Title

Identifying species-specific k-mers for fast and accurate metagenotyping with Maast and GT-Pro

Journal

STAR Protocols, 4(1)

ISSN

2666-1667

Authors

Shi, Zhou Jason
Nayfach, Stephen
Pollard, Katherine S

Publication Date

2023-03-01

DOI

10.1016/j.xpro.2022.101964

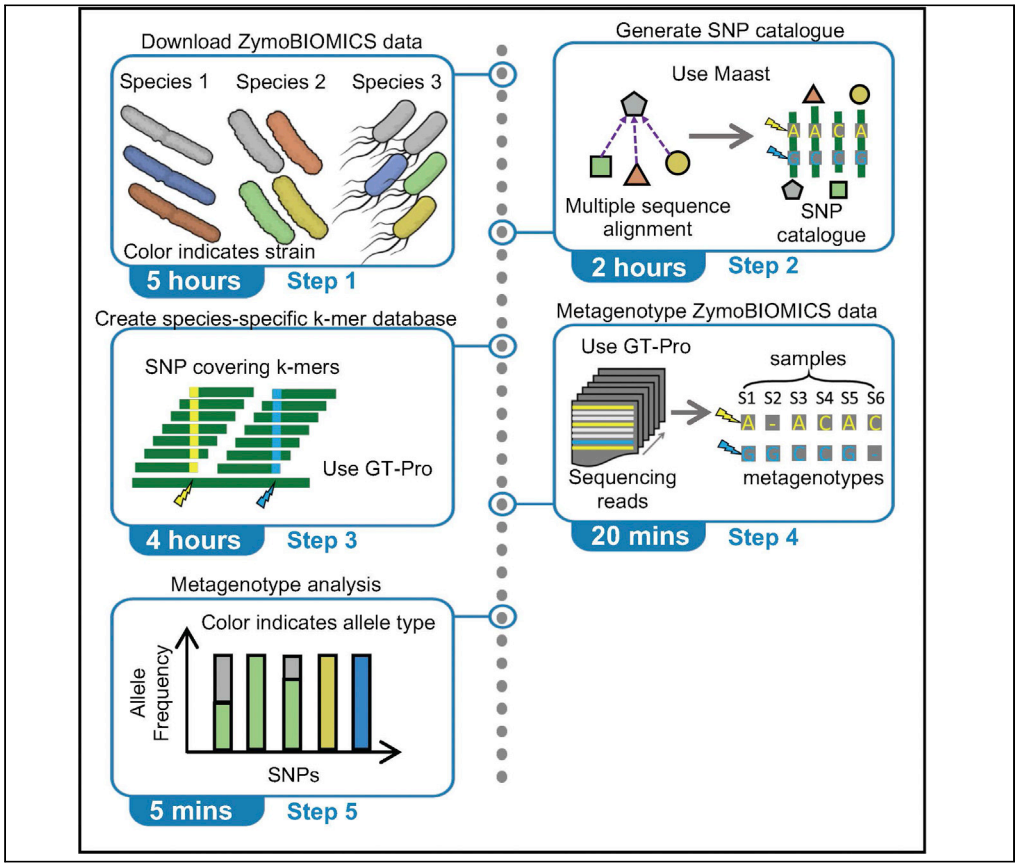
Peer reviewed

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Identifying species-specific k-mers for fast and accurate metagenotyping with Maast and GT-Pro



Genotyping single-nucleotide polymorphisms (SNPs) in microbiomes enables strain-level quantification. In this protocol, we describe a computational pipeline that performs fast and accurate SNP genotyping using metagenomic data. We first demonstrate how to use Maast to catalog SNPs from microbial genomes. Then we use GT-Pro to extract unique SNP-covering k-mers, optimize a data structure for storing these k-mers, and finally perform metagenotyping. For proof of concept, the protocol leverages public whole-genome sequences to metagenotype a synthetic community.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Zhou Jason Shi, Stephen Nayfach, Katherine S. Pollard

katherine.pollard@gladstone.ucsf.edu

Highlights

Computational pipeline for identifying SNPs in shotgun metagenomics sequencing data

Efficiently call SNPs in hundreds of genomes using Maast

Perform fast and accurate SNP genotyping on metagenomics data with GT-Pro

Shi et al., STAR Protocols 4, 101964
March 17, 2023
https://doi.org/10.1016/j.xpro.2022.101964

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Identifying species-specific k-mers for fast and accurate metagenotyping with Maast and GT-Pro

Zhou Jason Shi,[1,2,6,7] Stephen Nayfach,[3,4] and Katherine S. Pollard[1,2,5,8,*]

[1]Chan Zuckerberg Biohub, San Francisco, CA, USA

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

[3]Department of Energy, Joint Genome Institute, Walnut Creek, CA, USA

[4]Lawrence Berkeley National Laboratory, Environmental Genomics and Systems Biology Division, Berkeley, CA, USA

[5]University of California San Francisco, Department of Epidemiology and Biostatistics, San Francisco, CA, USA

[6]Present address: Department of Cancer Immunology, Genentech Inc., South San Francisco, CA 94080, USA

[7]Technical contact: zhoushihn@gmail.com

[8]Lead contact

*Correspondence: katherine.pollard@gladstone.ucsf.edu
https://doi.org/10.1016/j.xpro.2022.101964

SUMMARY

Genotyping single-nucleotide polymorphisms (SNPs) in microbiomes enables strain-level quantification. In this protocol, we describe a computational pipeline that performs fast and accurate SNP genotyping using metagenomic data. We first demonstrate how to use Maast to catalog SNPs from microbial genomes. Then we use GT-Pro to extract unique SNP-covering k-mers, optimize a data structure for storing these k-mers, and finally perform metagenotyping. For proof of concept, the protocol leverages public whole-genome sequences to metagenotype a synthetic community.

For complete details on the use and execution of this protocol, please refer to Shi et al. (2022a)[1] and Shi et al. (2022b).[2]

BEFORE YOU BEGIN

It is becoming common to analyze metagenomic data from a population genetic perspective.[3–5] For example, strain-level analysis of microbial communities can shed light on transmission, evolution, and phenotype associations. The first step in many of these analyses is to call genetic variants in shotgun metagenomics data, a process known as metagenotyping.[1,6–9] Downstream inferences depend upon accurate metagenotypes. It is also critical that metagenotyping pipelines are computationally efficient, because comparative population genetic studies often involve analyzing many samples. One approach to achieve both rapid and accurate metagenotyping is by performing exact matches between reads and SNP-covering k-mers (sck-mers).[1]

In this protocol, we demonstrate the specific steps involved in metagenotyping with sck-mers on published metagenomic sequencing data from a synthetic microbial community.[7] This ZymoBIOMICS Microbial Community consists of eight microbial species: *Bacillus subtilis, Enterococcus faecalis, Escherichia coli, Lactobacillus fermentum, Listeria monocytogenes, Pseudomonas aeruginosa, Salmonella enterica,* and *Staphylococcus aureus*. The steps described in this protocol are generalizable and can be adapted with minor modifications to metagenotype other microbial communities with any composition and diversity.

We illustrate how to metagenotype the ZymoBIOMICS community from scratch with five main steps (Graphical Abstract). First, all genomic and metagenomic data used in this protocol is downloaded

from a public repository. This includes a large collection of whole-genome sequences from the eight species plus three replicate DNA libraries of the ZymoBIOMICS Microbial Community sequenced ultra-deeply. Second, a catalog of common single nucleotide polymorphisms (SNPs) is generated for each of the eight species using Maast.[2] Maast is also used to dynamically identify tag genomes, a minimum set of phylogenetically diverse genomes that contains the maximum number of common SNPs, which enable efficient and accurate SNP calling. Third, species-specific sck-mers in these species are identified and used for building an optimized database with GT-Pro.[1] Fourth, exact match of sck-mers is performed using GT-Pro between the database and the reads from each of the three replicates of the ZymoBIOMICS Microbial Community. This generates metagenotypes for each species in each replicate. Finally, population SNVs (popSNVs)[7] are tabulated for each species between each pair of replicates, and these results are compared across pairs. All steps are accompanied with executable command lines.

Next, we describe the hardware and basic software requirements of this protocol, as well as the steps for installation of software dependencies.

> *Note:* Throughout this protocol, each line of executable code is preceded by a dollar sign ($). The protocol assumes the first line of the code is executed from the user's home directory (i.e., $HOME). To ensure full reproducibility, we recommend against directory changes during the first-time execution of this protocol unless the protocol instructs so. No root privilege or special administrator permission is required for any of the steps described in this protocol.

Hardware

Running through this protocol requires a computer with at least an 8-core processor, 32 GB of RAM, 512 GB of hard disk space, as well as a broadband Internet connection. We used an AWS instance (type: m5.2×large) for all steps in this protocol.

Software

This protocol at minimum requires a POSIX system (Unix, Linux, or macOS) with a version of Bash newer than 4.2.46. All steps described in this protocol were run on a Linux distribution Ubuntu 18.04.6 LTS with Bash version 4.4.20.

> *Note:* We prepared this protocol with the goal of minimizing upfront requirements. It is implemented with automatic installation of basic software tools such as the Python interpreter and C/C++ compiler. It is expected to work on a system where these tools are not already installed.

Vocabularies and abbreviations

In this protocol, sck-mers is the abbreviation for "SNP-covering k-mers".

Download and install software and tools

> Timing: 1 h

The following steps are to download and install software and tools for executing our pipeline.

1. Download and install conda.

```
$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh
$ bash ~/miniconda.sh -b -p
$ source $HOME/miniconda3/bin/activate
```

2. Create and activate a conda environment.

```
$ conda create -n metagenotyping_pl

$ conda activate metagenotyping_pl
```

3. Install compilers and basic tools.

```
$ conda install -c conda-forge -c gcc gxx make unzip
```

4. Install required software dependencies.

```
$ conda install -c zjshi -c conda-forge -c bioconda maast python=3.6.9 fasttree mash mummer4
pigz lz4 lbzip2 kmc ncbi-datasets-cli sra-tools
```

5. Download and install GT-Pro.

```
$ git clone https://github.com/zjshi/gt-pro.git

$ cd gt-pro

$ make

$ export PATH="$PATH:$PWD"
```

Note: To work properly, GT-Pro requires a C/C++ compiler that is compatible with C++ 11 standards. All the tests have been done and passed with clang-900.0.38 and GNU C/C++ Compiler (newer than 5.4.0). We have not tested GT-Pro with older versions of C/C++ compiler, but we expect it to run similarly as long as the compilation is successful.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
ZymoBIOMICS Microbial Community dataset	(Olm et al.[7])	https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA648136
Assembly Accession numbers for eight ZymoBIOMICS species	N/A	https://github.com/zjshi/zymo_accs
Software and algorithms		
GT-Pro (version>=1.0.1)	(Shi et al.[1])	https://github.com/zjshi/gt-pro
Maast (>=1.0.1)	(Shi et al.[2])	https://github.com/zjshi/Maast
NumPy (>=1.19.5)	(Harris et al.[10])	https://numpy.org/
Biopython (>=1.79)	(Cock et al.[11])	https://biopython.org/
Mash (>=2.3)	(Ondiv et al.[12])	https://github.com/marbl/Mash
MUMmer (>=4.0.0rc1)	(Marçais et al.[13])	https://github.com/mummer4/mummer
FastTreeMP (>=2.1)	(Price et al.[14])	http://www.microbesonline.org/fasttree/
KMC (>=3.2.1)	(Kokot et al.[15])	https://github.com/refresh-bio/KMC
NCBI sra-tools (>=2.11.0)	N/A	https://github.com/ncbi/sra-tools
NCBI datasets (>=13.35.0)	N/A	https://www.ncbi.nlm.nih.gov/datasets/docs/v1/download-and-install/

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
pigz (>=2.6)	N/A	https://zlib.net/pigz/
lbzip2 (>=2.5)	N/A	https://github.com/kjn/lbzip2
lz4 (>=1.9.3)	N/A	https://github.com/lz4/lz4

STEP-BY-STEP METHOD DETAILS

Step 1: Download ZymoBIOMICS microbial community data

Timing: 5 h (could vary depending on Internet bandwidth)

In this step, we download two types of genomic records that are related to the ZymoBIOMICS Microbial Community, including whole genome sequences of many strains of the eight species in this synthetic microbial community and shotgun metagenomic sequencing reads generated from three replicates of the community.

1. Download accession lists.

```
$ git clone https://github.com/zjshi/zymo_accs.git
```

Note: We prepared this Github repository for this protocol. This command will result in a directory named "zymo_accs", which contains a list (i.e., ZymoBIOMICS_species.list) of the names of the eight member species as well as eight lists of accession numbers for these species (e.g., Bacillus_subtilis__accessions.list). Each accession number can be used to access a whole genome sequence in the NCBI Assembly database. All the accession numbers were retrieved from NCBI assembly database as of Aug 3, 2022. Only assemblies in the highest tier of quality (i.e., complete level) are included.

2. Download whole genome sequences from NCBI assembly database.

```
$ cat ZymoBIOMICS_species.list | xargs -I[] bash -c "datasets download genome accession
-inputfile []__accessions.list -filename [].zip -exclude-genomic-cds -exclude-gff3
-exclude-protein -exclude-rna"
```

Note: This command will download eight zip files (e.g., Bacillus_subtilis.zip), each containing whole genome sequences of a species included in the accession list.

3. Download shotgun metagenomic sequencing reads from NCBI SRA.

```
$ mkdir fastq

$ fasterq-dump -O fastq/SRR12324251_sra SRR12324251 -f

$ fasterq-dump -O fastq/SRR12324252_sra SRR12324252 -f

$ fasterq-dump -O fastq/SRR12324253_sra SRR12324253 -f
```

Note: This command will create a directory named "fastq" and three subdirectories (e.g., SRR12324251_sra). Each subdirectory contains shotgun metagenomic sequencing reads of

one replicate of the ZymoBIOMICS Microbial Community. Sequencing reads will be downloaded in FASTQ format.

Step 2: Generate a catalog of common SNPs using Maast to compare genome assemblies

Timing: 2 h

4. Prepare Maast input files.

```
$ cat ZymoBIOMICS_species.list | xargs -I[] -n1 -P8 bash -c "unzip -o -d [] [].zip"

$ cat ZymoBIOMICS_species.list | xargs -I[] bash -c "mkdir []/genomes && ls []/ncbi_dataset/data/ | grep -v json | xargs -I{} -n1 -P8 bash -c 'cat []/ncbi_dataset/data/{}/*.fna > []/genomes/{}.fna'"
```

Note: These commands will prepare input files required by Maast for common SNP calling. Upon the successful execution, these commands will for each species create a working directory named by the species and extract all whole genome sequences in FASTA format into a subdirectory named "genomes" inside that working directory.

5. Call common SNPs with Maast.

```
$ cat ZymoBIOMICS_species.list | xargs -I[] -n1 -P1 bash -c "maast genomes -fna-dir []/genomes/ -out-dir []/maast_out/ -min-prev 0.9 -snp-freq 0.01 -threads 8"
```

Note: For each species, this command will generate an output directory named "maast_out" as well as the following files in the output directory: 1) a centroid genome, named 'reference.fna', which is picked as the reference genome for SNP calling, 2) a multiple sequence alignment (MSA) of tag genomes named 'tag_msa.fna', 3) a VCF file named 'core_snps.vcf' describing common SNP sites, and 4) a tab-separated values (tsv) file named 'coords.tsv' describing the coordinates of the core genome regions. In this command, a genomic site is called a SNP site if the site is present in >90% of the tag genomes and has two or more alleles with a minor allele frequency >1% in these genomes. Both centroid and tag genomes are automatically picked by Maast. SNP sites and core genome regions are described in the coordinates of the reference genome.

Step 3: Identify and create an optimized database of species-specific sck-mers with GT-Pro

Timing: 4 h

6. Prepare input files for GT-Pro database building.

```
$ cat ZymoBIOMICS_species.list | xargs -I[] -n1 -P8 bash -c "cp []/maast_out/*.* []/"

$ cat ZymoBIOMICS_species.list | xargs -I[] -n1 -P8 bash -c "mv []/tag_msa.fna []/msa.fa"

$ cat ZymoBIOMICS_species.list | xargs -I[] -n1 -P8 bash -c "echo $PWD/[]" > build.list
```

Note: This command will prepare all input files required by the database building module of GT-Pro.

7. Build a GT-Pro database.

```
$ GT_Pro build -in build.list -out db_build -dbname zymo8spec_db -threads 4 -overwrite
```

Note: This command will build a GT-Pro database named "zymo8spec.bin" which stores in binary format all species-specific sck-mers (for k = 31) as well as the unique identifiers of target SNPs of these sck-mers. It will also create a SNP dictionary file named "zymo8spec.snp_dict.tsv" that will be used for parsing the raw output of metagenotyping from GT-Pro. For the convenience of computing, the species in the database are encoded with eight six-digit integer IDs, ranging from 100001 to 100008. Complete mapping between the species and ID is described in an output file named "zymo8spec.species_map.tsv". This command is time consuming and takes 95% of the step 3 runtime (~3.8 h).

8. Optimize the GT-Pro database.

```
$ GT_Pro optimize -db db_build/zymo8spec_db -in $HOME/gt-pro/test/SRR413665_2.fastq.gz -tmp ./here
```

Note: This command will optimize the data structure for storing sck-mers into a GT-Pro database according to the specifications of the local computing environment. This can be used for rapid and RAM-efficient metagenotyping. This command requires an input file of metagenomic reads in FASTQ format for running a test. This file may contain a small number of reads for a quick test. A small test dataset downloaded with GT-Pro code repository is used here.

Step 4: Metagenotype the ZymoBIOMICS microbial community using GT-Pro

Timing: 20 min

9. Run GT-Pro to metagenotype the dataset.

```
$ GT_Pro genotype -d build/zymo8spec_db -o ./fastq/%{in} ./fastq/SRR12324251_sra/SRR12324251_1.fastq ./fastq/SRR12324252_sra/SRR12324252_1.fastq ./fastq/SRR12324253_sra/SRR12324253_1.fastq
```

Note: This command performs the metagenotyping on shotgun metagenomic sequencing data of the ZymoBIOMICS Microbial Community. For simplicity, this command only metagenotypes the forward reads (fastq 1). In practice, users would genotype forward and reverse reads, if both are available. This can be done by supplying both forward and reverse reads as either two individual input files or a single concatenated file. The command does not need to be modified otherwise.

10. Parse GT-Pro metagenotyping output.

```
$ echo -e "SRR12324251\nSRR12324252\nSRR12324253" > zymo_sample_sra.list

$ cat zymo_sample_sra.list | xargs -I[] -n1 -P3 bash -c "GT_Pro parse -in fastq/fastq_[]_sra_[]_1.tsv -dict ./build/zymo8spec_db.snp_dict.tsv -out fastq/[]_1__gtpro.tsv "
```

Note: This command will parse the GT-Pro output, which is encoded in a compact machine-readable format, into human-readable (tsv) format. The resulting output file has eight fields: 1) Species ID: six-digit integer which specifies a species, 2) Global Pos: an integer with up to seven digits which specifies the global position of a SNP in the representative genome of a species, 3) Contig: a string with arbitrary length which specifies the contig of the reference genome where a SNP is located, 4) Local Pos: an integer with up to seven digits which specifies the position of a SNP on a contig, 5) Allele 1: single character, A, C, G or T, which specifies allele 1 of a SNP, 6) Allele 2: similar as allele 1 but specifies allele 2 of a SNP, 7) Allele 1 Cnt: an integer specifying the count of metagenomic reads matching allele 1 and 8) Allele 2 Cnt: an integer specifying the count of reads matching allele 2.

Step 5: Compare identified genotypes between technical replicates

Timing: 5 min

11. Extract genotypes for each species stratified by technical replicate.

```
$ cat zymo_sample_sra.list | xargs -I[] -n1 -P3 bash -c 'sed "1d" fastq/[]_1__gtpro.tsv | awk "$0"' '{print $0 > "fastq/"$1"__[]_1__gtpro.tsv"}'
```

Note: This command generates a total of 24 (8 species × 3 replicates) output files, each containing genotypes of one species in a replicate.

12. Calculate the population average nucleotide identity (popANI) for each species across technical replicates.
    a. Prepare an input file.

```
$ join -j 9999 zymo_sample_sra.list zymo_sample_sra.list | awk '$1 < $2 {print $1, $2}' OFS='\t' | xargs -I[] bash -c "seq 100001 100008 | sed 's/$/\t[]/g'" | sort -k1,3 > paired_input.tsv
```

Note: This command will create a file in the format of tab-separated values, which contains all unique combinations of species and replicates. Each row of the file is a 3-tuple of species ID, replicate 1 and replicate 2.

    b. Copy and paste the text below with a text editor, and save it to a bash script named "count_popSNV.sh".

```
while IFS=$'\t' read -r SPEC SAMPL1 SAMPL2;

do

N_SITE=`comm -12 <(awk '$7 + $8 >=5 && ($7 == 0 || $8 == 0)' fastq/"$SPEC"__"$SAMPL1"_1__gtpro.tsv | awk '{print $1"_"$2}' | sort) <(awk '$7 + $8 >=5 && ($7 == 0 || $8 == 0)' fastq/"$SPEC"__"$SAMPL2"_1__gtpro.tsv | awk '{print $1"_"$2}' | sort) | wc -l`

N_MATCH=`comm -12 <(awk '$7 + $8 >=5 && ($7 == 0 || $8 == 0)' fastq/"$SPEC"__"$SAMPL1"_1__gtpro.tsv | awk '{if($7 > $8){print $1"_"$2"_"$5}else{print $1"_"$2"_"$6}}' | sort) <(awk '$7 + $8 >=5 && ($7 == 0 || $8 == 0)' fastq/"$SPEC"__"$SAMPL2"_1__gtpro.tsv | awk '{if($7 > $8){print $1"_"$2"_"$5}else{print $1"_"$2"_"$6}}' | sort) | wc -l`

echo $SPEC $SAMPL1 $SAMPL2 $N_SITE $N_MATCH $(expr $N_SITE - $N_MATCH)

done < ./paired_input.tsv
```

Note: We applied a filter to keep sites with 5× or higher coverage, similar to the previous study[7] for the purpose of comparison. Compared to alignment-based methods, GT-Pro tends to be more resistant to false positives due to sequencing errors or reads from one species mapping to another species.[1] Thus, GT-Pro enables accurate metagenotyping without coverage filters.

c. Run the bash script to count popSNVs.

```
$ bash count_popSNV.sh
```

Note: This command will create a file in the format of tab-separated values. Each row has six fields: 1) species ID: six-digit integer which specifies a species, 2) replicate 1: a string with a length of 11 which specifies the name of replicate 1, 3) replicate 2: similar as replicate 1 but specifying the name of replicate 2, 4) no. of sites: an integer specifying the number of SNP sites genotyped in both replicates, 5) no. of matches: an integer specifying the number of SNP sites genotyped with the same allele in both replicates, and 6) no. of popSNVs: an integer specifying the number of population SNVs which is calculated as number of sites - number of matches. Here the definition of a popSNV is similar to that in the previous study,[7] which is a site at which both replicates share no alleles, major or minor.

EXPECTED OUTCOMES

In this protocol, we described a computational pipeline which enables rapid and accurate identification of metagenotypes directly from shotgun metagenomic sequencing data. The pipeline is fully generalizable and easily customized for metagenotyping any bacterial, archaeal, and viral species as long as each species has at least 10 whole genome sequences. To demonstrate the pipeline, we designed an example application in which the pipeline was used to metagenotype three technical replicates of the ZymoBIOMICS Microbial Community. This analysis perfectly captured the sameness of strains between the replicates of the community.

We first downloaded from NCBI Assembly database 7,371 whole genome sequences that belong to the eight species in the ZymoBIOMICS Microbial Community. The number of genomes per species ranged from 36 to 2,988. The species also vary in their biology (e.g., pathogenicity) and genomic features (e.g., GC content). We generated a catalog of common SNPs (prevalence > 90% and minor allele frequency > 1%) for each species using Maast. The number of SNPs cataloged ranged from 88,389 to 885,074 across species, which account for more than 3 million SNPs in total. Among them, most are bi-allelic (range 72%–98%; Figure 1). We focused on leveraging these bi-allelic SNP sites for metagenotyping as they represent the majority of intra-specific genetic variation. We extracted >300 million candidate sck-mers and identified >200 million of these as unique, species-specific sck-mers using GT-Pro (Figure 2). More than 2 million SNPs have at least one species-specific sck-mer, and these sck-mers cover 50%–95.3% of total bi-allelic SNPs across species (Figure 2).

Next, we metagenotyped three replicates of the ZymoBIOMICS Microbial Community using GT-Pro and the sck-mer database we had built. These replicates contain ~233 million shotgun metagenomic sequencing reads (total across the eight species), which represents an average of > 300× coverage per replicate. We metagenotyped an average of 1.72 million SNP sites per replicate, which is highly consistent between replicates for all species (Figure 3). For each pair of replicates, we counted the number of popSNVs, and we then compared popSNVs across pairs. Since the strains are identical in the three replicates, zero popSNVs are expected to be found between any pair of replicates and any popSNVs reported are false positives. Our pipeline correctly detected zero

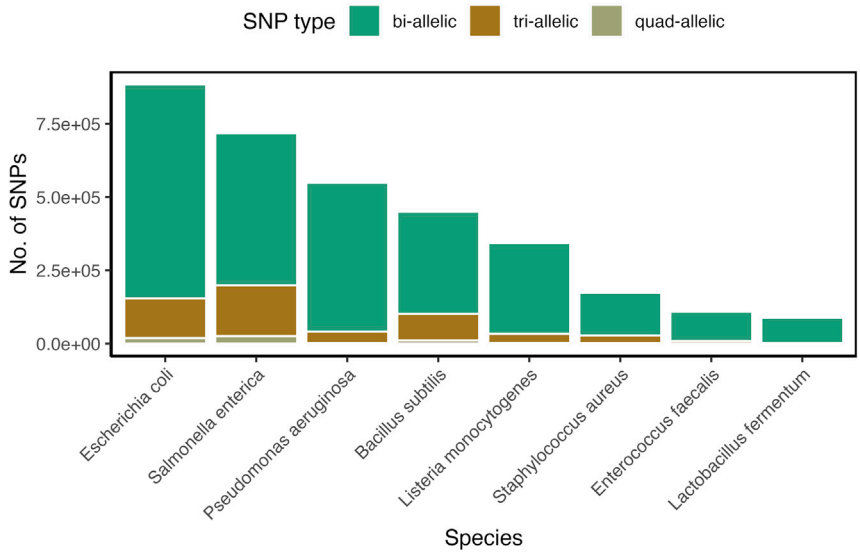


Figure 1. Calling common SNPs with Maast

The counts of bi-, tri- and quad-allelic common SNPs discovered in 7,371 genomes from eight species, summarized across species.

popSNVs for all species across all pairs of replicates (Figure 3), which is lower than the near-zero popSNVs reported by inStrain.[7] We note that the performance of inStrain is a result of using the exact reference genomes of original strains for read alignment, which precluded any negative impact of reference bias,[9] while our pipeline assumes no knowledge of exact reference genomes and uses public genomes. When the exact reference genomes are not available, which is generally the case for microbiome studies, the accuracy of inStrain as well as other alignment-based methods may significantly deteriorate due to the reference bias.[8] All these outcomes show that the pipeline described in this protocol is a powerful and general tool for rapid and accurate metagenotyping.

While our protocol is demonstrated using the ZymoBIOMICS Microbial Community, it can be easily adapted to metagenotype any set of metagenomic sequencing files, including those containing reads from diverse communities containing other microbial species. To customize the protocol, users do not need to change the commands, tools or parameters described in this protocol. Instead, users will only need to specify the input files, i.e., their fastq files plus a list of target species and multiple whole genome sequences for each species. We showed how to download whole genome sequences using command lines in this protocol and NCBI accession numbers, formatted as in the examples in our GitHub repository (https://github.com/zjshi/zymo_accs). For the species whose whole genome sequences are locally available, users may simply skip step 1 and perform SNP calling on these species using Maast as described in step 2. Our protocol can work with microbial species with varying numbers of whole genome sequences. To ensure a fair amount of intraspecific variation can captured, i.e., thousands of SNPs can be called, we recommend selecting species with whole genome sequences from at least 5 non-clonal strains. Metagenotyping with species-specific sckmers can maintain a low rate of false positives as sequencing depth goes extremely low (e.g., 0.001×). But we recommend 1× or higher for meaningful comparisons of metagenotypes between samples; at lower depths, very few SNPs overlap between samples. We also suggest using two checkpoints to ensure the validity of user-specific input files: (1) at task 5, step 2, by examining the "core_snps.vcf" file and verifying >1000 SNPs are called successfully, and (2) at task 7, step 3, by confirming two key database files, "zymo8spec.bin" and "zymo8spec.snp_dict.tsv ", are non-empty. With these minor changes, users can metagenotype their own community.

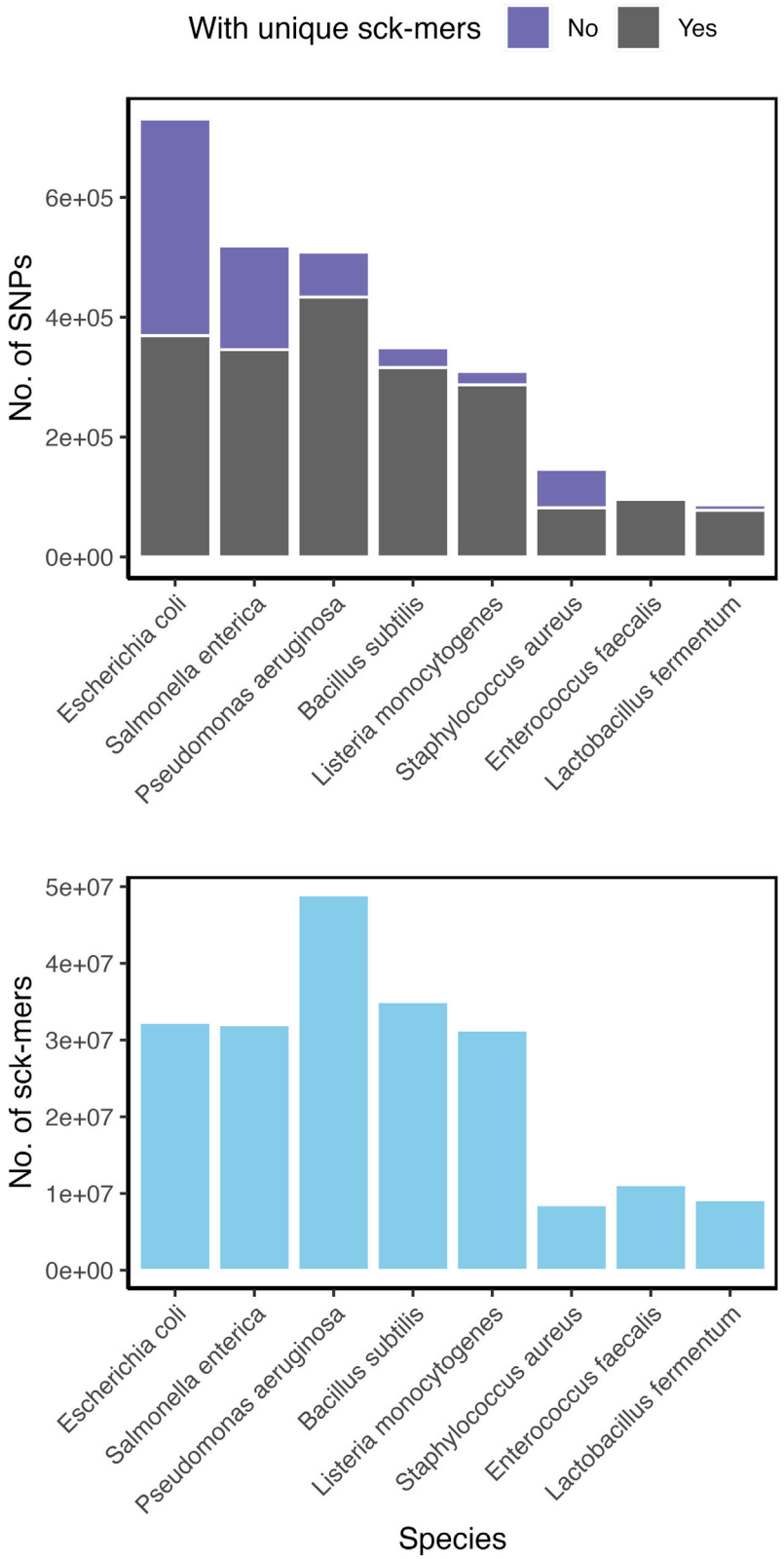


Figure 2. Building database of species-specific sck-mers with GT-Pro

(upper) The percentage of common SNPs with at least one species-specific sck-mer across eight species. (lower) The counts of species-specific sck-mers across species.

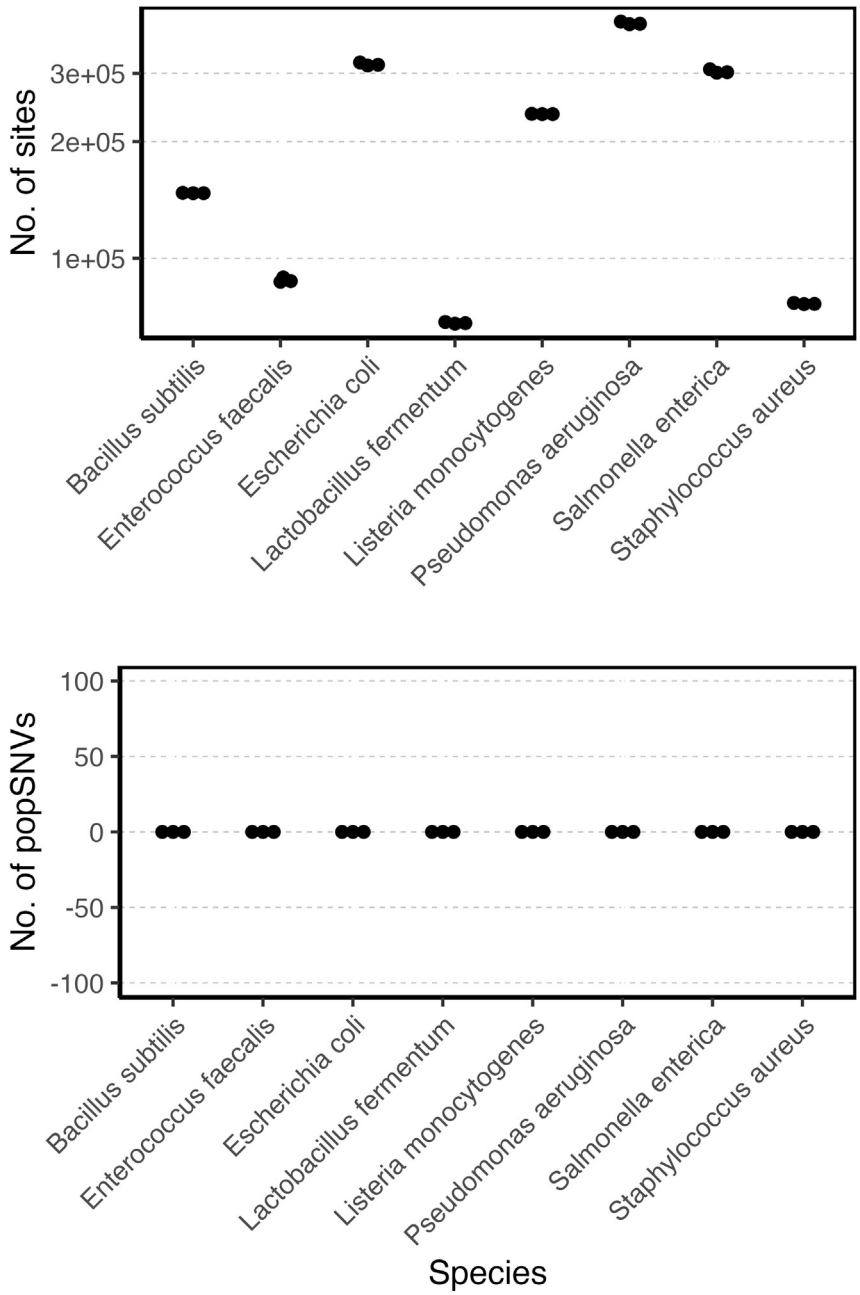


Figure 3. Comparison of genotypes between replicate samples

Each dot represents a pair of samples. The number of SNP sites genotyped (upper) and the number of population SNVs (popSNVs; lower) in both samples are summarized across species.

LIMITATIONS

Our protocol was tested on Ubuntu (tested on Ubuntu 18.04.6). For other UNIX -like system users, minor modifications of shell command lines and scripts may be needed due to different usage of basic tools, e.g., awk. For windows users, third-party software, such as a virtual machine or Cygwin, may be needed. Although GT-Pro has been shown to be able to detect novel strains and subspecific population structure, it is not designed for detecting novel SNPs, SNPs with three or four alleles, or SNPs without species-specific sck-mers. The current version of GT-Pro only achieved limited success

with microbial Eukaryotes with large genomes, though performance is expected to be acceptable for Eukaryotes with genome size smaller than 10 million bases.

TROUBLESHOOTING

Problem 1

Error "conda: command not found" while typing conda in command terminal. This error may arise at any stage of the protocol.

Potential solution

This is likely due to the base environment of conda not being activated. Activation may be achieved by typing in the following command.

```
$ source $HOME/miniconda3/bin/activate
```

Problem 2

Error "GT_Pro: command not found" while typing GT_Pro in command terminal. This error may arise at any stage of the protocol.

Potential solution

This error usually results from a failure to include a GT-Pro root directory in PATH, which defines a set of directories for the shell to search for executable files. Make sure the current directory is the root directory of GT_Pro, then type in the following command.

```
$ export PATH="$PATH:$PWD"
```

Problem 3

Error "XXX: command not found" where XXX is any of following commands: datasets, fasterq-dump, make, g++, maast, unzip and python. This error may arise at any stage of the protocol.

Potential solution

This error is usually due to inactivation of the conda environment where all dependencies are installed. This error may be solved by typing the following command.

```
$ conda activate metagenotyping_pl
```

Problem 4

At task 5, step 2, "core_snps.vcf" is found to be empty or only has a small number of SNPs after the successful running of Maast.

Potential solution

This problem usually is due to the intraspecific diversity of the input whole genome sequences being either too low or too high. To solve the problem, users may verify for each species that the input whole genome sequences are (1) conspecific, using annotations or by calculating average nucleotide identity (ANI) or an estimate thereof such as MASH distance, and (2) non-clonal (ANI or MASH distance < 99.9%).

Problem 5

At task 7, step 3, the database of species-specific sck-mers, "zymo8spec.bin", is found to be empty, or any species is found missing in the SNP dictionary file, "zymo8spec.snp_dict.tsv", after the successful execution of GT-Pro database building.

Potential solution

This problem often is due to highly similar or identical species being included in the input. It may also arise when the genomes a user supplies for different species are highly similar or identical. The problem can be solved by using annotations and/or ANI to spot highly similar genomes and then purging both redundant species and redundant genomes from the input files. Input species should be distinct from each other (ANI < 95%) and share no genomes.

RESOURCE AVAILABILITY

Lead contact

Katherine S. Pollard (katherine.pollard@gladstone.ucsf.edu).

Materials availability

This study did not generate new materials.

Data and code availability

This paper analyzes existing, publicly available tools and data. Details of these datasets are listed in the key resources table. This paper does not report original code.

ACKNOWLEDGMENTS

This work was funded by the Chan Zuckerberg Biohub, Gladstone Institutes, NSF grant #1563159, and NIH grant #HL160862.

AUTHOR CONTRIBUTIONS

Z.J.S. designed the protocol, conducted the analyses, and drafted the manuscript. K.S.P. and S.N. provided feedback and curated the results. K.S.P. supervised the study and edited the manuscript. All authors reviewed and approved the manuscript.

DECLARATION OF INTERESTS

K.S.P. is on the scientific advisory board of Phylagen. Z.J.S. is an employee of Genentech since October 24, 2022.

REFERENCES

1. Shi, Z.J., Dimitrov, B., Zhao, C., Nayfach, S., and Pollard, K.S. (2022a). Fast and accurate metagenotyping of the human gut microbiome with GT-Pro. Nat. Biotechnol. *40*, 507–516. https://doi.org/10.1038/s41587-021-01102-3.

2. Shi, Z.J., Nayfach, S., and Pollard, K.S. (2022b). Maast: genotyping thousands of microbial strains efficiently. Preprint at bioRxiv. https://doi.org/10.1101/2022.07.06.499075.

3. Garud, N.R., and Pollard, K.S. (2020). Population genetics in the human microbiome. Trends Genet. *36*, 53–67. https://doi.org/10.1016/j.tig.2019.10.010.

4. Ghazi, A.R., Münch, P.C., Chen, D., Jensen, J., and Huttenhower, C. (2022). Strain identification and quantitative analysis in microbial communities. J. Mol. Biol. *434*, 167582. https://doi.org/10.1016/j.jmb.2022.167582.

5. Shoemaker, W.R., Chen, D., and Garud, N.R. (2022). Comparative population genetics in the human gut microbiome. Genome Biol. Evol. *14*, evab116. https://doi.org/10.1093/gbe/evab116.

6. Nayfach, S., Rodriguez-Mueller, B., Garud, N., and Pollard, K.S. (2016). An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography. Genome Res. *26*, 1612–1625. https://doi.org/10.1101/gr.201863.115.

7. Olm, M.R., Crits-Christoph, A., Bouma-Gregson, K., Firek, B.A., Morowitz, M.J., and Banfield, J.F. (2021). inStrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains. Nat. Biotechnol. *39*, 727–736. https://doi.org/10.1038/s41587-020-00797-0.

8. Zhao, C., Dimitrov, B., Goldman, M., Nayfach, S., and Pollard, K.S. (2022). MIDAS2: metagenomic intra-species diversity analysis system. Preprint at bioRxiv. https://doi.org/10.1101/2022.06.16.496510.

9. Zhao, C., Shi, Z.J., and Pollard, K.S. (2022). Pitfalls of genotyping microbial communities with rapidly growing genome collections. Preprint at bioRxiv. https://doi.org/10.1101/2022.06.30.498336.

10. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. Nature *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

11. Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M.J.L. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics *25*, 1422–1423. https://doi.org/10.1093/bioinformatics/btp163.

12. Ondov, B.D., Treangen, T.J., Melsted, P., Mallonee, A.B., Bergman, N.H., Koren, S., and Phillippy, A.M. (2016). Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. *17*, 132. https://doi.org/10.1186/s13059-016-0997-x.

13. Marçais, G., Delcher, A.L., Phillippy, A.M., Coston, R., Salzberg, S.L., and Zimin, A. (2018). MUMmer4: a fast and versatile genome alignment system. PLoS Comput. Biol. *14*, e1005944. https://doi.org/10.1371/journal.pcbi.1005944.

14. Price, M.N., Dehal, P.S., and Arkin, A.P. (2010). FastTree 2 – approximately maximum-likelihood trees for large alignments. PLoS One *5*, e9490. https://doi.org/10.1371/journal.pone.0009490.

15. Kokot, M., Długosz, M., and Deorowicz, S. (2017). KMC 3: counting and manipulating k-mer statistics. Bioinformatics *33*, 2759–2761. https://doi.org/10.1093/bioinformatics/btx304.