metaSNV v2 User Manual
2021-11-23

If you use this software, please cite:
Thea Van Rossum, Paul I Costea, Lucas Paoli, Renato Alves, Roman Thielemann, Shinichi Sunagawa, Peer Bork. metaSNV v2: detection of SNVs and subspecies in prokaryotic metagenomes. Bioinformatics. 2021. https://doi.org/10.1093/bioinformatics/btab789

1 Installation and set up

See the instructions on the git repository https://github.com/metasnv-tool/metaSNV

To see all parameter options for each script described below, run:

python metaSNV.py --help
python metaSNV_Filtering.py --help
python metaSNV_DistDiv.py --help
Rscript metaSNV_subpopr.R --help

2 Method description

This section describes how the tool works. To see an explanation of the output files, see Section 3 below.

2.1 SNV calling

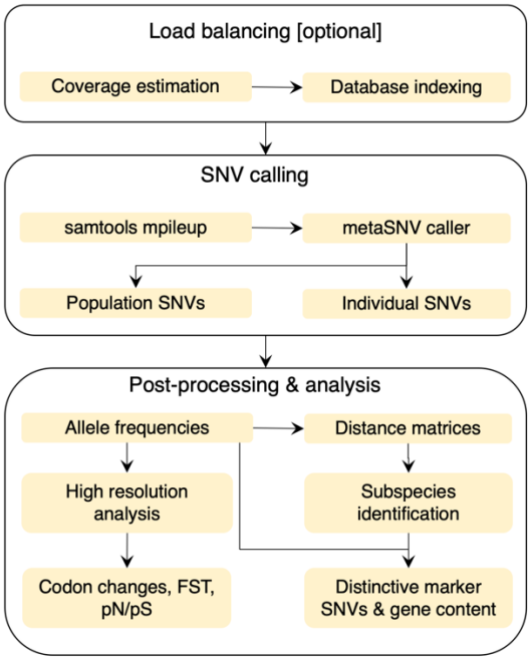

Figure 1 Diagram of metaSNV SNV calling, filtering, and post-processing

2.1.1 Input

As input, metaSNV v2 takes metagenomic short read sequences that have been mapped against a reference database. Metagenomic reads should be filtered for quality, with a minimum read length of 45 bp and a minimum base quality score of 20. If appropriate, reads should also be filtered to remove adapters and human data.

Mappings must have a minimum match length of 45bp and minimum percent identity of 97%. Each sample should be mapped against the database separately so that one BAM file per sample is produced. Only reads that map uniquely should be kept. BAM files should be sorted. Using 97% identity mapping aims to make sure reads are only mapped to a genome from their own species.

The reference database can be user-defined, but should contain one representative genome per species. Sequence names in the reference database should be formatted as: [primary species ID].[secondary species ID].[contig ID] without the brackets and without spaces, e.g. “525903.PRJNA29531.CP001818”. Having multiple contigs per species reference is supported. The “primary species ID” will be used in the results. Species and contig IDs should not contain periods or other special characters.

The reference database provided is from Progenomes2 (Mende 2019, https://doi.org/10.1093/nar/gkz1002). If there were multiple representative genomes in a Progenomes2 species cluster, then the longest genome was used. The reference database must include only one genome per species to detect subspecies.

Mappings can be run using any mapping software (BWA, bowtie, Minimap2, etc.), however, we recommend using ngless (https://doi.org/10.1186/s40168-019-0684-8) to facilitate mapping and filtering. Mapping can be done with BWA through ngless using something like the following (see the ngless documentation for details). This script will perform the mapping, only keep mappings with at least 97% identity and 45 bp length, and discard multi-mapped reads:

```
# ngless script to map metagenomic reads against a species reference database
# ARGV[1] and ARGV[2] are command line arguments passed to script, see below
# ARGV[1] is the directory where the input metagenomic fastq files are stored
# ARGV[2] is a file of file names, with one input file base name per line (such as “sample1” if there were two input files: sample1_R1.fq and
sample1_R2.fq)
ngless "1.1"
import "parallel" version "0.6"
import "mocat" version "0.0"
import "samtools" version "0.0"

samples = readlines(ARGV[2])
# grab one input sample to process
sample = lock1(samples)
# adjust naming scheme as required here:
input = paired(ARGV[1] + '/' + sample + '_R1.fq', ARGV[1] + '/' + sample + '_R2.fq')

fastaRefDb="/path/to/progenomes2_speciesReps_genomes.fna"

mapped = map(input, fafile= fastaRefDb, mode_all=True)
# only keep mappings with at least 97% identity and 45 bp length
mapped = select(mapped) using |mr|:
    mr = mr.filter(min_match_size=45, min_identity_pc=97, action={unmatch})

# only keep reads that mapped uniquely (discard multi-mappers)
mapped_unique = select(mapped, keep_if=[{mapped}, {unique}])
mapped_unique = samtools_sort(mapped_unique)
write(mapped_unique, ofile='outputs/' + sample + '.unique.sorted.bam')

collect(qcstats({fastq}), ofile='cdiffGenus_fqstats.txt', current=sample, allneeded=samples)
collect(qcstats({mapping}), ofile='cdiffGenus_mapstats.txt', current=sample, allneeded=samples)
```

Then this script, called “map.ngl” can be called using a command like below (with 8 threads). Every time this command is run, another file will be mapped. This can be easily parallelised using a scheduler array job.

```
ngless map.ngl -j 8 /path/to/my/fastq/file/dir/  /path/to/a/file/of/fastq/file/prefixes.txt
```

If you run a different mapping approach and have then filtered your mapping results to only include mappings with a minimum percent identity of 97% and a minimum alignment length of 45 bp, then you can remove any multi-mappers as follows:

Assuming all ambiguous alignments of a read are marked with quality = 0 (e.g. from bwa mem -a), then you'd process the resulting bam file with:

```
samtools view -buh -F 0x900 [-q 1] <bamfile> > <new_bamfile>
```

The -F 0x900 filters away all ambiguous alignments (0x100) that are flagged as secondary alignments as well as supplementary alignments (0x800), which are basically chimeric alignments. The -q 1 (alignment quality >= 1) would then discard the primary alignment of a multi-mapping read.

The -buh ensures that you have a properly formatted bamfile.

As an alternative to full length genomes, universal marker genes can be used as the reference for SNV calling. This results in less sensitivity but is faster. A custom database can be used, as described above.

2.1.2 SNV calling with `metaSNV.py`

SNV calling is run using the `metaSNV.py` script. This script takes as input a plain text file with a list of paths to alignment BAM files (one file per line) and the fasta reference database file that was used in mapping (described above). Optionally, this also takes a reference database annotation file that has information on gene coordinates, which enables codon changes to be calculated. Use the downloadable file for the Progenomes2 customised database. If you are using a custom database, see the script `src/gff2metaSNV_annotation.py` to convert a GFF file to the annotation format.

If you encounter an error, check whether your ".fai" file is in sync with your reference database fasta file. This file is usually created during execution of metaSNV and should be located in the same location as the fasta reference file. You can create this file yourself using the following command:

samtools faidx <ref.fasta>

2.1.2.1 Coverage calculation

Internally, metaSNV will then determine the average coverage over each reference genome in each sample. This will allow metaSNV v2 to prepare a balanced load between threads by grouping species into balanced sets based on their coverage. Coverage information is also used during the next step (filtering) to only keep samples that meet the horizontal and vertical coverage requirements per species.

Coverages are calculated using an embedded code version of https://github.com/CosteaPaul/qaTools with parameters "-c 10 -d" (maximum coverage to consider in histogram = 10; coverage histograms printed over each individual contig/chromosome in file <output>.detail)

2.1.2.1.1 Metagenomic SNV calling

metaSNV identifies single nucleotide variants (SNVs) on a per-nucleotide basis, building upon the mpileup tool in the samtools package. All reads from all samples that align to a given position are considered together. All SNVs must be observed in at least 4 reads (summing across all samples).

metaSNV differentiates SNVs into two classes of variants:

1. population SNVs (pSNVs): a non-reference nucleotide observed in at least 1% of reads across all samples
2. individual SNVs (iSNVs): a non-reference nucleotide that does not meet the population SNV criteria, but is observed in at least one sample with at least 4 reads

2.1.3 SNV filtering with `metaSNV_Filtering.py`

Further filtering can then be done on the SNV output to reduce false positives and focus on more prevalent SNVs. This is controlled by the following parameters passed to the `metaSNV_Filtering.py` script.

For each reference sequence (species), metaSNV first select samples that have a minimum of horizontal coverage and vertical coverage. This is controlled by parameters:

- [-b] Coverage breadth: minimal horizontal genome coverage percentage per sample per species (default: 40.0)
- [-d] Coverage depth: minimal average vertical genome coverage per sample per species (default: 5.0)
- [-m] Minimum number of samples remaining per species after filtering (default: 2, suggest 50 for subspecies calling)

After this sample-based sequencing, SNVs can also be filtered directly using these parameters:

- [-c] Minimum vertical coverage per sample at this SNV position per sample (default: 5)
- [-p] Minimum proportion of samples that have information at this SNV position (coverage is greater than zero) (default: 0.5). Denominator is the number of samples that met the previous filtering criteria for this species.

2.1.4 SNV post processing with `metaSNV_DistDiv.py`

Basic analysis of the species and samples can be done using `metaSNV_DistDiv.py`. Several things can be computed based on the filtered SNVs:

Dissimilarities between samples are computed per species based on their SNV frequency profiles as Manhattan distances ("mann") or Major Allele distances ("allele"). These distances are based on allele frequencies across all the pair-wise observed variants. The Manhattan distance adds the absolute frequency difference per site and normalizes to the total number of comparisons. That is, the number of sites for which the comparison was possible; if a position was not observed in a sample, it is ignored in the calculation. The "major allele" distance only considers differences in the major allele per site; that is, frequency differences greater than or equal to 60%. If a position has multiple variants, these are considered independently.

Nucleotide diversities and fixation indexes of samples are also calculated per species, per sample, within and between samples, specifically: nucleotide diversity (π), fixation indexes (FST), and synonymous (piS) and nonsynonymous (piN) diversity.

Equations from the original metaSNV publication (https://doi.org/10.1371/journal.pone.0182392):

Finally, nucleotide diversity (π) within and between samples and fixation indexes (FST) can be adapted to metagenomics data and computed for each species as previously described (Schloissnig *et al.*, 2013):

$$\pi(S_1, S_2, G) = \frac{1}{G}\sum_{i=1}^{n_{SNVs}} \sum_{N_1 \in \{ATGC\}} x_{S_1,i,N_1} \sum_{N_2 \in \{ATGC\}/N_1} x_{S_2,i,N_2}$$

$$F_{ST} = 1 - \frac{\pi_{within}}{\pi_{between}} = 1 - \frac{(\pi(S_1, S_1, G) + \pi(S_2, S_2, G))/2}{\pi(S_1, S_2, G)}$$

Where G is the size of the genome, and xS,i,N the frequency of nucleotide N, at position i in the genome, in sample S. All measured described above result in values from 0 to 1, with 1 denoting the greatest dissimilarity between two populations.

2.2 Subspecies detection module ("subpopr")

2.2.1 What is a subspecies?

Adapted from
Van Rossum, T., Ferretti, P., Maistrenko, O.M. et al. Diversity within species: interpreting strains in microbiomes. Nat Rev Microbiol 18, 491–506 (2020). https://doi.org/10.1038/s41579-020-0368-1 :

"Subspecies are groups of conspecific strains, and many definitions of the term exist[130]. In classic microbiology, subspecies are clusters of strains that are genetically or phenotypically distinct, have a type strain available[44] and are named (for example, Bacillus subtilis subsp. subtilis). Over time, the basis for classification of subspecies has shifted from qualitative phenotypic measures to genomic similarities between isolates[131]. This change has resulted in classification switches, such as the demotion of species to subspecies (for example, in Bifidobacterium longum[132]) and vice versa (for example, in Polynucleobacter necessarius[133]). Thus, named classical subspecies do not (yet) necessarily align with distinct genomic clusters. By contrast, in a population biology context, a subspecies is a set of local populations that live in a subdivision of a species' spatial range and that differ from other populations of the same species[75], for example, by genotype or phenotype[130]. Adapting the term 'subspecies' for microbiomics implies the same usage dichotomy as described for strains: classification of reads to an existing 'classic subspecies' and discovery of 'population subspecies' by clustering the within-species genetic variation observed across spatial scales"

130 Patten, M. A. Subspecies and the philosophy of science. Auk 132, 481–485 (2015).
44 International Committee on Systematics of Prokaryotes. International Code of Nomenclature of Prokaryotes: Prokaryotic Code (2008 Revision). Int. J. Syst. Evol. Microbiol. 69, S1–S111 (2019).
131 Meier-Kolthoff, J. P. et al. Complete genome sequence of DSM 30083(T), the type strain (U5/41(T)) of Escherichia coli, and a proposal for delineating subspecies in microbial taxonomy. Stand. Genomic Sci. 9, 2 (2014).
132 Fukuyama, M. et al. Unification of Bifidobacterium infantis and Bifidobacterium suis as Bifidobacterium longum. Int. J. Syst. Evol. Microbiol. 52, 1945–1951. (2002).
133 Hahn, M. W., Schmidt, J., Pitt, A., Taipale, S. J. & Lang, E. Reclassification of four Polynucleobacter necessarius strains as representatives of Polynucleobacter asymbioticus comb. nov., Polynucleobacter duraquae sp. nov., Polynucleobacter yangtzensis sp. nov. and Polynucleobacter sinensis sp. nov., and emended description of Polynucleobacter necessarius. Int. J. Syst. Evol. Microbiol. 66, 2883–2892 (2016).
75 Monroe, B. A modern concept of the subspecies. Auk 99, 608–609 (1982).

2.2.1.1 Am I detecting "subspecies" using metaSNV v2?

As the threshold between divergent conspecific subpopulations and subspecies is not well defined in microbiology, we provide the user with these tuneable parameters that reflect this spectrum. The default settings were selected to target "population subspecies" detection in human gut microbiome based on data from real population distributions (Costea et al 2017).

If you use the default parameters, you are likely detecting subspecies since metaSNV v2 was tuned and benchmarked against known population subspecies. (But see the previous section about classic vs population subspecies.) Thus, metaSNV v2 provides an operational definition of population subspecies from metagenomic data.

If you change the metaSNV v2 parameters or you don't use the mapping thresholds suggested above, then you may not be detecting subspecies. Instead, you might be detecting subpopulations defined in a more stringent or lenient way (i.e. with varying levels of diversity between and within them). These could be called within-species clusters or "phylotypes". These more general terms are recommended for use when it is unclear whether clusters detected by the subpopr module correspond to subspecies.

2.2.2 General methodology overview

First, samples are selected in which a species' population is dominated only by strains that could belong to a single subspecies. These samples contain minimal internal allele variation relative to the population SNV landscape (e.g. 80% of the population-wide SNVs have the same allele in over 80% of reads in a sample). These cut-offs are tuneable (see guidance below).

Second, these selected samples are then tested for clustered substructure using the dissimilarities between them with confidence in clustering results assessed using repeated subsampling and prediction strength(Tibshirani and Walther, 2005).

Third, distinctive genotyping SNV alleles are identified for each subspecies and used to estimate the relative abundance of subspecies in all samples. Filters for the abundance and prevalence of genotyping SNVs are applied. This profiling step can be performed on the original set of samples, which allows profiling of samples that did not meet SNV-discovery inclusion criteria (e.g. due to low species abundance) or on an additional set of new samples.

Fourth, subspecies-specific genes are detected by testing for correlations in abundance between genes and subspecies across samples. Gene abundance profiles are compiled by mapping reads against the relevant species' pangenomes(Maistrenko *et al.*, 2020) or a gene catalogue. The former approach is faster due to a smaller gene set, while the latter allows for novel species-gene associations to be discovered.

Finally, all results are summarised in plain text and html reports with embedded plots and statistical test results.

Comparison to StrainPhlAn

StrainPhlAn is an excellent tool that also measures similarity between populations based on SNV profiles, however, there are three major differences that set metaSNV v2 apart from StrainPhlAn and, we think, lead to better subspecies characterization.

1. MetaSNV v2 identifies within-species clusters (subspecies), while StrainPhlAn outputs only within-species trees/dissimilarities. StrainPhlAn yields dissimilarities between samples, but does not define clusters (subspecies). MetaSNV v2 also outputs dissimilarities but goes a step further to demarcate distinctive clusters. This demarcation is not trivial as there are many informatic decisions to be made during cluster detection. Additionally, MetaSNV v2 evaluates clustering results by calculating confidence scores for both the number of clusters and the cluster membership stability using rarefaction approaches.
2. MetaSNV v2 identifies SNVs in species' core genomes, while StrainPhlAn identifies SNVs only in clade-specific marker genes. This difference means that the resolution of MetaSNV v2 should be higher because it has more genetic material from which to detect SNVs. It also means that metaSNV v2 can use a user-defined reference genome set.
3. StrainPhlAn only reports the dominant strain, while metaSNV v2 reports multi-allelic SNVs. MetaSNV v2 only can identify subspecies if they are the dominant strain in multiple samples, but once identified, metaSNV v2 can profile multiple subspecies within each sample since it reports multi-allelic positions.

2.2.3 Data requirements

Two main requirements are intrinsic to the methodological approach of metaSNV v2. First, the subpopr module relies on the existence of many metagenomic samples where one subspecies is very dominant (e.g. by default: in at least 50 metagenomic samples, at least 80% of the dataset-wide species SNVs have the same allele in over 90% of reads in a metagenome). Second, the SNV calls are based on reference genomes. If the default suggested mapping parameters are used (minimum match of 97% identity), then metaSNV can only detect SNVs in sections of the genome that meet or surpass this threshold. This threshold is typical for within-species similarity. This means that for unusually divergent species complexes, such as Prevotella copri (Tett *et al.*, 2019), the full range of the sample population cannot be covered.

2.2.4 System and time requirements

Subpor can be run with 1 to 12 threads. More threads are possible but this tends to cause errors. Multithreading is enabled by R's BiocParallel package using the bplapply method.

Running 7524 samples on 71 species and correlating with 136k genes took 12 threads 6 CPU hours each (total 74 hrs) with maximum memory usage of 160 GB.
Running 1663 samples on 71 species with 12 threads took 37 CPU hours (total) with maximum memory usage of 86 GB. Without gene content calculations, this took less than 36 GB.

To require less memory, fewer threads can be used (i.e. fewer species will be analysed at once). Additionally, gene content and/or report generation can be skipped (to do this, use these parameters, respectively: --onlyDoSubspeciesDetection TRUE --createReports FALSE).

For human faecal microbiome samples, we suggest at least 100 samples with sufficient coverage per species for robust results.

2.2.5 Input

At minimum, subpopr takes the output from metaSNV.py, metaSNV_Filtering.py, and metaSNV_DistDiv.py as input. All three of these scripts must be run before metaSNV_subpopr.R can be run. Specifically, it uses the raw SNV files, the filtered SNV profiles, the Manhattan distance matrices, the bed_header file, the all_samples file, and the coverage files ("*.all_perc.tab" and "*.all_cov.tab").

For human faecal microbiome samples, we suggest at least 100 samples with sufficient coverage per species for robust results.

If you would also like gene content to be calculated, then gene abundance and species abundance profiles are required.

- **Species abundance profiles** are tsv files where rows are species and columns are samples. Column names must match file names used as metaSNV input (bam files). This file should be generated from mOTUs2 (https://motu-tool.org/index.html). These IDs will match the species reference genomes used in ProGenomes2. If you use a custom reference database for SNV calling, then the reference sequence IDs must match the species IDs in the abundance file and you must pass --isMotus=FALSE as a parameter.
- **Gene abundance profiles** are tsv files where rows are genes (or gene families) and columns are samples. The first column must be named (e.g. geneFamily) and contain gene family names. Subsequent column names must be sample IDs that match the BAM file names. Columns must sum to 1.

2.2.6 Sample selection for subpopulation discovery set (parameter selection)

Samples with relatively small similarities to all other samples are removed because they interfere with robust clustering. Samples are removed if their minimum dissimilarity to any other samples is more than 3 standard deviations away (+/-) from the mean of the minimum dissimilarity for all samples. If more than 3 samples fit this description, then a warning is printed in the log file and no samples are removed. If samples are removed, this is also tracked in the log file.

subpopr selects samples in which a species' population is dominated only by strains that could belong to a single subpopulation or subspecies. These samples contain minimal internal allele variation relative to the population SNV landscape (e.g. 80% of the population-wide SNVs observed in a sample have the same allele in over 90% of reads in that sample).

This is controlled with two parameters:

- [-x, --fixReadThreshold] Maximum abundance of minor alleles per locus for it to be considered "fixed". This filters SNVs. If the summed abundance of the minor alleles at this SNV position is more than this value, then the position is considered "unfixed". (hr). Default is 0.1 (i.e. at least 90% of reads have major allele). Range possible is 0.5 to 1.
- [-y, --fixSnvThreshold] Minimum proportion of fixed SNVs per sample. This filters samples. If the proportion of SNVs that are "fixed" in this sample (based on parameter fixReadThreshold) is greater than this number, then the sample is kept. (hs) Default is 0.8 (i.e. 80% of SNV loci have major allele with frequency > fixReadThreshold parameter).

As the threshold between divergent conspecific subpopulations and subspecies is not well defined in microbiology, we provide the user with these tuneable parameters that reflect this spectrum. The default settings were selected to target "subspecies" detection in human gut microbiome based on data from real population distributions (Costea et al 2017).

See also section 2.2.1

These filters will conservatively pick samples that have little within-species variation, which will form the basis of the clustering and subpopulation detection. Thus, this discovery subset does not include samples that might contain a somewhat even mixture of different subspecies.

If the x and y parameters are relaxed (raised and lowered, respectively) then this allows more strain variety within a sample. This will result in trying to define subpopulations based on samples that contain strains from multiple distinct subpopulations. This will make any clustering less distinct and will make it more difficult to identify genotyping SNVs for the resultant subpopulations. If the x and y parameters are too stringent, then too few samples and SNVs will pass filtering to give robust results. To guide parameter refinement, see the plots produced for each species in the `snvFreqPlots` output folder, which plot the effect of varying the x and y parameter.

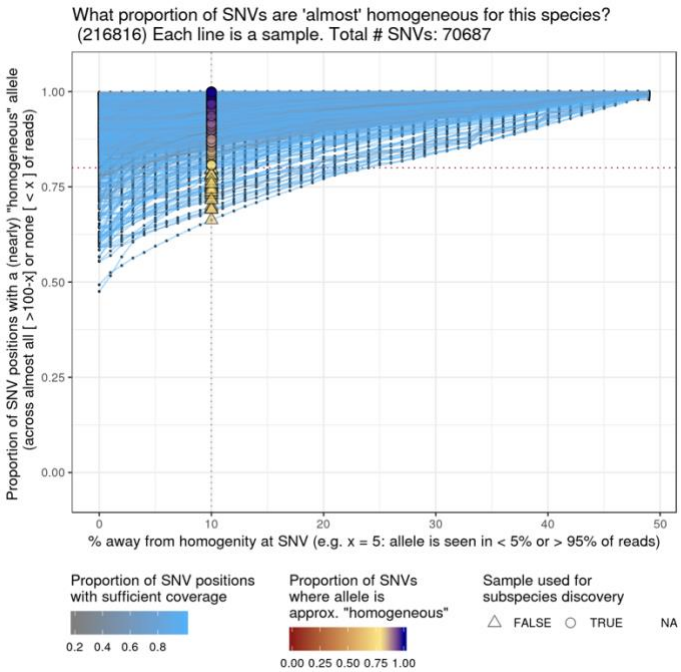

Figure 2 Amount of variability per sample across SNV positions for species 216816. Each line is one sample. Plots found in snvFreqPlots.

For human gut microbiome samples, there tends to be little strain diversity, thus the results are not so sensitive to these parameters. To identify subpopulations that are likely reflective of subspecies, the default parameters are suggested.

Since filtered, population SNVs are used here, only SNVs that are seen in at least 1% of reads across the whole sample set and that have coverage in >50% of samples are used. Additionally, from previous filtering, only samples that have the species present at rather high abundance (based on genome coverage) are used. Together, these filters allow to select samples per species that may have multiple strains but wherein these strains share a large proportion of SNVs that are also common in other samples, and therefore could reveal subpopulation structure in the species.

2.2.7 Clustering

The most appropriate number of clusters for each species is determined using the Prediction Strength algorithm from (Tibshirani and Walther, 2005) as implement in version 2.1-5 of the fpc R package (Hennig). A slightly modified version of this code is included directly in metaSNV v2, which allows the method to accept a distance matrix. From 1 up to 10 clusters are tried and a clustering Prediction Strength threshold of 0.8 is used. This threshold was recommended in the original publication but can be altered directly in metaSNV_subpopr.R using the CLUSTERING.PS.CUTOFF variable. For the Prediction Strength algorithm, the “pam” clustering method from the “cluster” R package is used, classifications are based on centroids, and 50 repetitions (parameter M) are performed. The mean Prediction Strength (PS) value across the 50 iterations is used to decide on the number of clusters.

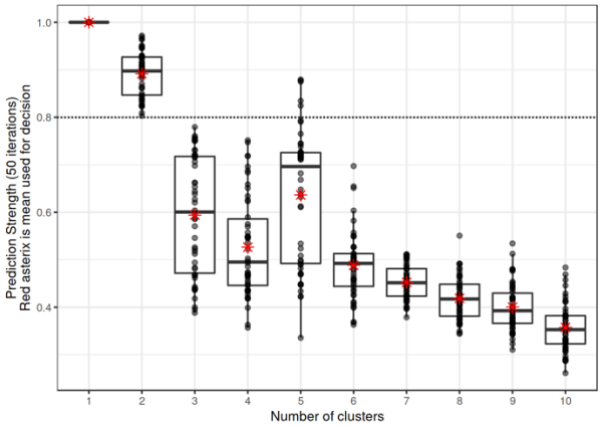

Figure 3 Prediction Strength values for species 216816

One the optimal number of clusters has been established, then the samples are assigned to that number of clusters using the “pam” clustering method from the “cluster” R package.

Clusters that have fewer than 3 samples are discarded.

Note on the fpc::prediction.strength implementation: the scoring of a cluster of size one is given 0 (lowest possible) here. In newer versions of fpc, it is given a score of 1 (highest possible). The old scoring is kept here as otherwise it results in very high cluster numbers for smaller sets of samples.

2.2.7.1 Clustering stability

To assess the stability of the clustering result and its sensitivity to the exact sample of the population, the clustering process described above is repeated 10 times for data subsampled to 30%, 40%, 50%, 60%, 70%, 80%, and 90% of the data. This results in the data illustrated below in Figures 4 and 5, for species 216816 and 155864 respectively.

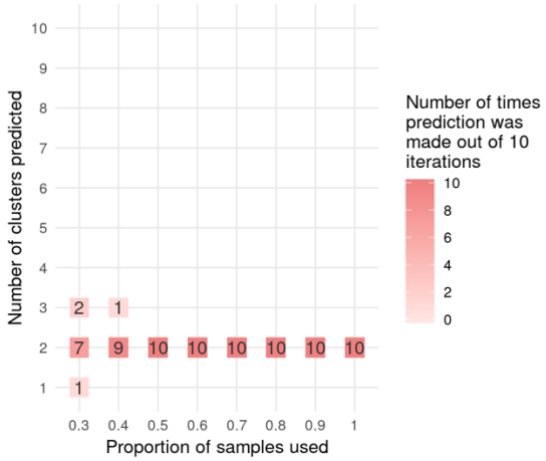

Figure 4 Clustering stability for species 216816 (high confidence)

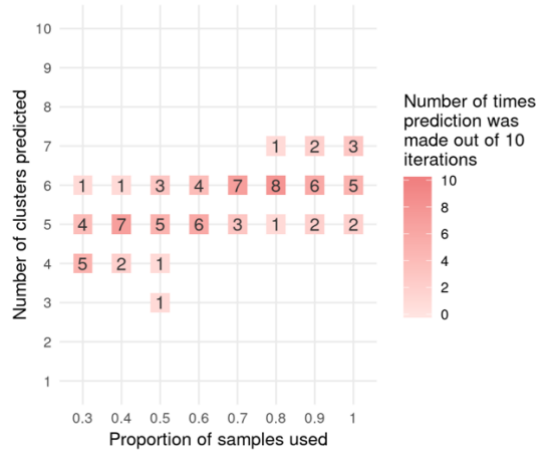

Figure 5 Clustering stability for species 155864 (low confidence)

Classification in the confidence in the number of clusters is derived based on these results.

- High confidence: same number of clusters predicted in every iteration when using 100% & 90% & 80% of data
- Medium confidence: same number of clusters predicted in every iteration when using 100% of data
- Low confidence: number of clusters predicted is unstable

The stability of the cluster membership is also assessed using the clusterboot method from the fpc package, using the "subset" bootstrapping method. This results in Jaccard subset mean data and the subset recovery index (subsetrecover/subsetbrd = [clusterwise number of times a cluster has been successfully recovered] / [clusterwise number of times a cluster has been dissolved]).

Cluster membership stability is also scored, per cluster:

- High: both stability measures > 0.9 when using >=70% of data
- Medium: both stability measures > 0.9 when using >=90% of data
- Low: does not meet criteria above

The results in these plots give a reading of whether the number of samples used in the study was high enough relative to the diversity within the (sub)populations. The minimal number of samples that are required will vary based on the populations and the species being considered.

2.2.8 Identifying genotyping SNVs

If subpopulation clusters were detected for a species, then SNVs are identified that are distinctive to the clusters. These SNVs are called "genotyping SNVs" or "gSNVs". These SNVs will be used to detect the presence and abundance of each subpopulation across all the samples in the study (in the raw SNV results).

The mean abundance of each SNV (allele frequency within a sample) across samples is compared between samples within a cluster versus those in another cluster (pairwise). The intersection of these pairwise comparisons is taken, so that each genotyping SNV is specifically distinctive of each cluster. A SNV is considered a genotyping SNV if an allele's mean abundance within the cluster is more than 80 percentage points higher than the mean abundance in all other clusters. This threshold can be altered using the `genotypingThreshold` parameter. Positions with insufficient depth of coverage were removed prior to calculation of means. A SNV is only considered as a possible genotyping SNV if it has sufficient coverage in at least 80% of samples within each cluster. Sufficient coverage is determined by the filtering parameters used in metaSNV_Filtering.py

The median abundance of the genotyping SNVs is then calculated for each sample in each cluster. If the relative abundance of the clusters within a sample sum to greater than 120% or to less than 80% then the sample is rejected as not being well classified. If more than 15% of samples are rejected then the genotyping is considered to have failed and genotyping SNVs are not produced for this species.

2.2.9 Profiling subspecies using genotyping SNVs

Genotyping SNVs are used to profile the abundance of subpopulations in all samples. For each genotyping SNV, the raw SNV calls for all samples are gathered and allele frequencies are calculated (number of reads with the allele divided by number of reads aligned at the position). Subpopulation abundances are then calculated as the median abundance of each cluster's genotyping SNVs. If a sample has less than 80% of the genotyping SNVs covered, then it is not used. If the relative abundance of the clusters within a sample sum to greater than 120% or to less than 80% then the sample is rejected as not being well classified. If a sample has an extreme mismatch between the median abundance of a cluster's genotyping SNVs (>30%) and the presence of its genotyping SNV positions (<80%), then it is rejected due to lack of confidence that the cluster is present. These samples are likely not well characterised by the discovery dataset.

Sums should ideally be 100 because the profiling should measure the relative abundance of all subspecies in the sample. The thresholds of 80 and 120 allow for some noise (though usually the sums are very close to 100). Samples discarded here are not discarded because they represent a mixture of subspecies, but rather because their compositions are not well represented by the subspecies genotyping data. This is rare, but when many samples are rejected in this way, the user is warned.

See section 2.3 for information on how to use genotyping SNVs from a previous metaSNV v2 run on new samples.

2.2.10 Gene content

Genes associated with clusters can be identified by looking for genes with abundances that correlate tightly with the cluster abundances. Here, cluster frequencies within species (e.g. 90%) need to be adjusted to be the abundance of the cluster across all species, i.e. relative to the whole community. To get these values, the within-species cluster abundances are multiplied by the respective relative species abundance. Once cluster and species abundances are gathered for all samples, then these are both tested for correlation with gene (or gene family) abundance profiles. Correlations are kept if Spearman correlation R is greater than 0.6 and if Pearson correlations using log10 transformed values have R greater than 0.8.

Gene abundance profiles can be compiled from any valid approach (e.g. custom mapping to gene catalogues or using functional profiling tools like HUMAnN or MEGAN). See section 2.2.5. See also the metaSNV v2 paper for an example, where gene abundance profiles were compiled per sample by mapping each sample against a set of species' pangenomes, selected to reflect the species with subspecies in this study. Briefly, metagenomic reads were mapped against species pangenomes (Maistrenko 2020, https://doi.org/10.1038/s41396-020-0600-z) using BWA and ngless (Coelho 2019, https://doi.org/10.1186/s40168-019-0684-8). Reads were kept if they mapped uniquely to a gene with a minimum match length of 45bp and minimum percent identity of 97%. Relationships between genes and COG categories were produced by eggNOG5 (Huerta-Cepas 2019, https://doi.org/10.1093/nar/gky1085) and obtained from the pangenomes, with species provenance information maintained. COG category abundances were calculated as sums of their member gene abundances.

2.3 Profiling clusters/subspecies in additional samples

The genotyping SNVs identified by subpopr can be used to estimate the relative abundance of each subspecies in any metagenome, including metagenomes from later independent studies, which were not included in the original subspecies identification analysis.

Within metaSNV_subpopr.R, the cluster abundance profiling described in section 2.2.9 will be run automatically on the original input files used. This section will describe how to run similar profiling on additional samples. This profiling can be run for one species at a time.

To profile new samples using genotyping SNVs identified from a previous metaSNV_subpopr.R analysis, a few steps must be taken:

1. The new samples must first be mapped against the genomes for the species of interest, using the same parameters as used in the original subpopr analysis that generated the genotyping SNVs. Here, the whole same reference database can be used, or a subset of the original database can be used. If using a subset, multiple species references must be included to enable competitive mapping between genomes, which effectively removes from consideration highly conserved regions shared across species. Thus, this subset should include multiple species references per genus.
2. SNVs in these bam files must then be detected using metaSNV.py. Filtering and distance calculations do not need to be run.

Once these bam files and SNV profiles have been created, the "profileSamplesUsingGenotypes.R" R script can then be used. This is located in the "src/" folder. This script takes as input:

- Path to directory containing metaSNV executables. Required. This is the directory where the "metaSNV_subpopr.R" script is.
- Path to directory containing results from metaSNV.py run (e.g. contains folder called snpCaller). Required. This is the directory that has the SNV results for the new samples that are to be profiled.
- Path to directory containing results from metaSNV_subpopr.py run. Required. This is the directory that has the results from the previous metaSNV v2 run that produced the genotyping SNVs (e.g. called "params.hr10.hs80.ps80.gs80/outputs" and containing files with pattern "*_hap_positions.tab" and "*_hap_freq_median.tab").
- ID of species to use (e.g. 155864). Required. This script can handle only one species at a time.
- Name of directory to create for results output [default= results/]

This script will copy the required results from the previous run of the subpopr module to the results directory. It will use these files in conjunction with the new metaSNV.py SNV calls. It will produce files as described in the output section below for the SNV allele read counts (*.pos), SNV allele frequencies (*.pos.freq), and stats on the profiling results (*_extended_clustering_*). The subspecies abundances are in the file with suffix: "_extended_clustering_wFreq.tab".

To demonstrate the functionality of metaSNV v2, we analysed 7,523 human faecal metagenomes from adults and infants from 27 countries for 70 prevalent and abundant gut species, of which 42 stratified into subspecies. The results of this analysis are described in Suppl. Info 1 of the paper. Supplementary tables and data from that analysis, including genotyping SNVs, are available on FigShare: https://doi.org/10.6084/m9.figshare.15131400 .

3 Output file descriptions

3.1 SNV calling

3.1.1 Operational files

Produced by: metaSNV.py

Location: [metaSNV_output_dir]/

- "all_samples": all samples used in this run of metaSNV. Order here is used elsewhere.
- "bed_header": a list with start and end positions for each sequence in the reference (format: sequence_id start end). Tab separated. One row per referenece sequence
- "bestsplits/": directory with files defining how the database is split up to parallelise processing. One file per split with its reference IDs.

Reference sequence coverage:

MetaSNV will filter samples, species, and SNVs based on of horizontal and vertical coverage of reference sequences (see parameters). These coverages are calculated using an embedded code version of https://github.com/CosteaPaul/qaTools with parameters "-c 10 -d" (maximum coverage to consider in histogram = 10; coverage histograms printed over each individual contig/chromosome in file <output>.detail)

Several files are produced:
- "cov/": directory with files summarising the coverage of each reference sequence for each sample.
- "[run name].all_cov.tab" : tab delimited file with mean coverage per sample. Two lines of headers, with first containing samples names.
- "[run name].all_perc.tab" : tab delimited file with proportion of reference sequence with at least 1x coverage per sample. Two lines of headers, with first line containing samples names.

3.1.2 SNVs raw read counts (unfiltered)

Produced by: metaSNV.py

Location: [metaSNV_output_dir]/snpCaller/

File name pattern:
[SNV_type].[split] (see below)

Format:

Tab delimited file with one row for each position in a reference sequence that had a non-reference nucleotide in the mapping results.

Columns are:

1. Reference sequence ID
2. Gene containing the position (if gene annotation file was given to metaSNV) or a "-" if no annotation was given or if the position is intergenic
3. Position
4. Reference allele
5. Mapping depth for this position in each sample (total read count at this position). Samples are separated by "|"
6. **"**Allele frequency strings" for each non-reference allele. Multiple non-reference allele frequency strings for one position are separated by commas

Each "allele frequency string" is a "|" delimited string that contains:

1. the summed frequency of the allele
2. the non-refence allele nucleotide
3. the codon change (reference then variant) and whether it is synonymous ("S") or non-synonymous ("N") (e.g. S[GCA-GCT]) (if gene annotation file was given to metaSNV). If no gene annotation, then a "."
4. The read count / mapping depth / frequency of this allele in each sample

e.g.: 134|G|N[TGT-GGT]|2|17|0|2|1|31|0|0|13|68

This file does not have a header. The read counts per sample are in the same order as the list of bam files provided when running metaSNV, which is also stored in the "[metaSNV_output_dir]/all_samples" file.

If the reference database was split to improve speed (with the "--n_splits" parameter) then there will be the corresponding number of files produced, each with the suffix ".best_split_N" where N is the split number.

metaSNV differentiates SNVs into two classes of variants:

1. population SNVs (pSNVs): a non-reference nucleotide observed in at least 1% of reads across all samples
2. individual SNVs (iSNVs): a non-reference nucleotide that does not meet the population SNV criteria, but is observed in at least one sample with at least 4 reads

Population SNVs are stored in files called e.g. "called_SNPs.best_split_0".
Individual SNVs are stored in files called e.g. "indiv_called.best_split_0".

Example:

called_SNPs.best_split_12

Reference sequence ID	Codon annotation	position	Reference allele	Mapping depth for this position in each sample	**"**Allele frequency strings" for each non-reference allele.
537011.PRJNA30025.GG703863	-	16111	T	0|10|0|232|0|235|151	38249|C|.|0|0|0|232|0|0|151,7128|G|.|0|10|…
537011.PRJNA30025.GG703862	-	46850	C		35735|T|.|0|13|

3.1.3 SNVs frequencies (filtered)

Produced by: metaSNV_Filtering.py

Location:

- [metaSNV_output_dir]/filtered/pop/
- [metaSNV_output_dir]/filtered/ind/

File name pattern:
[speciesID].filtered.freq

Format:

Tab delimited file of SNV frequencies (abundances) per sample, with one row for each variant allele that passed the user-defined filtering thresholds.

Header contains samples names, with leading tab.

Columns are:

1. Variant description string
2. Frequency of this variant in each sample. Frequencies range from 0-100 or -1 for no coverage. One column per sample.

Each “variant description string” is a “:” delimited string that contains:

1. Reference sequence ID
2. Gene annotation or a “-” if none
3. position in reference
4. variant allele (formatted as: reference>variant)
5. codon change or “.” if none

e.g.
657313.PRJNA39169.FP929055:RTO_R_32970:181:T>C:S[GAT-GAC]
657313.PRJNA39169.FP929055:-:124:A>G:.

Frequency values:

- 100 = 100% this allele
- 0 = 0% this allele
- -1 = insufficient coverage at this position (default min depth is 5)

MetaSNV differentiates two classes of variants, population SNVs and individual SNVs (see above). Each type is filtered separately and the results are written to the respective subdirectories (“pop” and “ind”).

3.1.4 Per-species distance matrices of samples

Produced by: metaSNV_DistDiv.py --dist

Location: [metaSNV_output_dir]/distances

File name pattern:
[speciesID].filtered.[distance measure].dist

Format:

Tab delimited symmetric matrix with header and 1st column containing the samples names. Numeric values are distances based on Manhattan distance (“mann”) or Major Allele distance (“allele”). These distances are based on allele frequencies across all the pair-wise observed variants. The Manhattan distance adds the absolute frequency difference per site and normalizes to the total number of comparisons. That is, the number of sites for which the comparison was possible; if a position was not observed in a sample, it is ignored in the calculation. The “major allele” distance only considers differences in the major allele per site; that is, frequency differences greater than or equal to 60%. If a position has multiple variants, these are considered independently.

3.1.5 Per-species nucleotide diversities and fixation indexes of samples

Produced by: metaSNV_DistDiv.py --div --divNS

Location: [metaSNV_output_dir]/distances

File name patterns:
[speciesID].diversity (from --div)
[speciesID].FST (from --div)
[speciesID].S_diversity (from --divNS)
[speciesID].N_diversity (from --divNS)

Format:
Tab delimited triangular matrices of values within and between samples for: nucleotide diversity (π), fixation indexes (FST), and synonymous (piS) and nonsynonymous (piN) diversity. For equations see the original publication https://doi.org/10.1371/journal.pone.0182392

3.2 Subspecies

All files in this section are produced by: metaSNV_subpopr.py

Results files are stored in a directory named using the values of key parameters used in running subpopr:
e.g. [subpopr_output_dir]/params.hr10.hs80.ps80.gs80/outputs/

3.2.1 Key output files

The main file to use which summarises all results and will allow you to navigate to more detailed results per species is **resultsSummary.html**

File name	Format	Contents
resultsSummary.html	HTML report	Summary of all clustering results for all species. This is the main file to use and will allow you to navigate to more detailed results per species.
summary_allResults.csv	csv with header	Text version of data displayed in resultsSummary.html
summary_clustering.csv	csv with header	Text summary of clustering specific results, content used in summary_allResults.csv
summary_clusteringExtension.csv	csv with header	Text summary of cluster extension specific results, content used in summary_allResults.csv
summary_geneFamilyCorrAssoc.csv	csv with header	Text summary of gene association specific results, content used in summary_allResults.csv

3.2.2 Logging

Three types of log file are created.

File name	Contains
log.txt	Parameters used in running the metaSNV_subpopr.R and tracking of progress and errors. Use "tail -f" to monitor progress during execution.
log_clusteringSummaryPerSpecies.txt	Number of clusters per species and/or errors
threadLogs/	Detailed logging for any errors that occur during analysis. One file per thread.

3.2.3 Detailed per-species output files

The files described below are found in:
[subpopr_output_dir]/params.hr10.hs80.ps80.gs80/

Directory	Contains
outputs/	Results files for species where clustering was detected
outputs/clustMedoidDefnFailed/	Results files for species where cluster medoid identification failed, likely this is due to too few samples
outputs/noClustering/	Results files for species where clustering was not detected
outputs/snvFreqPlots/	Plots that show the proportional abundance (frequency) of SNVs within samples for each species

3.2.3.1 Analysis summaries

File name	Format	Contents
[speciesID]_detailedSpeciesReport.html	html report	Summary of the gene content associated with this species
[speciesID]_geneContentReport.html	html report	Summary of the clustering data, procedure, and results for this species

In creation of these files, temporary directories are created. Their names contain time stamps (e.g. tmp_155864_2021-02-13-155853/). These should be deleted at end of execution.

3.2.3.2 Defining clustering

File name	Format	Contents
[speciesID]_freq_composition.tab	tsv with one row per sample and column headers that indicate cut offs, e.g. freq_data_sample_20_80 has the proportion of SNVs that had <20% frequncy or >80% frequency	Proportion of SNVs within each sample that have a very high or very low abundance
[speciesID]_mann_distMatrixUsedForClustMedoidDefns.txt	Tab delimited symmetric matrix with header and 1st column containing the samples names. Numeric values are Manhattan distances.	Distance matrix used for idnetifying cluster medoids. This is filtered to contain samples that passed filtering paramters.
[speciesID]_mann_distMatrixUsedForClustMedoidDefns_heatmap.png	Plot: columns and rows are selected samples. Cell colours are from low (yellow) to high (red) dissimilarity	Heatmap of the distance matrix used for idnetifying cluster medoids. Samples are clustered using average hierarchical clustering, which is not used in the identification of clusters. This is for illustrative purposes only

[speciesID]_mann_clusNumStability-heatmap.png	Plot: columns are rarefactions, rows are the number of clusters, numbers indicate the number of times each cluster number was detected	Rarefaction analysis results to assess robustness of clustering result to sample population. Sample set is subsampled to 90%, 80%, 70% etc and repeated 10 times for each subsample. Results are presented as a heatmap.
[speciesID]_mann_clusMembStability-recover.png	Plot	Rarefaction analysis results to assess stability of cluster membership
[speciesID]_mann_clusMembStability-jaccard.png	Plot	Rarefaction analysis results to assess stability of cluster membership
[speciesID]_mann_PS_values.tab	tsv with the number of clusters in cloumn 1 and the PS value in column 2	Cluster Prediction Strength values. See (Tibshirani and Walther, 2005) and metaSNV v2 publication for details
[speciesID]_mann_clusteringResult.rds	RDS	R object stored as RDS that contains all the (intermediate) clustering results for this species.
[speciesID]_mann_clustering.png	Plots: Upper PCoA: samples are coloured by cluster they were assigned to or left empty if they were not used in the cluster determination. Lower PCoA: samples are coloured by the proportion of SNVs that pass the 'homogeneity' threshold (parameter "--fixReadThreshold")	PCoA plot of samples based on distance matrix, coloured by cluster or by purity of SNV alleles
[speciesID]_mann_clustering.tab	tsv with sample names in column 1 and the cluster number to which each sample belongs in column 2	Cluster assignment per sample

3.2.3.3 Identification of cluster genotyping SNVs

File name	Format	Contents
[speciesID]_[clusterNumber].pos	Same as the metaSNV output SNV raw read counts (e.g. called_SNPs.best_split_12). See above.	Raw SNV data for positions that are genotyping SNVs for this cluster
[speciesID]_[clusterNumber].pos.freq	Same as the filtered metaSNV output, see above	Frequencies of SNV allele data for positions that are genotyping SNVs for this cluster
[speciesID]_[clusterNumber]_hap_positions.tab	tsv with 3 columns: row numbers, the “variant description string” (see above), whether the cluster has the reference allele	Records the allele (reference or variant) for each cluster. One file per cluster. When "flip" is TRUE, the cluster has the reference allele. (Flip means whether you should flip the abundance of the SNV in the pos.freq file.)
[speciesID]_hap_out.txt	text	Logging info on the genotyping of the cluster.
[speciesID]_hap_freq_mean.tab	tsv with 3 columns: sample name, frequency of the genotyping SNVs, cluster to which the genotyping SNVs belong	Mean of the frequency of the genotyping alleles per sample, per cluster
[speciesID]_hap_freq_median.tab	tsv with 3 columns: sample name, frequency of the genotyping SNVs, cluster to which the genotyping SNVs belong	Median of the frequency of the genotyping alleles per sample, per cluster

The [species]_[clusterID]_hap_positions.tab files tell you which allele it is (reference or variant) for each cluster

Flip means whether you should flip the abundance of the snp in the pos.freq file, i.e. whether the cluster has the reference allele for this position (flip = T -> reference allele)

posID look like this:
216816.PRJNA251950.CP008885:-:32:T>C:.

And code for:
[reference sequence ID]:[gene annotation or “-” if none]:[nucleotide position in genome]:[reference allele]>[variant allele]:[codon change]

Example for 2 clusters (files are identical except for flip):
Cluster 1

	posId	Flip
1	216816.PRJNA251950.CP008885:-:32:T>C:.	FALSE
2	216816.PRJNA251950.CP008885:-:35:G>A:.	FALSE

Cluster 2

	posId	flip
1	216816.PRJNA251950.CP008885:-:32:T>C:.	TRUE
2	216816.PRJNA251950.CP008885:-:35:G>A:.	TRUE

Cluster 1

	posID	
1	537011.PRJNA30025.GG703863:-:16111:T>C:.	TRUE
2	537011.PRJNA30025.GG703863:-:16175:T>A:.	TRUE

Cluster 2

	posID	
1	537011.PRJNA30025.GG703864:-:29181:C>T:.	FALSE
2	537011.PRJNA30025.GG703864:-:30873:G>A:.	FALSE

Cluster 3

	posID	
1	537011.PRJNA30025.GG703863:-:9223:G>A:.	FALSE
2	537011.PRJNA30025.GG703863:-:9394:T>C:.	FALSE

[speciesID]_[clusterNumber].pos files are tsv files that look like this (where each column is tab separated)

Genome ID		location	Reference allele	Read count for **position** in samples Each | separates a sample	Read count for **allele** in samples [sum]|[allele]|.|[sample1]|[sample2]|…
216816.PRJNA251950.CP008885	-	32	T	6|4|0|0|1|4|2|6|	34743|C|.|6|4|0|0|1|4|2|6

When you have 2 clusters, the 2 .pos files are identical

[species]_[clusterID].pos.freq

[reference sequence ID]:[gene annotation or "-" if none]:[nucleotide position in genome]:[variant allele]	Frequency in samples Each tab separates a sample			
216816.PRJNA251950.CP008885:-:32:C	100.0	-1	-1	0

100 = 100% this allele
0 = 0% this allele
-1 = insufficient coverage at this position

When you have 2 clusters, the 2 .pos.freq files are identical

3.2.3.4 Profiling of clusters based on genotyping SNVs (cluster "extension")

File name	Format	Contents
[speciesID]_extended_clustering_abundanceSummaryStats.tsv	tsv with headers; last two columns are the sample and cluster; prevalence is the proportion of SNVs present with at least 1 ("prevalence") or 5 ("prevalenceGte5") supporting reads.	Statistics for the abundance of the genotyping SNVs per sample
[speciesID]_extended_clustering_wFreq.tab	tsv with first column for sample name and subsequent columns for frequencies of each cluster (1 column per cluster)	Within-species frequency of each cluster in each sample based on the clusters' genotyping SNVs
[speciesID]_extended_clustering.tab	tsv with 2 columns: sample name and cluster classification	Classification of samples into clusters based on genotyping SNVs. If one cluster is clearly dominant in a sample (>80% abundance) then this is indicated here. If no cluster is clearly dominant (or present) then "NA" is written.
[speciesID]_extended_clustering_stat.txt	Plain text	Summary statement on cluster extension from genotyping SNVs
[speciesID]_clust_[clusterNumber]_hap_coverage_extended_normed.tab	tsv with 2 columns: the sample names and the relative abundance of the cluster (relative to sum of all species)	Relative abundance of one cluster normalised by species abundance (abundance relative to sum of all species)
[speciesID]_allClust_relativeAbund.tab	tsv with first column for sample name and subsequent columns for frequencies of each cluster (1 column per cluster)	Relative abundance of all clusters normalised by species abundance (abundance relative to sum of all species)

[species]_extended_clustering_wFreq.tab

Number = mean abundance of genotyping SNV positions that were seen in this sample
Positions with insufficient depth of coverage were removed prior to calculation
Samples can have at most 20% uncovered positions to still be considered in the extension

If you have neither of the two clusters present (e.g. a third strain type is present) then it might have a mix of the two genotypes (e.g. 40% of SNVs from cluster 1 and 60% from cluster 2). This is the case when the non-cluster strain is most similar to the reference genome

Samples are removed from this result if they have high abundance but low prevalence of genotyping SNVs. These samples are likely not well characterised by the original dataset. For example, if the subspecies has over 30% abundance, then 80% of its genotyping SNV positions should be covered. If this is not the case, then the sample is not included in the extension results.

[species]_extended_clustering.tab

If one cluster is clearly dominant in a sample (>80% abundance) then this is indicated here
If no cluster is clearly dominant (or present) then “NA” is written

3.2.3.5 Detection of cluster-associated gene content

File name	Format	Contents
[speciesID]_corrGenes-spearman.tsv	tsv with header	Results from Spearman correlation analysis between cluster abundance and gene abundance profiles
[speciesID]_corrGenes-pearson.tsv	tsv with header	Results from Pearson correlation analysis between cluster abundance and gene abundance profiles
[speciesID]_corrGenes-clusterSpecificGenes.tsv	tsv with header	Table of genes that are cluster-specific based on correlation analysis
[speciesID]_corrGenes-speciesSpecificGenes.tsv	tsv with header	Table of genes that are species-specific based on correlation analysis
[speciesID]_corrGenes-clusterSpecificGeneAbundances.tsv	tsv with header	Abundances of gene families and clusters for genes that cluster-specific. Abundances are relative to whole sample.