided in Support Protocol 1)

Software—MIDAS2 (see Support Protocol 1 for installation)

Input Files—Paired-end or single-end metagenomic sequencing data in FASTQ format, optionally compressed with Gzip (see Strategic Planning)

Outputs of Basic Protocol 1

Outputs of Basic Protocol 2

NOTE: The RAM and disk consumption depend largely on the size of the input data.

1. Perform species prescreening as described in Basic Protocol 1. We recommend users look at the SCG-based species profiling results (`species_profile.tsv`), particularly the following two columns: `median_marker_coverage` and `unique_fraction_covered`. Users can adjust these two parameters for selecting the list of species to be genotyped in step 3 below. Higher values indicate more reads mapping to the marker genes, and these parameter values can be adjusted upward if the user is hesitant to metagenotype low abundance species or downward if they wish to include more species at the cost of potentially lower accuracy.
2. Execute the `run_snps` command for each sample. Conceptually, a typical invocation of the `run_snps` command proceeds by four steps:
   a. Select the list of species for accurate metagenotyping based on the species profiling results and user-defined species selection criterion. Taking SRR172902 as an example, `run_snps` expects to find the species profiling results at `midas2_output/SRR172902/species/species_profile.tsv`.
   b. Compile the representative genomes for these species and build a sample-customized rep-genome Bowtie2 index.
   c. Align reads to this index with Bowtie2.
   d. Output a read alignment summary and pileup result for each species.

```
for sample_name in SRR172902 SRR172903
do
midas2 run_snps \
--sample_name ${sample_name} \
–1 reads/${sample_name}.fastq.gz \
--midasdb_name uhgg --midasdb_dir midasdb_uhgg \
--select_by median_marker_coverage,unique_fraction_covered \
--select_threshold=0,0.6 \
--num_cores 8 midas2_output
done
```

The number of CPUs used is specified via `--num_cores 8`. This step can also be parallelized over multiple samples (e.g., using shell background processes or `xargs`).

For each sample, the above command performs SNV calling for all species meeting the user-defined species filtering criteria: `median_marker_coverage>0` and `unique_fraction_covered>0.6`. This means a species is metagenotyped if the sequences of its fifteen SCGs have a median of at least two unique aligned reads and 60% horizontal coverage. The higher the cutoff of these parameters, the smaller the list of selected species will be. This is because only highly abundant species are selected with high parameter values. For examples, see step 5 of Basic Protocol 1. We recommend users set `unique_fraction_covered` to no lower than 0.5.

The output files are generated automatically under the directory `midas2_output/SRR172902/snps/` and `midas2_output/SRR172903/snps/`.

3. Confirm `run_snps` has finished successfully. Once the single-sample SNV analysis is complete without any reported error, check for the output files:

   - `snps_summary.tsv`: an eight-column, tab-delimited file containing a summary of read alignment and pileup for all the species in the Bowtie2 index. Among all the reported columns, horizontal genome coverage (`fraction_covered`, the fraction of bases covered by at least one read) and vertical genome coverage (`mean_coverage`, the average read depth across all the bases covered by at least one read), are particularly useful.

   It is important to note that MIDAS2 purposely holds off on any species selection or site filtering with the single-sample pileup results until across-samples SNV analyses are performed. Therefore, the number of reported per-species pileup results for each sample is the same as the number of species passing the SCG-based selection (eighteen for SRR172902 and eight for SRR172903).

   In step 5, across-samples SNV analysis will filter species based on the single-sample horizonal genome coverage (`fraction_covered`) and vertical genome coverage (`mean_coverage`). Using this file, we can observe that there are ten species with `fraction_covered` >= `0.4` for SRR172902 and five species for SRR172903:

```
$awk ‘$7>=0.4’ midas2_output/SRR172902/snps/snps_summary.tsv |
grep -v species_id | wc -l
10
$awk ‘$7>=0.4’ midas2_output/SRR172903/snps/snps_summary.tsv |
grep -v species_id | wc -l
5
```

- `<species_id>.snps.tsv.lz4`: Per-species reads pileup for all the species in the rep-genome Bowtie2 index. Positions are filtered to all genomic sites in the reference genome covered by at least two reads. These single-sample pileup files are the input to the `midas2 merge_snps` command, which calls population SNVs across samples. They can also be used to call SNVs in individual samples if desired.

4. Prepare sample manifest file for merging pileup results across samples. We can use the same file `list_of_samples.tsv` generated by step 6 in Basic Protocol 1.

5. Upon the completion of `run_snps` for all the samples in the file `list_of_samples.tsv`, MIDAS2 compute the per-species population SNVs with the `merge_snps` command. There are four main steps for each species:

   a. Sample selection. Just because shotgun metagenomics reads aligned to one genome does not guarantee the presence of that species in the sample. Therefore, it is common practice to select <species, sample> pairs based on the horizontal genome coverage (`genome_coverage`) and vertical genome coverage (`genome_depth`). Higher values restrict the list of species metagenotyped. In this protocol, we want to genotype low abundance species (`genome_depth` > `0.1X`) with relatively high confidence (`genome_coverage` > `0.4`). Users should adjust these parameters based on their own research objectives.

   b. For each genomic site in the representative genomes, MIDAS2 determines the set of alleles present across all samples where the species is detected.

   c. For each genomic site, population major and minor alleles are then identified based either on the accumulated reads counts or sample counts in step 5b (above). The population major allele is the allele with highest frequency across samples and the population minor allele is the second most frequent. In the case of ties, the alphabetically first allele is the major allele.

   d. Finally, MIDAS2 reports the vertical coverage (read depth) and population minor allele frequency of each site in each sample.

```
midas2 merge_snps --samples_list list_of_samples.tsv \
--midasdb_name uhgg --midasdb_dir midasdb_uhgg \
--genome_coverage 0.4 --genome_depth 0.1 --sample_counts 2 \
--snp_type bi --num_cores 8 midas2_output/merge
```

This command selects species present in both samples with horizontal genome coverage >40% and average vertical genome coverage >0.1×. There are four species meeting these selection criteria. By default, MIDAS2 reports all types of non-fixed population alleles. In this protocol, because we only have two samples,

there cannot be any sites with three or four alleles. Hence, we choose to report only bi-allelic SNVs (`--snp_type bi`). The number of CPUs used is specified via `--num_cores 8`. These parameters are all adjustable.

The output files are generated automatically under the directory: `midas2_output/merge/snps/`.

6. Optional: See additional parameters.

   MIDAS2 has many additional, optional parameters not shown in the above example. These and the parameters shown above can be used to alter the behavior of the SNV module and to tailor the analysis to the microbial communities in the samples. Support Protocol 4 describes parameter choices, including parameters that control post-alignment filtering, species selection, and site selection.

7. Population SNV analysis has finished successfully when all the following output files are created under the directory `midas2_output/merge/snps/` without any error message.

   - `snps_summary.tsv`: Merged single-sample pileup summary containing information such as horizontal genome coverage (`fraction_covered`) and vertical genome coverage (`mean_coverage`) for each sample.

   For each species passing the species selection filter, information about SNVs identified across samples are organized by `species_id`, with three LZ4 files per subdirectory:

   - `<species_id>/<species_id>.snps_info.tsv.lz4`: Metadata of population SNVs (e.g., biological annotations);
   - `<species_id>/<species_id>.snps_allele_freq.tsv.lz4`: Site-by-sample matrix of population minor allele frequencies;
   - `<species_id>/<species_id>.snps_depth.tsv.lz4`: Site-by-sample read depth matrix.

     More information on how to interpret the population SNV results can be found in Guidelines for Understanding Results.

BASIC PROTOCOL 4: PAN-GENOME COPY NUMBER VARIANT CALLING

This protocol describes the CNV module of MIDAS2, which takes as input metagenomic sequencing reads from a set of samples and generates files with CNV genotypes for each sample for all detected species. There are two steps for population CNV calling: (1) single-sample quantification of copy number for each gene in the pangenome of each species with the `midas2 run_genes` command and (2) population CNV calling with the `midas2 merge_genes` command. Basic Protocols 1 (Species) and 2 (MIDASDB) should be run before this protocol. These CNV outputs can be used for downstream analyses with other tools or user-supplied scripts, including for tracking and transmission studies, evolutionary

analyses, and testing for associations with traits of the microbes, their environments, or their hosts. Users can modify default parameters in order to alter which species, sites, and samples are genotyped.

Necessary Resources

Hardware—A 64-bit Linux system with at least 16 GB of RAM (installation instructions are provided in Support Protocol 1)

Software—MIDAS2 (see Support Protocol 1 for installation)

Input Files—Paired-end or single-end metagenomic sequencing data in FASTQ format, optionally compressed with Gzip (see Strategic Planning)

Outputs of Basic Protocol 1

Outputs of Basic Protocol 2

NOTE: The RAM and disk consumption depend largely on the size of the input data.

1. Perform species prescreening as described in Basic Protocol 1.
2. Download MIDASDB as described in Basic Protocol 2.
3. Execute the `run_genes` command for each sample. Conceptually, a typical invocation of the `run_genes` command proceeds by five steps:
    a. Select the list of species abundant enough for accurate metagenotyping based on the species profiling results and user-defined species selection criterion. Taking SRR172902 as an example, `run_genes` expect to find the species profiling results at `midas2_output/SRR172902/species/species_profile.tsv`.
    b. Compile the pangenomes for these species and build a sample-customized pangenome Bowtie2 index.
    c. Align reads to this index with Bowtie2.
    d. For each gene in the pan-genome, normalize gene coverage by the mean coverage of all that species' SCGs to estimate copy number per cell (Parks et al., 2021).
    e. Output a read mapping summary and CNV estimates for each species.

```
for sample_name in SRR172902 SRR172903
do
midas2 run_genes \
--sample_name ${sample_name} \
−1 reads/${sample_name}.fastq.gz \
--midasdb_name uhgg --midasdb_dir midasdb_uhgg \
```

```
--species_list 100122,100277 \
--select_by median_marker_coverage,unique_fraction_covered \
--select_threshold=0,0.6 \
--num_cores 8 midas2_output
done
```

The number of CPUs used is specified via `--num_cores 8`.

For each sample, the above command performs CNV calling for two species of interest: `species_id=100122` (*Staphylococcus epidermidis*) and `species_id=100277` (*Streptococcus mutans*). The species specified by the users are still subject to the species selection: `median_marker_coverage>0` and `unique_fraction_covered>0.6`.

The output files are generated automatically under the directories `midas2_output/SRR172902/genes/` and `midas2_output/SRR172903/genes/`. This step can be parallelized over multiple samples (e.g., using shell background processes or `xargs`).

4. Confirm `run_genes` command has finished successfully. Once the single-sample CNV analysis is complete without any reported error, check for the output files:

   - `genes_summary.tsv`: An eight-column, tab-delimited file containing a summary of read alignment and CNV estimates for all species in the Bowtie2 index. Among all the reported columns, the average vertical coverage of all pan-genes covered by at least two reads (`mean_coverage`) is particularly useful because it determines the scope of the population CNV analysis.
   - `<species_id>.genes.tsv.lz4`: A seven-column, tab-delimited file of per-species CNV estimates for all species in the Bowtie2 index. Any pan-gene covered by more than two reads is reported. Among the columns, average vertical coverage (`mean_coverage`) and estimated copy number per cell (`copy_number`) are most useful for downstream analyses.

   It is important to note that MIDAS2 purposely holds off on any species selection or site filtering with the single-sample CNV results until across-samples CNV analyses are performed. Therefore, the number of reported per-species CNV results for each sample is the same as the number of species specified in `--species_list` or the number of species in `species_profile.tsv` if `--species_list` is not used. Both options are subject to the species selection filter (`--select_by and--select_threshold`).

5. Prepare sample manifest file for merging purposes. We can use the same `list_of_samples.tsv` generated by step 6 in Basic Protocol 1.

6. Upon the completion of `run_genes` for all the samples listed in the `list_of_samples.tsv`, MIDAS2 merges the CNV profiles across samples with the `merge_genes` command.

```
midas2 merge_genes --samples_list list_of_samples.tsv \ --
midasdb_name uhgg --midasdb_dir midasdb_uhgg \
--min_copy 0.5 \
--num_cores 2 midas2_output/merge
```

The number of CPUs used is specified via `--num_cores 2`. Pan-genes with copy number ≥0.5 are classified as present (`--min_copy 0.5`).

The output files are generated automatically under the directory: `midas2_output/merge/genes/`.

7. Optional: See additional parameters.

MIDAS2 has many additional, optional parameters not shown in the above example. These and the parameters shown above can be used to alter the behavior of the CNV module and to tailor the analysis to the microbial communities in the samples. Support Protocol 4 describes parameter choices, including parameters that control post-alignment filtering and species selection.

8. Population pangenome CNV analysis has finished successfully when all the following output files are created under the directory `midas2_output/merge/genes/` without any error message.

- `genes_summary.tsv`: Merged single-sample CNV summary containing information such as `mean_coverage`;
- `<species_id>/<species_id>.genes_copynum.tsv.lz4`: Gene-by-sample matrix of copy-number estimates;
- `<species_id>/<species_id>.genes_preabs q.tsv.lz4`: Gene-by-sample matrix of gene presence/absence;
- `<species_id>/<species_id>.genes_depth.tsv.lz4`: Gene-by-sample read coverage matrix;
- `<species_id>/<species_id>.genes_reads.tsv.lz4`: Gene-by-sample read counts matrix.

  More information on how to interpret the CNV results can be found in Guidelines for Understanding Results.

SUPPORT PROTOCOL 1: INSTALLING MIDAS2

MIDAS2 is written in Python 3 and can be executed on a 64-bit Linux system. MIDAS2 and its dependencies need to be pre-installed in order to run the commands described

in the basic protocols. We recommend the Conda package manager for installing these. Alternatively, users who want to ensure reproducibility by using a container with MIDAS2 and its computational environment may use Docker.

Necessary Resources

Hardware—64-bit Linux system with at least 16 GB of RAM

Software—Python (supported version is Python 3.7)

Miniconda; https://docs.conda.io/en/latest/miniconda.html

Or Docker, for users installing MIDAS2 in this way; https://www.docker.com/

Files—The latest MIDAS2 package; https://github.com/czbiohub/MIDAS2/

Installation using Conda package manager

1. We recommend installation using the Conda package manager as it encapsulates installation of the entire set of dependencies into a single command. If not already installed, execute the following command to install Miniconda:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-py37_4.12.0-Linux-x86_64.sh
bash Miniconda3-py37_4.12.0-Linux-x86_64.sh
```

2. Configure Conda channel.

```
conda config --set channel_priority flexible
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
conda config --add channels anaconda
```

3. Install MIDAS2:

```
conda install -c zhaoc1 midas2
pip install midas2
```

4. Optional. Alternatively, install MIDAS2 dependencies via a YAML file. This may be preferable if Conda takes a long time to resolve conflicts.

```
wget https://github.com/czbiohub/MIDAS2/releases/download/v1.0.2/midas2.yml
conda env create -n midas2 -f midas2.yml
```

```
conda activate midas2
pip install midas2
```

5. Verify your installation.

```
$midas2 --version
Metagenomic Intra-Species Diversity Analysis System 2 (MIDAS2),
Version 1.0.2
```

Installation using Docker

6. If Docker is properly installed on the system, users can also use the pre-built Docker container.

```
docker pull zhaoc1/MIDAS2:latest
```

SUPPORT PROTOCOL 2: COMMAND LINE INPUTS

MIDAS2 operates through a command-line interface (CLI). This interface enables reproducible analyses and allows MIDAS2 to be integrated into workflow management frameworks. The command-line allows basic specification of inputs, database, output, as well as advanced analyses.

Necessary Resources

Hardware—See Basic Protocol 1

Software—See Basic Protocol 1

Specifying output directory

MIDAS2 writes its outputs to a user-specified root directory, which is always passed as a mandatory argument to each of the six MIDAS2 analysis commands. For example, in this protocol `midas2_output` is the chosen output directory for single-sample analysis (SNV or CNV module) and `midas2_output/merge` is the chosen output directory for across-sample analysis (both modules). All analyses write to this output directory, with subsequent steps reading from it.

1. Single-sample commands:

   The three single-sample commands (`run_species`, `run_snps`, and `run_genes`) share three important command-line flags.

```
--sample_name <sample_name> Unique sample identifier;
−1 <file_name> Path to file containing forward reads;
−2 <file_name> Path to file containing reverse reads if applicable.
```

2. Across-sample commands:

   A tab-separated sample manifest file listing the `sample_name` and full path of the single-sample root output directory `midas2_output` is required for across-sample analyses.

```
--samples_list <file_name> Path to sample manifest file.
```

3. MIDAS reference database:

   For all MIDAS2 analyses, users need to specify the following parameters:

```
--midasdb_name <db_name> A valid precomputed MIDASDB name (e.g.,
uhgg or gtdb); --midasdb_dir <path_name > Local path for the
MIDASDB.
```

4. Other Parameters

```
--num_cores <cpu_counts> The number of physical cores to use;
-h All MIDAS2 commands print out a help message describing CLI
usage.
```

SUPPORT PROTOCOL 3: METAGENOTYPING WITH A CUSTOM COLLECTION OF GENOMES

This protocol describes how to build a MIDASDB from a custom collection of genomes and perform SNV metagenotyping with it. Other MIDAS2 commands can also be run with the new database. Single-sample SNV metagenotyping is shown as an example. There are three steps: Construct a custom rep-genome database for a collection of representative genomes of interest, build a Bowtie2 index, and execute `run_snps` command with the prebuilt Bowtie2 index.

Necessary Resources

Hardware—See Basic Protocol 1

Software—See Basic Protocol 1

Input Files—Collection of representative genomes (we have deposited example genomes organized in desired format on Zenodo: https://zenodo.org/record/6774976)

1. Download example genome collection folder from Zenodo to the work directory (`midas2_protocol`):

```
wget https://zenodo.org/record/6774976/files/midasdb_custom.zip
unzip midasdb_custom.zip
```

We have prepared two genomes for two species from the 21 National Center for Biotechnology Information (NCBI) genomes in the HMP mock community: *Staphylococcus aureus* (GCF_000013425.1) and *S. epidermidis* (GCF_006094375.1) for this protocol.

- `genomes.tsv`: The table-of-contents file specifying the assignment of representative genomes to species. We randomly assign each species a six-digit `species_id`.
- `cleaned_imports/`: The FASTA file of each representative genome, saved in the directory `<species>/<genome>/<genome>.fna`.
- `metadata.tsv`: Taxonomic assignment of the randomly assigned `species_id`.

2. MIDAS2 reserves the `--midasdb_name newdb` for building any new MIDASDB and the custom MIDASDB will be built at `--midasdb_dir midasdb_custom`.

3. Construct rep-genome component of MIDASDB.

Annotate all the genomes and build the files needed for the rep-genome database. These commands should be executed in the work directory `midas2_protocol`.

```
midas2 annotate_genome --species all \
--midasdb_name newdb --midasdb_dir midasdb_custom \
--debug --force
midas2 build_midasdb --generate_gene_feature \
--genomes all \
--midasdb_name newdb --midasdb_dir midasdb_custom \
--debug --force
```

There are two command-line parameters that users need to pass:

```
--debug: Keep the local file after successfully build the
database;
--force: Re-build the database even if one already exists locally.
```

4. Build one Bowtie2 index with the representative genomes.

```
midas2 build_bowtie2db \
--midasdb_name newdb --midasdb_dir midasdb_custom \
```

```
--species_list 100001,100002 \
--bt2_indexes_name repgenomes \
--bt2_indexes_dir bt2_index_custom \
--num_cores 8
```

We build the rep-genome Bowtie2 index for the two species specified via `--species_list` to the local directory `bt2_index_custom/`. Note we need to provide the custom MIDASDB `midasdb_custom/`to `---midasdb_dir`.

Users can also specify `--bt2_indexes_name` pangenomes to build the Bowtie2 index for pangenomes.

5. Execute `run_snps` with the rep-genome database:

```
for sample_name in SRR172902 SRR172903
do
midas2 run_snps \
--sample_name ${sample_name} \
−1 reads/${sample_name}.fastq.gz \
--midasdb_name newdb --midasdb_dir midasdb_custom \
--prebuilt_bowtie2_indexes bt2_index_custom/repgenomes \
--prebuilt_bowtie2_species bt2_index_custom/repgenomes.species \
--select_threshold=−1 \
--num_cores 8 midas2_output_custom
done
```

For each sample, this code performs single-sample SNV calling for all the species in the Bowtie2 database without any species filters (`--select_threshold=−1`). The number of CPUs used is specified via `--num_cores 8`.

The output directory is generated automatically under the directories: `midas2_output_ custom/SRR172902/snps` and `midas2_output_custom/SRR172903/snps`.

6. Confirm `midas2 run_snps` has finished successfully.

Once the single-sample SNV analysis is complete without any reported error, check for the output files (see Basic Protocol 3).

SUPPORT PROTOCOL 4: METAGENOTYPING WITH ADVANCED PARAMETERS

Basic Protocol 3 and Basic Protocol 4 demonstrate the core functionality of the MIDAS2 SNV and CNV modules using mostly default settings. This protocol describes all the available options for advanced SNV and CNV calling.

Necessary Resources

Hardware—See Basic Protocol 3

Software—See Basic Protocol 3

Input Files—See Basic Protocol 3

1. Single-sample post-alignment filter: Post-alignment filtering is an important component of the MIDAS2 metagenotyping approach, and filter thresholds can have a big impact on precision and recall (Zhao et al., 2022a). Users can adjust post-alignment filters via the following command-line options (default values indicated):

   - `--mapq 10`: Discard read alignments with alignment quality <10;
   - `--mapid 0.94`: Discard read alignments with alignment identity <0.94;
   - `--aln_readq 20`: Discard read alignments with mean quality <20;
   - `--aln_cov 75`: Discard read alignments with alignment coverage <0.75;
   - `--aln_baseq 30`: Discard bases with quality <30;
   - `--paired_only`: Only use properly aligned read pairs (both reads in a pair are retained if one of the reads passes the filter);
   - `--fragment_length 5000`: Maximum fragment length for paired-end alignments.

   For paired-ends read input, it is crucial to set a proper `--fragment_length` when `--paired_only` is specified. Incorrect fragment length affects the number of properly aligned read pairs. If the length is too short, few reads will be aligned but the alignment step runs faster.

2. Single-sample SNV calling: In recognition of the need for single-sample consensus allele calling, we provided the option `--advanced` to the `run_snps` command. This causes SNVs to be called using the pileup results for individual samples. SNVs are called for all the species in the rep-genome index. In the `--advanced` mode, per-species pileup results will report the within-sample major allele and minor allele for any genomic sites covered by at least two post-filtered reads. Custom filters can be applied to these outputs. Users are advised to use the setting `--ignore_ambiguous` to avoid falsely calling major/minor alleles for sites with tied read counts. The previously introduced post-alignment filter parameters can be used in `––advanced` mode:

```
for sample_name in SRR172902 SRR172903
do
```

```
midas2 run_snps \
--sample_name ${sample_name} \
−1 reads/${sample_name}.fastq.gz \
--midasdb_name newdb --midasdb_dir midasdb_custom \
--prebuilt_bowtie2_indexes bt2_index_custom/repgenomes \
--prebuilt_bowtie2_species bt2_index_custom/repgenomes.species \
--select_threshold=−1 \
--advanced --ignore_ambiguous \
--num_cores 8 midas2_output_custom_2
done
```

The output files are generated automatically under the directories: `midas2_output_custom_2/SRR172902/snps/`and `midas2_output_custom_2 /SRR172903/snps/`.

Once the single-sample SNV run is complete without any reported error, check for the output files (see Basic Protocol 3).

3. Population SNV filters: The population SNV analysis of MIDAS2 is by default only performed for species that are sufficiently well covered by aligned reads in sufficiently many samples.

   - `--genome_coverage 0.4`: Select species with horizontal coverage > 40%;
   - `--genome_depth 5`: Select species with vertical coverage >5×;
   - `--sample_counts 2`: Select species present in more than two relevant samples.

   For each genomic site for a given species, a sample is considered to be “relevant” if the corresponding site depth falls between the range defined by the input arguments `site_depth` and `site_ratio` × `genome_coverage`; otherwise, it is ignored for the across-samples SNV compute.

   - `--site_depth 5`: Minimal site depth >5;
   - `--site_ratio 3`: Maximum site depth less than three times the vertical genome depth;
   - `--snp_maf 0.05`: Minimal minor allele frequency to call an SNV is 0.05.

   For each species, a genomic site is considered to be “relevant” if the site prevalence across samples meets the range defined by the input arguments `snv_type` and `site_prev`. By default, common SNVs are reported.

   - `--site_prev 0.9`: Minimal proportion of samples in which site is present is 90%;

- `--site_type common`: Report common (alternatively rare) population SNVs;
- `--snp_type bi,tri,quad`: Report all possible SNVs or specify SNVs with two, three or four alleles only.

4. Chunk size: MIDAS2 subdivides the pileup work by splitting the species' genomes into smaller units called chunks. The chunk size is set so that each chunk takes a reasonable time to run (default `chunk_size` = `1000000`). In order to compute chunk-level population SNVs, all the pileup results of sites within the given chunk across all the samples need to be read into RAM. Therefore, at any given moment, a maximum number of pileup results, calculated from: total CPU cores × total number of sites per chunk × total number of relevant samples, will be read into RAM. Users can customize the chunk size according to their computing environment. There is a trade-off between chunk size, running time, and RAM usage. When metagenotyping hundreds or thousands of samples, MIDAS2 dynamically adjusts to a smaller chunk size (`--robust_chunk`).
   - `--chunk_size 1000000`: Number of sites in one chunk (smaller chunk size means less RAM needed but longer running time);
   - `--robust_chunk`: Dynamically adjust `chunk_size` based on species prevalence.

GUIDELINES FOR UNDERSTANDING RESULTS MIDAS2 Output

MIDAS2 stores all output files in <`midas_outdir`>, which is set by the users as the only mandatory positional argument to each of the MIDAS2 analysis commands. In this protocol, `midas2_output` is the chosen output directory, and all analysis steps operate within it. After running MIDAS2 on raw metagenomic sequencing reads from a set of samples, the output directory will contain (1) metagenotyping results for single samples that quantify the reads for each allele at each site of the genome of each genotyped species, and (2) merged results that quantify allele frequencies across the set of samples. Results will include SNVs and/or gene CNVs, depending on which modules were run. They will also include SCG-based estimates of the coverage of species across the samples, which MIDAS2 has used to determine which species are abundant and prevalent enough to metagenotype.

Single-sample results layout

For single-sample analysis, it is required that a unique `sample_name` per sample is provided. These names are used together with the output directory name <`midas_outdir`> to designate a unique output directory for each sample: <`midas_outdir`>`/`<`sample_name`>. MIDAS2 analysis usually starts with species prescreening which selects sufficiently abundant species in each sample (command `run_species`). After completing this step, users can run the SNV module with the `run_snps` command and/or the CNV module with the `run_genes` command. Single-sample results can be used to investigate genetic variation within samples (e.g., strain mixtures), and they can be the

input to downstream analyses that aim to deconvo-lute strains or investigate within-sample microbiome evolution.

Here is an example of output from a single-sample analysis in which the species, SNV, and CNV modules were run, as it would appear in the local filesystem:

```
Output                                Meaning
-------------------------------------------------------------------------
<midas_output>/<sample_name>
   |- species
      |- species_profile.tsv           Summary of species coverage
   |- snps
      |- snps_summary.tsv             Summary of read mapping to rep-genome
      |- <species>.snps.tsv.lz4        Per species pileups
   |- genes
      |- genes_summary.tsv             Summary of read mapping to pan-genome
      |- <species>.genes.tsv.lz4       Per species pan-gene coverage
```

Across-sample results layout

For a collection of samples, species coverage profiles are merged with `merge_species`, population SNVs are called using the commands `merge_snps`, and population CNVs are called using the command `merge_genes`. In this protocol, `midas2_output/merge` is the specified root directory for all the across-sample analyses. Output files specific to each species can be found in <`species`> subdirectories. Across-sample results can be used to perform association studies and evolutionary analyses, such as tracking transmission and ecological dynamics of strains across a set of samples from different locations or times.

After running all three commands, the output directory will be structured in the local filesystem as follows:

Output	Meaning
<midas_output>	
|- species	
|- species_prevalence.tsv	Profiling summary
|- species_marker_median_coverage.tsv	Species-by-sample matrix
|- species_unique_fraction_covered.tsv	Species-by-sample matrix
|- species_marker_coverage.tsv	Species-by-sample matrix
|- species_marker_read_counts.tsv	Species-by-sample matrix
|- species_relative_abundance.tsv	Species-by-sample matrix
|- snps	
|- snps_summary.tsv	Alignment summary
|- <species>/<species>.snps_info.tsv.lz4	Called SNVs information
|- <species>/<species>.snps_freqs.tsv.lz4	Site-by-sample matrix
|- <species>/<species>.snps_depth.tsv.lz4	Site-by-sample matrix
|-genes	
|- genes_summary.tsv	CNV summary
|- <species>/<species>.genes_copynum.tsv.lz4	Gene-by-sample matrix
|- <species>/<species>.genes_presabs.tsv.lz4	Gene-by-sample matrix
|- <species>/<species>.genes_depth.tsv.lz4	Gene-by-sample matrix
|- <species>/<species>.genes_reads.tsv.lz4	Gene-by-sample matrix

Understanding Population SNV Output—Given a collection of samples, MIDAS2 SNV module restricts population SNV calling to species that are “sufficiently well” covered in “sufficiently many” samples. The called population SNV results are found in `<midas_outdir>/snps/<species>`, and are organized by one subdirectory for each species (see Basic Protocol 3).

We start by describing the `<species>.snps_info.tsv.lz4` file: A seventeen-column, tab-separated, LZ4 compressed file listing details for all the SNVs. There are three tiers of information.

The first tier is basic information about the called population SNV allele (e.g., contig/scaffold, position). The first column is a unique identifier for each genomic site (`site_id`), composed of three parts: Reference identifier, reference position, and reference allele (`ref_id|ref_pos|ref_allele`). The second and third column report the population major allele (most abundant/prevalent in metagenomes) and the population minor allele (second most abundant/prevalent allele in metagenomes). The fourth column (`sample_counts`) reports the number of relevant samples that is eligible for population SNV calling for given species. By default, a given `<species,sample>` pair will only be kept if it has >40% horizontal genome coverage (`fraction_covered`) and five times vertical genome coverage (`mean_coverage`). The fifth column (`snp_type`) reports the number of alleles observed at the given site (e.g., fixed-site, bi-allelic site).

The second tier is information about the accumulated read counts and sample counts across all the relevant samples, which are used to call the population SNV. If users specify `--snp_method prevalence`, then counts of samples in which the site is present will be used to decide the major and minor allele. If users instead specify `--snp_method abundant`, then accumulated read counts will be used to decide the major and minor allele.

The third tier is gene-centric biological metadata about the population SNV, including the following:

`locus_type`: Denoting whether the allele is in a coding gene [coding sequence (CDS)], non-coding gene (RNA), or intergenic region (IGR). By default, MIDAS2 reports all the locus types and users can perform downstream filters, e.g., only population SNVs in coding genes.

`gene_id`: Gene identifier if locus type is CDS or RNA.

`site_type`: Degeneracy of given allele if CDS sites (i.e., one fold, two fold, three fold, or four fold).

`amino_acids`: Amino acids encoded by four possible alleles.

Next, we look at the `<species>.snps_depth.tsv.lz4` file, which is a site-by-sample read depth matrix, specifying the per-sample read counts of the population major and/or minor allele at each position. The rows of this table are indexed by the genomic site, matching the rows of the `snps_info` table. The columns are indexed by the species-specific relevant samples, i.e., only the samples with "sufficiently well" coverage for the given species. Note that not all the relevant samples are eligible for SNV calling for a given site. Specifically, for each genomic site, only samples with site depth falling in the user-defined range will be considered for population SNV calling. For ignored samples, the value in the `snps_depth` table is set to 0.

Last, we look at the `<species>.snps_freq.tsv.lz4` file, which is a site-by-sample allele frequency matrix, specifying the per-sample allele frequency of the population minor allele at each position. For example, `allele frequency = 1` in the `snps_freq` table means the corresponding sample has evidence for only one allele at the corresponding site and it is the same allele as the population minor allele. Frequencies between zero and one mean that there is evidence in the sample for more than one allele and the fraction represents the proportion of reads carrying the population minor allele. When `allele frequency = 0`, it means the sample has evidence for only one allele at that site and it is not the population minor allele. In most cases it is the population major allele but the sample could carry a strain with a third allele. Samples ineligible for the variant calling (see above) are encoded with −1.

We recommend users apply additional filters to the population SNV results. For example, we can select sites from CDS only or from only four-fold degenerate synonymous sites in proteins. We do this based on the `snp_info` table. We can further select samples with a stricter median site depth filter (e.g., 20×) based on the `snp_depth` table. The site-by-sample `snp_freq` table will be updated accordingly.

These outputs can be used for a variety of downstream population genetic analyses. For example, the site-by-sample `snp_freq` table may be used to compute between-samples distances (e.g., Manhattan distance). Principal coordinates analysis (PCoA) plots based on these pairwise distances can be used to visualize strain-level diversity (Zhao et al., 2022a). Alternatively, the `snp_freq` table can be provided as input to strain deconvolution tools,

such as StrainFacts (Smith, Li, Shi, Abate, & Pollard, 2022), which enable strain tracking, transmission, and investigations into strain-level ecological dynamics.

Understanding Population CNV Output—Given a collection of samples, MIDAS2 CNV module reports population CNVs for species with abundant pangenome coverage and high prevalence (both can be customized by users). All the genes covered by at least two reads will be reported. The per-species population CNV results are organized by species, with one subdirectory per species. The results of population CNV analysis are easy to interpret. For each species, there are three gene-by-sample matrices (see Basic Protocol 4). These can be used in downstream analyses to association gene presence/absence or copy number with traits of the microbes, microbial community, environment, or host.

More details regarding the interpretation of results and advanced parameters can be found online (https://midas2.readthedocs.io/en/latest/index.html). Examples of downstream analyses are given in the Commentary section.

COMMENTARY

Background Information

MIDAS2 is an extension of MIDAS (Nayfach, Rodriguez-Mueller, Garud, & Pollard, 2016), which was developed in 2016 to identify strain-level genetic variants in metagenomic data. MIDAS was created for the purpose of revealing the extensive population structure, functional variability, and strain dynamics that are overlooked when metagenomes are analyzed at the species taxonomic resolution.

With increasing popularity, it is critical to update metagenotyping pipelines like MIDAS to address computational efficiency and accuracy. Since 2016, the number of microbial genomes deposited in public collections has increased vastly, in particular with the addition of metagenome-assembled genomes from varied environments (Parks et al., 2017; Levin et al., 2021; Nayfach et al., 2021). Metagenotyping based on alignment to the huge number of available sequences poses a significant computational challenge, especially in environments with high species diversity (e.g., human gut microbiome) and in studies where a large number of deeply sequenced samples are analyzed together (e.g., metagenome-wide association studies; Power, Parkhill, & de Oliveira, 2017). Diverse reference databases with many closely related species can also reduce the precision and recall of read mapping (Bush et al., 2020), which can introduce bias into metagenotypes. MIDAS2 was developed to address these challenges. It is a faster and more scalable reengineering of the original MIDAS pipeline with new functionality for building custom databases that only include species present in the samples and tuning parameters that affect read mapping (e.g., paired-end filtering).

Despite sharing the same underlying structure, different metagenotyping tools vary in the availability/customization of reference databases, the implementation of post-alignment filters, and the contents of input and output files. For example, inStrain takes raw alignment files as input, implements post-alignment filters, and reports single-sample SNVs plus population SNVs between pairs of samples (Olm et al., 2021). On the other hand, metaSNV

v2 takes filtered alignment files as input, leaving the heavy lifting of post-alignment filtering to its users and it only reports across-samples non-reference variants (Van Rossum et al., 2021). MIDAS2 is a fully automated metagenotyping pipeline, which integrates generating and customizing the reference database to which reads are aligned, and includes read alignment, post-alignment filters, and variant calling. MIDAS2 outputs both single-sample SNV analysis and across-sample SNV/CNV analysis. MIDAS2 users can control each of these analysis steps and evaluate the effects of runtime parameter choices on the resulting SNV and CNV genotypes. As such, MIDAS2 enables reproducible, strain-level analyses.

A variety of population genetic analyses can be performed downstream of MIDAS2. These include studies focused on strain diversity, including statistical deconvolution of metagenotypes into strain genotypes and their relative abundances (Smith et al., 2022) and examination of evolutionary dynamics (Garud, Good, Hallatschek, & Pollard, 2019). In host-associated communities, metagenotypes enable the identification of strain transmission (Nayfach & Pollard, 2016; Brito et al., 2019; Chen & Garud, 2022), as well as strain-level associations with diet, disease, and medications (Roodgar et al., 2021; Bashiardes, Godneva, Elinav, & Segal, 2018). The protocols in this article empower users to generate the SNV and CNV data needed for such investigations from scratch, starting with metagenomic sequencing libraries.

Critical Parameters

Support Protocol 2 and Support Protocol 4 describe the critical parameters in depth. They also explain the impact that changing each parameter has on the results, compute time, and RAM usage. Another factor that will change the results is altering the reference database, as described in Support Protocol 3.

Troubleshooting

The most common issues are:

- Trouble installing MIDAS2 via Conda: When encountering an error message stating that the package is incompatible with your system, we recommend users install MIDAS2 via the provided YAML file (Support Protocol 2).
- Errors due to an invalid local MIDASDB path: Make sure to specify a valid database name via `--midasdb_name` and a valid local database path via `--midasdb_dir`. The name `--midasdb_name newdb` is reserved for custom built databases.

We also recommend browsing the frequently updated ReadTheDocs and Issues pages of the MIDAS2 Github repository at: https://github.com/czbiohub/MIDAS2.

Acknowledgments

This work was funded by the Chan Zuckerberg Biohub, Gladstone Institutes, National Heart, Lung, and Blood Institute (NHLBI) grant #HL160862, and National Science Foundation (NSF) grant #1563159.

Data Availability Statement

Data sharing not applicable: No new data generated.

Literature Cited

Almeida A, Nayfach S, Boland M, Strozzi F, Beracochea M, Shi ZJ, … Finn RD (2021). A unified catalog of 204,938 reference genomes from the human gut microbiome. Nature Biotechnology, 39(1), 105–114. doi: 10.1038/s41587-020-0603-3

Bashiardes S, Godneva A, Elinav E, & Segal E (2018). Towards utilization of the human genome and microbiome for personalized nutrition. Current Opinion in Biotechnology, 51, 57–63. doi: 10.1016/j.copbio.2017.11.013 [PubMed: 29223004]

Beghini F, McIver LJ, Blanco-Míguez A, Dubois L, Asnicar F, Maharjan S, … Segata N (2021). Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with bioBakery 3. bioRxiv. 2020:2020.11.19.388223 10.1101/2020.11.19.388223This article explains the use of marker genes for strain-level analysis.

Brito IL, Gurry T, Zhao S, Huang K, Young SK, Shea TP, … Alm EJ (2019). Transmission of human-associated microbiota along family and social networks. Nature Microbiology, 4(6), 964–971. doi: 10.1038/s41564-019-0409-6

Bush SJ, Foster D, Eyre DW, Clark EL, De Maio N, Shaw LP, … Walker AS (2020). Genomic diversity affects the accuracy of bacterial single-nucleotide polymorphism-calling pipelines. Gigascience, 9(2), giaa007. doi: 10.1093/gigascience/giaa007 [PubMed: 32025702] This article shows the importance of reference genomes to genotype accuracy.

Chen DW, & Garud NR (2022). Rapid evolution and strain turnover in the infant gut microbiome. Genome Research, 32(6), 1124–1136. doi: 10.1101/gr.276306.121 [PubMed: 35545448]

Clarke EL, Taylor LJ, Zhao C, Connell A, Lee JJ, Fett B, … Bittinger K (2019). Sunbeam: An extensible pipeline for analyzing metagenomic sequencing experiments. Microbiome, 7(1), 46. doi: 10.1186/s40168-019-0658-x [PubMed: 30902113]

Garud NR, Good BH, Hallatschek O, & Pollard KS (2019). Evolutionary dynamics of bacteria in the gut microbiome within and across hosts. PLoS Biology, 17(1), e3000102. doi: 10.1371/journal.pbio.3000102 [PubMed: 30673701]

Garud NR, & Pollard KS (2020). Population genetics in the human microbiome. Trend in Genetics, 36(1), 53–67. doi: 10.1016/j.tig.2019.10.010

Ghazi AR, Munch PC, Chen D, Jensen J, & Huttenhower C (2022). Strain identification and quantitative analysis in microbial communities. Journal of Molecular Biology, 434(15), 167582. doi: 10.1016/j.jmb.2022.167582 [PubMed: 35398320]

Langmead B, & Salzberg SL (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4), 357–359. doi: 10.1038/nmeth.1923 [PubMed: 22388286]

Levin D, Raab N, Pinto Y, Rothschild D, Zanir G, Godneva A, … Segal E (2021). Diversity and functional landscapes in the microbiota of animals in the wild. Science, 372(6539), eabb5352. doi: 10.1126/science.abb5352 [PubMed: 33766942]

Nayfach S, & Pollard KS (2016). Toward accurate and quantitative comparative metagenomics. Cell, 166(5), 1103–1116. doi: 10.1016/j.cell.2016.08.007 [PubMed: 27565341] This article describes the original development and implementation of MIDAS.

Nayfach S, Rodriguez-Mueller B, Garud N, & Pollard KS (2016). An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography. Genome Research, 26(11), 1612–1625. doi: 10.1101/gr.201863.115 [PubMed: 27803195] This article describes the original development and implementation of MIDAS.

Nayfach S, Roux S, Seshadri R, Udwary D, Varghese N, Schulz F, … Eloe-Fadrosh EA (2021). A genomic catalog of Earth's microbiomes. Nature Biotechnology, 39(4), 499–509. doi: 10.1038/s41587-020-0718-6

Olm MR, Crits-Christoph A, Bouma-Gregson K, Firek BA, Morowitz MJ, & Banfield JF (2021). inStrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains. Nature Biotechnology, 39(6), 727–736. doi: 10.1038/s41587-020-00797-0

Parks DH, Chuvochina M, Rinke C, Mussig AJ, Chaumeil P-A, & Hugenholtz P (2021). GTDB: An ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. Nucleic Acids Research, 50(D1), D785–D94. doi: 10.1093/nar/gkab776

Parks DH, Rinke C, Chuvochina M, Chaumeil PA, Woodcroft BJ, Evans PN, … Tyson GW (2017). Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. Nature Microbiology, 2(11), 1533–1542. doi: 10.1038/s41564-017-0012-7

Power RA, Parkhill J, & deOliveira T (2017). Microbial genome-wide association studies: Lessons from human GWAS. Nature Reviews Genetics, 18(1), 41–50. doi: 10.1038/nrg.2016.132

Roodgar M, Good BH, Garud NR, Martis S, Avula M, Zhou W, … Snyder MP (2021). Longitudinal linked-read sequencing reveals ecological and evolutionary responses of a human gut microbiome during antibiotic treatment. Genome Research, 31(8), 1433–1446. doi: 10.1101/gr.265058.120 [PubMed: 34301627]

Shoemaker WR, Chen D, & Garud NR (2022). Comparative population genetics in the human gut microbiome. Genome Biology and Evolution, 14(1), evab116. doi: 10.1093/gbe/evab116 [PubMed: 34028530]

Smith BJ, Li X, Shi ZJ, Abate A, & Pollard KS (2022). Scalable microbial strain inference in metagenomic data using StrainFacts. Frontiers in Bioinformatics, 2, 867386. doi: 10.3389/fbinf.2022.867386 [PubMed: 36304283]

Truong DT, Tett A, Pasolli E, Huttenhower C, & Segata N (2017). Microbial strain-level population structure and genetic diversity from metagenomes. Genome Research, 27(4), 626–638. doi: 10.1101/gr.216242.116 [PubMed: 28167665]

Van Rossum T, Ferretti P, Maistrenko OM, & Bork P (2020). Diversity within species: Interpreting strains in microbiomes. Nature Reviews Microbiology, 18(9), 491–506. doi: 10.1038/s41579-020-0368-1 [PubMed: 32499497]

Van Rossum T, Costea PI, Paoli L, Alves R, Thielemann R, Sunagawa S, & Bork P (2021). metaSNV v2: Detection of SNVs and subspecies in prokaryotic metagenomes. Bioinformatics, 38(4), 1162–1164. doi: 10.1093/bioinformatics/btab789

Zhao C, Dimitrov B, Goldman M, Nayfach S, & Pollard KS (2022a). MIDAS2: Metagenomic intra-species diversity analysis system. bioRxiv. 2022.06.16.496510 10.1101/2022.06.16.496510This article describes the MIDAS2 software tool used in the protocol.

Zhao C, Zhou JS, & Pollard KS (2022b). Pitfalls of genotyping microbial communities with rapidly growing genome collections. bioRxiv 2022.06.30.498336 10.1101/2022.06.30.498336This article presents a benchmarking study that quantifies the effects of closely related species and within-species diversity on alignment and metagenotype accuracy and it also reviews the literature on these topics.

Internet Resources

https://github.com/czbiohub/MIDAS2 Github repository for MIDAS2.

https://midas2.readthedocs.io/en/latest/ MIDAS2 manual.