Forschungszentrum Borstel
Leibniz Lungenzentrum

MTBseq

Frequently Asked Questions

My installation from source failed!

Please make sure that you read the manual carefully. Most importantly, we only tested the installation on Ubuntu linux 16.04 LTS. Any other operating system will probably require some additional work especially to get the third-party software correctly installed and running, e.g. re-compilation of individual tools. A good option would be to install on a virtual machine running with Ubuntu linux 16.04 LTS.

General problem solving using the MTBseq log files.

For each run, MTBseq produces one main log file documenting the processing steps. In addition, each module will produce a specific log file that also contains the output of third-party software called in that module. These log files are an excellent first step to look into any issues, especially if the problem is with third-party tools.

Issues with the joint comparison of a collection of isolates.

There can be several seemingly confusing results when working with the joint comparison. Ideally, please read the paragraphs “How does the joint comparison in MTBseq work?” in order to understand what MTBseq actually does when comparing several samples as this will help you avoid almost all potential pitfalls. You may also want to read the paragraph “Why don’t we just combine variant tables for a joint comparison?” to understand the reasoning behind the workflow. In the following you will find a description of specific issues.

The distances between samples calculated from the set of phylogenetically informative positions are different to results from another pipeline:

MTBseq follows a distinct strategy for a joint comparison. In short, comprehensive information is gathered from the original mappings for any position with a variant detected in any of the samples in the comparison. As explained in the paragraph “Why don’t we just combine variant tables for a joint comparison?” this will prevent errors made when simply joining together variant tables.

The filtered set of SNPs for phylogenetic analysis contain little to no positions:

Please make sure that all samples in the comparison have sufficient sequence data. We recommended at least 30-fold mean coverage depth and 95% coverage breadth (you can get these numbers from the dataset overview file created by MTBseq). The reason being that for phylogenetic analysis, the full list of variant positions is also filtered for quality. Only positions that have either WT base or a SNP in all! of the isolates are retained. This means that the position must be covered by reads in all isolates, and cannot be deleted in any of the isolates. In addition, we demand that a certain percentage of all isolates fulfill our quality thresholds at this position. By default, this is set to 95%, meaning that the quality thresholds for variant detection must be met by at least 95% of the isolates compared at a specific position. Otherwise, this specific position is removed from the analysis. Taken together, this means that even a single dataset with bad sequence data (such as very low coverage depth) can seriously limit the regions of the genome taken into account.

The joint comparison table reports a variant not contained in the variant table of that respective sample:

When preparing the joint comparison report, MTBseq drafts a table containing every position for which a variant has been detected in any of the samples in the comparison. That table is then complemented with the sequence information from the respective mapping files. Therefore, it can happen that the table contains variant positions in a sample dataset that did not pass the threshold criteria for variant calling.

Joint comparison fails without producing result files:

In the current implementation, the extensive calculations done by MTBseq for a joint comparison require a lot of RAM. If your joint analysis fails, please check whether the process is being stopped due to RAM shortage, as there is currently no warning given by MTBseq if that happens.

Amend comparison table fails without producing result files:

When instructing MTBseq to perform a joint comparison, you always need to provide MTBseq with a list of samples. That is also true, if you already have a joint table file ready and want to restart with the second step in the joint analysis (Amend). If you do not provide the (correct) file, MTBseq will not be able to deduce the correct file name for the joint table file, and give the “No joint variant file” error.

How does the joint comparison in MTBseq work?

The MTBseq analysis pipeline has two distinct branches, a primary sample-specific workflow and a secondary joint analysis workflow. Every dataset has to be analyzed with the sample specific workflow first. Here, reads will be mapped against a reference genome and variants called. After the primary analysis for one sample has been done, alignments and variant tables have been created for that sample, and from then on that sample can be included in any secondary joint analysis. During the primary analysis, a set of parameters are used to define a valid variant call, this includes among others the number of reads indicating the allele mapped in forward or reverse orientation. If a variant position does not fulfill the criteria for a valid call, it will not be included in the variant list of that specific dataset.

For the joint analysis, MTBseq first combines all positions for which variants were reported in any of the datasets included in the comparison. Then, MTBseq retrieves the sequence data for each of these positions from the read alignments (or more precisely, from the position lists encoding that information). MTBseq has thus created a table of positions with comprehensive sequence information for every position for which a variant was called in any of the datasets in the joint analysis (= the Amend files). It can happen that this table includes a position that has a valid variant call in one dataset, but sequence quality at this position in another dataset was not sufficient to make a variant call (even though available information indicates the variant). As MTBseq retrieved the comprehensive sequence data, we can actually consider this information in the analysis instead of suspecting wild type sequence. MTBseq will therefore use the consensus base for that position in that dataset when calculating pairwise distances and creating the fasta files of aligned SNP positions for phylogenetic analysis.

For phylogenetic analysis, the full list of positions are then filtered. Only positions that have either WT base or a SNP in all! of the isolates are retained. This also means that the position must be covered by reads in all isolates, and cannot be deleted in any of the isolates. In addition, we demand that a certain percentage of all isolates fulfill our quality thresholds at this position. By default, this is set to 95%, meaning that our quality thresholds for variant detection must be met by at least 95% of the isolates compared at a specific position. Otherwise, this specific position is removed from the analysis.

As important points to consider, this means that an extensive calculation is needed for each joint comparison as it extends way beyond simply combining individual sample variant tables. In addition, just one dataset with very bad sequence quality, e.g. very low coverage depth, can seriously limit the resolution power of the phylogenetic analysis, as for that only regions of the reference genome with sequence information in all compared samples are taken into account.

Why don’t we just combine variant tables for a joint comparison?

Relying on fixed variant tables per isolate for a joint comparison and disregarding the actual full mapping information for each isolate will likely result in errors when calculating the pairwise distance between isolates, with the distance most often being overestimated.

As a practical example, imagine that a variant was called at position 1224 in dataset 'A', and due to poor sequence data the variant was not called in dataset 'B'. If simply comparing the respective variant tables against each other, there would be 1 SNP difference between the datasets. With the MTBseq approach, there are two possible outcomes. If position 1224 has good sequence data in the majority of isolates analyzed, the actual sequence information at position 1224 will be used for dataset 'B', resulting in a distance of 0 SNPs between 'A' and 'B'. In contrast, if position 1224 has poor sequence data in the majority of isolates, the position is discarded from the calculation, also resulting in a distance of 0 SNPs. This should illustrate one common cause for overestimating the distance between isolates in a joint comparison relying on simply combining fixed variant tables per isolate.

Handling regions with either deletions, missing coverage, or bad sequence data in one or more isolates is a challenge for SNP-based approaches, especially if the goal is continued surveillance. Disregarding these positions in the genome altogether from the joint analysis increases robustness and reliability, but decreases resolution, and necessitates continuous re-evaluation of the analysis if new samples are added. As the number of isolates will be ever growing in prospective approaches, this will likewise demand a rising cost in calculation power and time required. However, as explained above, simply assuming wild type sequence for the problematic regions of respective datasets will likely overestimate the distance between isolates (mostly due to of missing data or unclear base calls). As a more refined option, the respective positions could be blanked in pairwise comparisons only, but this would require knowledge about the sequence quality in these regions which is not present in plain variant tables. Basically, this would require genome-wide information in the variant report for each sample, i.e. at least genome-wide consensus information.

All in all, for us the strategy implemented in MTBSeq is a reasonable compromise for a comprehensive comparison. As discussed previously, this has the downside of requiring a complete recalculation of a joint analysis if a new dataset is added to it.