The *Mycobacterium tuberculosis* transposon sequencing database (MtbTnDB): a large-scale guide to genetic conditional essentiality

Adrian Jinich[a*#], Anisha Zaveri[b*], Michael A. DeJesus[c], Emanuel Flores-Bautista[d], Clare M. Smith[e], Christopher M. Sassetti[f], Jeremy M. Rock[c], Sabine Ehrt[b], Dirk Schnappinger[b], Thomas R. Ioerger[g], Kyu Rhee[a#]

a. Division of Infectious Diseases, Weill Department of Medicine, Weill-Cornell Medical College, New York, NY;
b. Department of Microbiology and Immunology, Weill Cornell Medical College, New York, NY;
c. Laboratory of Host-Pathogen Biology, The Rockefeller University, New York, NY;
d. Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA;
e. Department of Molecular Genetics and Microbiology, Duke University, Durham, NC;
f. Department of Microbiology and Physiological Systems, University of Massachusetts Medical School, Worcester, MA;
g. Department of Computer Science and Engineering, Texas A&M University, TX.

Abstract

Characterization of gene essentiality across different conditions is a useful approach for predicting gene function. Transposon sequencing (TnSeq) is a powerful means of generating genome-wide profiles of essentiality and has been used extensively in *Mycobacterium tuberculosis* (Mtb) genetic research. Over the past two decades, dozens of TnSeq screens have been published, yielding valuable insights into the biology of Mtb in vitro, inside macrophages, and in model host organisms. However, these Mtb TnSeq profiles are distributed across dozens of research papers within supplementary materials, which makes querying them cumbersome and assembling a complete and consistent synthesis of existing data challenging. Here, we address this problem by building a central repository of publicly available TnSeq screens performed in *M. tuberculosis*, which we call the Mtb transposon sequencing database (MtbTnDB). The MtbTnDB encompasses 64 published and unpublished TnSeq screens, and is standardized, open-access, and allows users easy access to data, visualizations, and functional predictions through an interactive web-app (www.mtbtndb.app). We also present evidence that (i) genes in the same genomic neighborhood tend to have similar TnSeq profiles, and (ii) clusters of genes with similar TnSeq profiles tend to be enriched for genes belonging to the same functional categories. Finally, we test and evaluate machine learning models trained on TnSeq profiles to guide functional annotation of orphan genes in Mtb. In addition to facilitating the exploration of conditional genetic essentiality in this important

human pathogen via a centralized TnSeq data repository, the MtbTnDB will enable hypothesis generation and the extraction of meaningful patterns by facilitating the comparison of datasets across conditions. This will provide a basis for insights into the functional organization of Mtb genes as well as gene function prediction.

Introduction

Assessing the essentiality profile of a gene across different conditions is often a useful first step towards determining its function. When determined for all genes in an organism, similar patterns of essentiality across a set of conditions can indicate a shared or common function, as in the case of enzymes in a metabolic pathway [1]. Evidence of such shared or related functions can be further informed by constructing gene-gene interaction networks consisting of genes that are essential on background mutant strains that harbor single or multiple gene deletions [2,3]. In the context of pathogenic bacteria, genes that are essential for virulence in clinically relevant conditions represent potential high priority targets for drug development [4,5].

For many bacterial organisms, including the human pathogen *Mycobacterium tuberculosis* (*Mtb*), one of the main approaches to investigate the effect of gene knockouts on fitness at genome scale is to generate pooled transposon insertion mutant libraries and measure the relative growth or survival via transposon sequencing (TnSeq) [3,6,7]. In TnSeq, a saturated transposon insertion library is generated and cultured in a defined experimental condition. Amplification and sequencing of the output and control libraries yields, for each genomic region, a ratio of experiment-to-control library insertion counts. These insertion ratios are analyzed statistically in order to categorize genes as conditionally essential or non-essential [2,8,9].

Two decades ago, Rubin et al. [10] adapted transposon mutagenesis, coupled to microarray-based genome-wide fitness measurements, to *Mtb*. This method was subsequently modified using next-generation sequencing to allow single base-pair mapping of insertions [1,11]. Since then, TnSeq has been used to identify *Mtb* conditionally essential genes using *in vitro* cultures, *in vivo* mouse models as well as macrophage infection assays, and in a wide variety of conditions, such as different carbon sources, stress conditions, and a diverse array of *Mtb* clinical and mutant strains (Table 1) [1,2,11–23]. Unfortunately, these valuable genetic datasets remain scattered in supplementary tables throughout literature, making rapid analysis of the conditional essentiality profiles of genes of interest time consuming. Additionally, a lack of analytical standardization makes comparison difficult even when

studies are identified, as it is often not clear how to properly compare essentiality calls that are made with different statistical approaches.

To address this gap, we compiled a wide array of publicly available *M. tuberculosis* TnSeq datasets into a single standardized, open-access database, which we call the *Mtb* transposon sequencing database (MtbTnDB) (Fig 1). The MtbTnDB contains 76 unique screens from 19 different publications and from the Functionalizing Lists of Unknown TB Entities (FLUTE) database (orca2.tamu.edu/U19), covering a range of experimental and physiological conditions (Table 1). To help facilitate access to the database for the TB research community, we developed an online tool that allows querying the database by either TnSeq screen or gene of interest and provides informative analyses and visualizations of the compiled data. We illustrate the potential value of the MtbTnDB for TB research through example applications, such as easily identifying orphan (i.e unannotated) genes that are conditionally essential in a given TnSeq screen. Finally, we apply unsupervised and supervised machine learning techniques to extract insights into the function of genes from their TnSeq profiles.

Results

Compiling and standardizing *M. tuberculosis* TnSeq datasets.

We set out to build a database that compiles all publicly available *M. tuberculosis* TnSeq datasets in a standardized, open-source format (Fig. 1) We assembled data for *Mtb* TnSeq screens encompassing a total of 76 comparisons between experimental and control conditions from two main sources. A first set of 55 comparisons were compiled from 19 publications [1,2,11–27]. The remaining 15 were obtained from the Functionalizing Lists of Unknown TB Entities (FLUTE) database [28].

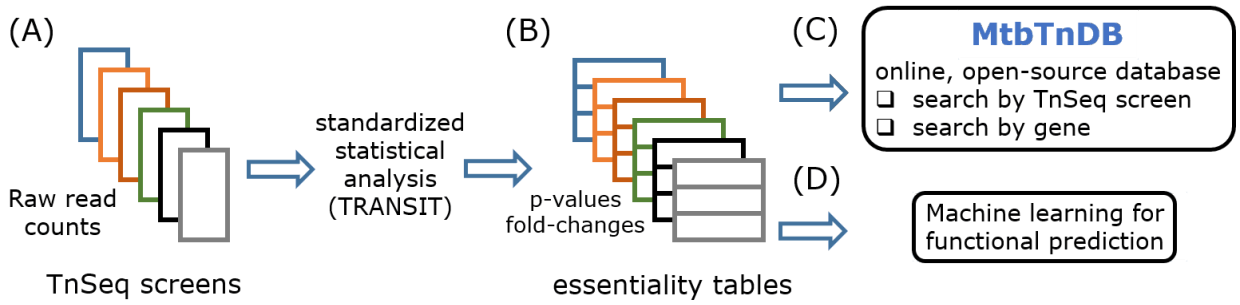

Fig. 1: Schematic diagram of compilation and standardization of *M. tuberculosis* TnSeq screens. (A) Raw sequencing reads were collected from publicly available TnSeq screens in the literature and in the FLUTE database. (B) Using a standardized statistical processing framework with TRANSIT [9], each set

of raw reads was converted into an essentiality table, with genes in the Mtb genome assigned log2 fold-change values (relative to specified control conditions) and p-values corrected for multiple hypothesis testing. (C) The set of TnSeq data can be queried in the MtbTnDB online portal, either by screen or by gene of interest. (D) The database is amenable to both supervised and unsupervised machine learning approaches to generate hypotheses about gene function.

The set of unique TnSeq screens in the MtbTnDB cover a wide range of different experimental conditions (Table 1). Out of 76 comparisons, the majority (47) were performed as *in vitro* bacterial culture experiments, while 27 were *in vivo* experiments using mouse infection models, and 2 used cell culture (macrophage and dendritic cell) infection models. The *in vitro* set of TnSeq screens covers many different culture media (carbon sources and limiting nutrients) and stress conditions (antibiotics, low oxygen tensions, acidic pH, nitrosative stress, iron limitation, and amino acid starvation). Although the large majority of the experiments were performed using the standard laboratory *Mtb* strain (H37Rv), one publication [20] performed comparative screens on eight different clinical isolates. A significant number of the TnSeq comparisons (34) were performed on knockout *Mtb* background strains, yielding information on genetic interaction networks underlying observed conditional essentiality profiles. Finally, for completion we also included data for 5 screens from the literature that were performed using microarray-based transposon site hybridization (TraSH) [24–27].

Table 1: The space of experimental conditions covered by the *Mtb* TnSeq Database.

model systems	*in vitro* (bacterial culture)	*in vivo* (mouse)	in macrophage or DC	
carbon sources	glycerol	cholesterol	trehalose	minimal media
stress conditions	antibiotics	hypoxia	amino acid starvation	nitrosative stress
		low pH	low iron	
M. tuberculosis strains	H37Rv	8 clinical isolates		
mouse strains	wildtype	MHCII-/-	C3HeB	BALB/cJ	Nos2-/-
background knockout strains (genetic interactions)	ponA1 ldtB ponA2 ΔRv0307c ΔRv0950c ΔRv0954	ΔRv1096 ΔRv1432 ΔRv1565c ΔRv2680 ΔRv3005c ctpC	otsA ΔRv3594 ΔRv3684 ΔRv3717 ΔRv3811 ΔRv3916c	marP eccD1 espI PE35 PPE68

Standardized statistical analysis of TnSeq screens using TRANSIT

In order to facilitate comparative analyses across different TnSeq screens, we set out to re-process the raw data for as many screens as possible using a single standardized statistical method based on TRANSIT, a Python-based TnSeq analysis tool (see Methods for details). Although TRANSIT accommodates a variety of statistical approaches for making essentiality calls, we utilized the resampling method to compare every experimental condition to an appropriate reference or control condition. Briefly, resampling compares read-counts between two different conditions utilizing a permutation test to identify significant differences in the mean read-counts [9]. Resulting hits are labelled as *conditionally* essential genes: genes that are essential in one background condition (experiment) but not another (reference or control). By comparing against a suitable reference or control condition, we sought to reduce potential biases that can exist due to differences in mutant libraries. Whenever possible, the reference condition was chosen to be the one described in the original TnSeq screen. When no suitable reference or control condition was specified or included in the screen, we utilized a fully saturated in-vitro dataset [29] as the control condition.

In order to analyze the effect of our standardized processing of the underlying TnSeq sequencing data, we compared the number of conditionally essential genes detected in each screen (using fixed threshold values for the corrected p-value<=0.05, and a log(2)-fold change>=1) against those reported in the corresponding original, individual datasets (Table S1). Across all screens analyzed, we find that the standardized data and the original analyses differ in around 2% of the conditional essentiality calls. Approximately half of these are cases where the standardized dataset suggests that a gene is conditionally essential in a particular screen, whereas the original analysis did not, and the other half of the cases correspond to the opposite situation. A few individual screens have a relatively high number of non-consensus essentiality calls between the standardized and original datasets. In particular, one TnSeq screen in an *in vivo* mouse model [13] has 403 genes that were called essential in the original datasets, but not in the standardized analysis. One possible reason behind this large inconsistency is the different statistical methods used to analyze the sequencing data. We note that both the original and the standardized data sets are included in the MtbTnDB (see section on online database).

In addition, we inspected the standardized TnSeq screens in search of batch effects, i.e. variations in the datasets that occur independently of biological signal and noise. Dimensionality reduction of data vectors representing each TnSeq screen (Methods) revealed that, of the 64 screens included in the database, a set of eight TnSeq screens formed a distinctly separate cluster (Fig. S1). This cluster consists

of screens where essentiality counts in eight clinical strains are compared relative to a control strain (H37Rv), which could either be indicative of experimental batch effects or true biological differences between domesticated laboratory strains and clinical isolates.

The statistics of conditional essentiality across the MtbTnDB

Fig. 2A shows the distribution of conditional essentiality calls across all genes in the *Mtb* genome. We find that 2453 genes are not labelled as conditionally essential in any of our standardized screens. We note that a significant subset of these (425) are genes that are essential or have a significant growth defect in the fully saturated *in-vitro* screen [29] that we used as a control (reference) condition for a large number of comparisons. Thus, despite being essential for growth in standard in-vitro culture conditions, these genes are not detected as conditionally essential in any screen in the MtbTnDB: i.e there is no significant difference in their degree of essentiality in standard in-vitro culture in comparison to the experimental conditions covered in the database. The rest of the genes with zero conditionally essential calls likely represent a mixture of putatively dispensable genes and (i) genes with essential roles in experimental conditions that are outside the coverage of our database, (ii) false negatives, i.e. genes that do not meet the threshold of statistical significance due to experimental factors like low coverage.

At the tail of the distribution, we find approximately 120 genes that are conditionally essential in 10 or more comparative TnSeq screens (SI Table 2). Among these (Fig. 2B) we find a strong enrichment for phthiocerol synthesis polyketide synthase type I genes (*ppsA-E*), with all members of these PDIM biosynthetic pathway genes having at least 24 conditional essentiality counts. One possible explanation for this is the fact that loss of PDIM biosynthesis genes *in vitro* confers a fitness advantage to *Mtb* [30]. The fully saturated in-vitro dataset [34] that was used as the control condition for a large number of comparisons contains PDIM biosynthesis genes, but some of these genes could have been lost in several of the experimental conditions, leading to few or no detectable transcription insertion counts and resulting in false positive conditionally essential call. A second set of genes with a high number of conditionally essential calls are the members of the mycobactin siderophore biosynthetic pathway (*mbtA, mbtB, mtbD, mbtE, mbtF*) [31] (Fig. 2B). Interestingly, in more than half of these instances the mycobactin biosynthesis genes are detected as a significant hit because they are *less* essential (i.e. p-val (adjusted) <= 0.05 and log2FC >= 1) implying their loss provides a fitness advantage across a range of different conditions. One might expect that iron-scavenging would not be required across several *in-vitro* conditions, as most media used in TnSeq experiments is replete with iron.

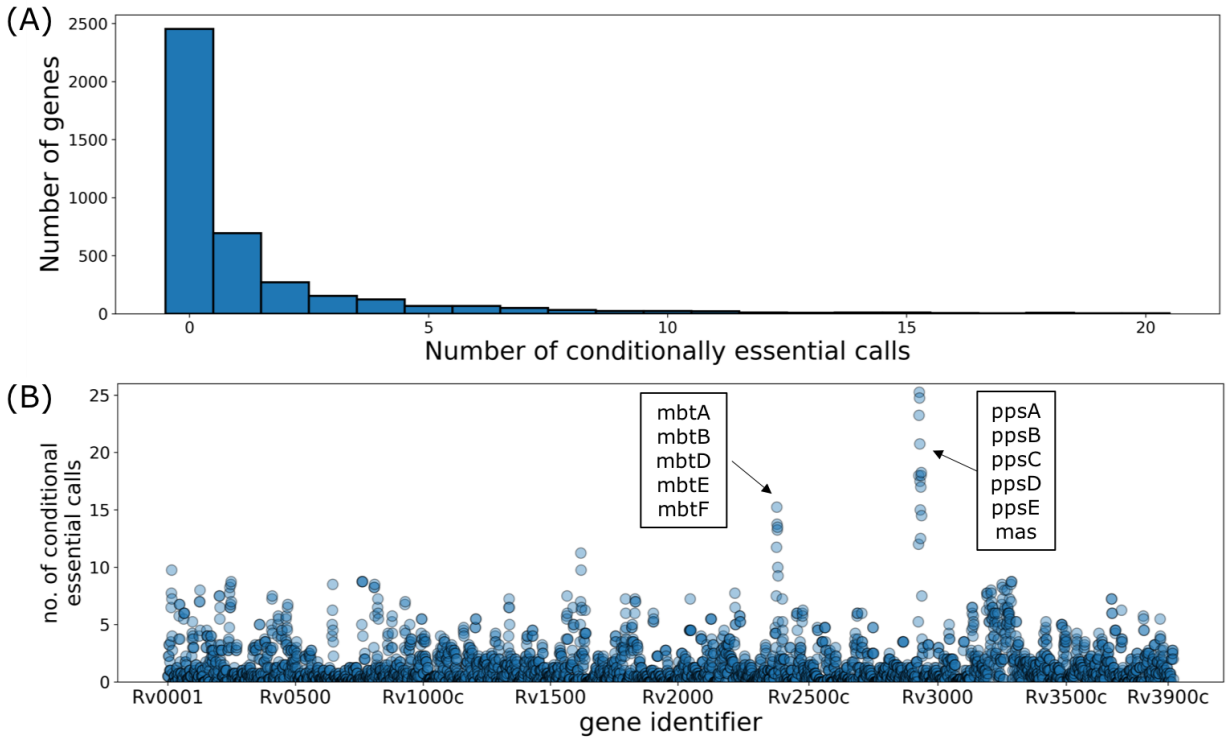

Fig. 2: Distribution of conditional essentiality calls in the MtbTnDB across all genes. (A) Distribution of the number of conditionally essential calls across all genes in the *Mtb* genomes (B) Number of conditional essentiality calls vs. position in the genome across the MtbTnDB for all H37Rv *Mtb* genes. The mycobactin and PDIM biosynthetic operons, which tend to have a large number of conditionally essential calls, are highlighted.

MtbTnDB, an online database of standardized Mtb TnSeq screens

To allow efficient querying of the standardized data, we developed an MtbTnDB web interface (www.mtbtndb.app). The interface allows two methods of querying the database - by screen (Fig. 3) and by gene. Each screen is a pairwise comparison between an experimental and control condition. The "Analyze datasets" tab provides an information-rich and interactive overview of the selected comparison. An "About this dataset" section includes a description of the screen, links to the original publication, and the number of replicates used for both the experimental and control conditions. A volcano plot and accompanying table allow interrogation into differentially represented mutants. Sliders for significant log2FC and qval cut-offs allow interactive coloring of data points on the volcano plot. Mutants of interest can also be highlighted on the volcano plot by selection in the accompanying table. We also provide two plots depicting broader analyses of the data. First, a bar plot depicts the enrichment of COG functional categories in over/under-represented mutants. Second, a bubble plot allows identification of the subset of

genes that are both conditionally essential in this analysis and the least annotated subset, providing useful pointers for follow-up experiments. An example case using this type of plot is provided in the next section.

The second functionality provided in the web interface is to query genes of interest and list the set of TnSeq screens in which mutants for this gene are differentially represented. In addition to log2FC and q-values we also provide track-views, which are visual depictions of the magnitude and location of insertions within the selected gene.

The data used in MtbTnDB is provided in the "About" tab, and users can also choose to download individual screens from the "Analyse datasets" tab. We also note the modularity of the database and its ability to accommodate additional datasets in the future, potentially including conditional essentiality screens using CRISPRi.

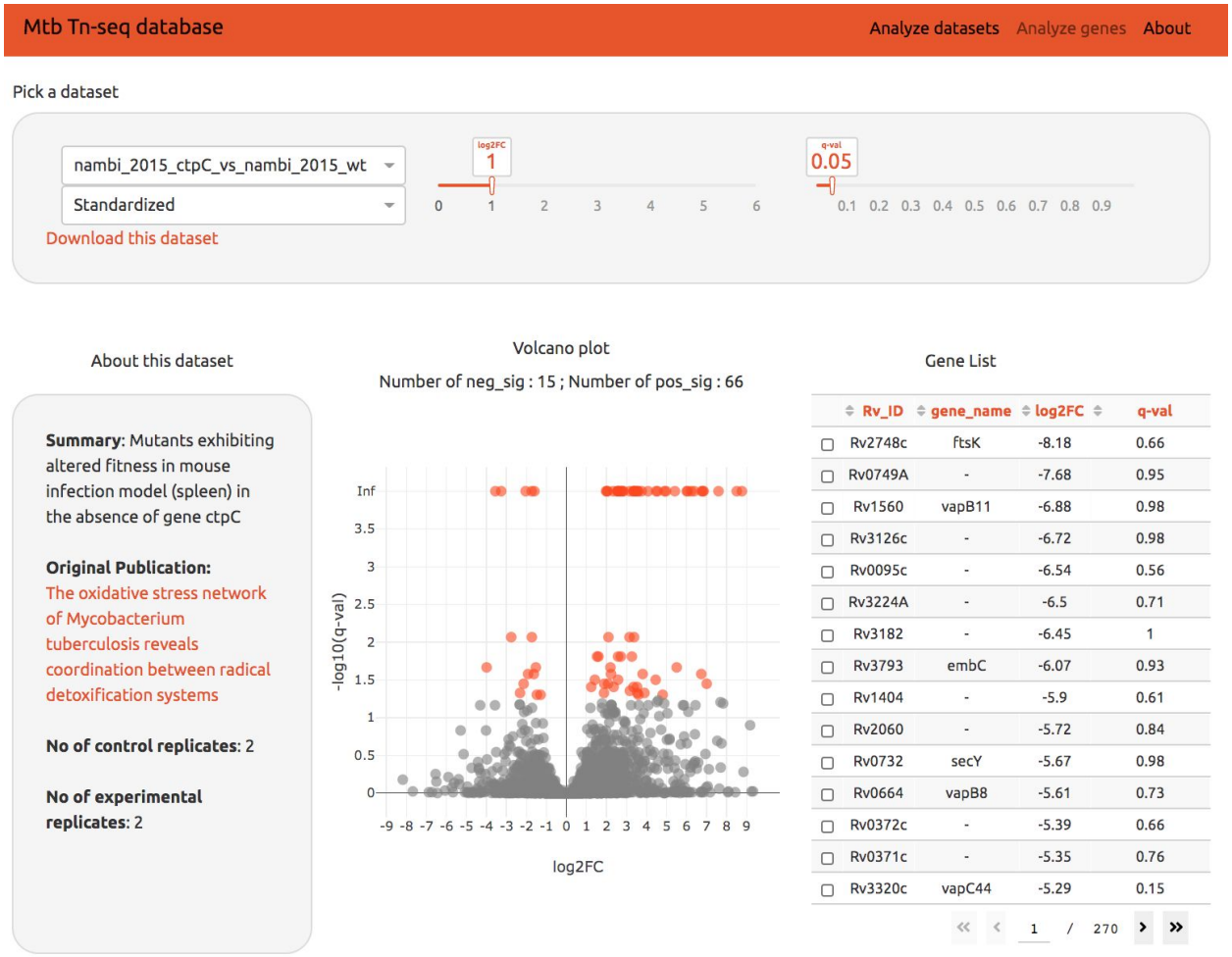

Fig. 3: An online explorer of TnSeq datasets. A screenshot of the "Analyze datasets" modality is shown. Users can select which TnSeq screen to explore, whether to display either the standardized or the original publication's dataset, as well as significance thresholds for log2 fold-changes and q-values. A brief summary of the TnSeq screen is shown in the left panel, alongside a link to the original publication and the number of replicates for both control and experimental conditions. An interactive volcano plot highlights conditionally essential genes, which can also be accessed and selected through the table on the right.

Example usage: selecting experimental conditions to study unannotated genes.

A large fraction of genes in the *Mtb* genome remains unannotated, and the MtbTnDB could help annotate such "orphan" genes by guiding the design of experimental conditions in which to assay their function. This is motivated by the notion that experimental conditions in which a mutant strain displays a growth or survival defect are more likely to reveal the orphan gene's physiological role. For instance, if an enzyme of unknown function participates in cholesterol catabolism, the probability of discovering its function will be highest if the mutant strain's metabolic profile is obtained when grown with cholesterol as its primary carbon source.

To help select experimental conditions in which to study orphan genes, we used experimental annotation scores from Uniprot, consisting of 5 categories ranging from least to most well characterized (see Methods), to classify genes. For every TnSeq screen in the MtbTnDB, we then obtain the unknown essentials: the set of genes that are both conditionally essential (in that experimental screen) and fall in the category of least well characterized genes (Fig. 4, top-left corner). The number of unknown conditional essentials varies across each TnSeq screen in the MtbTnDB, with 10 screens having more than 30 conditionally essential genes with the lowest annotation score. In particular, two mouse *in vivo* screens have more than 65 genes that fall in the most poorly annotated category, underscoring the fact that the function of many potential drug targets remains to be discovered. This also illustrates how the MtbTnDB can be used to systematically select experimental conditions that result in measurable growth phenotypes for orphan genes of interest for further study.

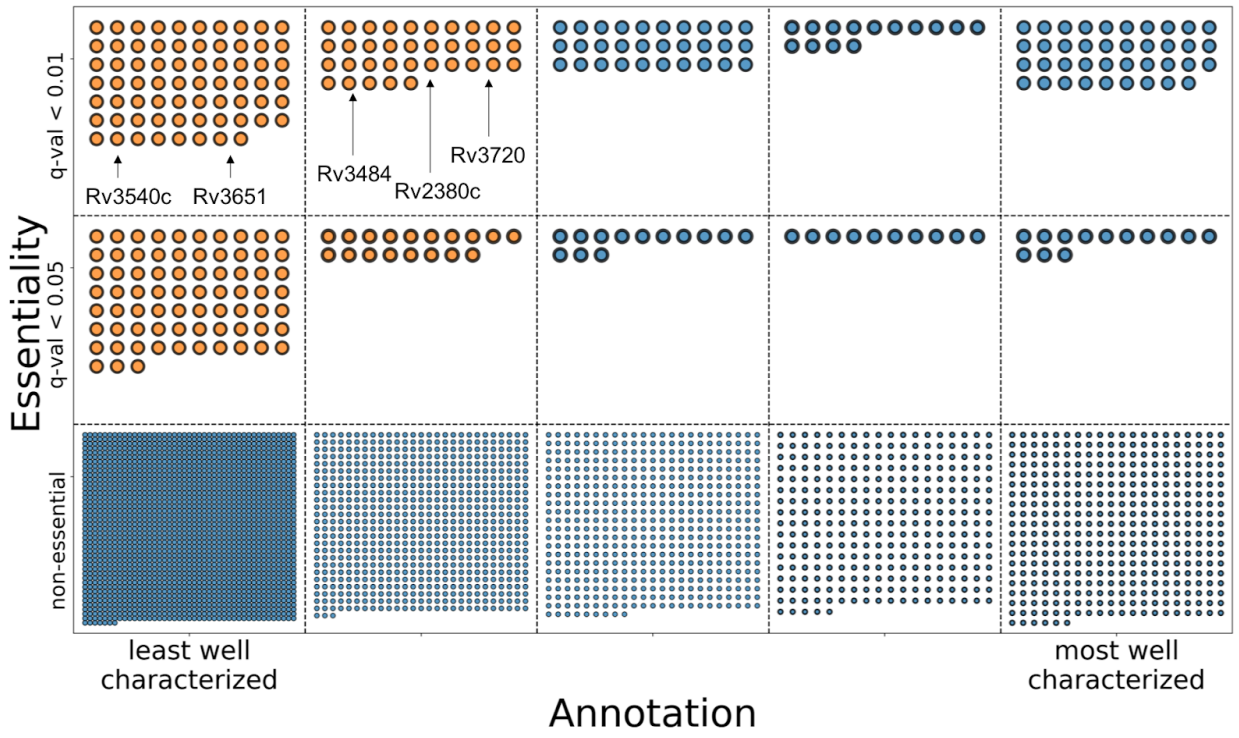

Fig. 4: Mapping out conditionally essential genes of unknown function. Data shown corresponds to a single TnSeq screen in the MtbTnDB: an H37Rv *in vivo* (mouse) screen [2]). Genes are represented as circles, and are binned along the x-axis into five distinct categories according to their annotation level and along the y-axis according to their conditional essentiality (the exact position within each bin/category is not meaningful). The least well characterized genes (i.e orphans) that are conditionally essential in this particular TnSeq screen are shown in orange (the gene identifiers for five example genes are shown).

Predicting functional categories with the MtbTnDB

We asked whether similar TnSeq conditional essentiality profiles are indicative of similar biological function, where a conditional essentiality profile is defined as the set of (conditional) essentiality calls across all standardized TnSeq screens; i.e. a 64-dimensional vector for each gene in the *Mtb* genome. To explore this, we applied a uniform manifold approximation and projection (UMAP), a non-linear dimensional reduction tool [32], to the standardized TnSeq data. We hypothesized that, since genes belonging to the same operons are functionally related, neighboring genes in the genome are likely to have more similar TnSeq conditional essentiality profiles than random pairs of genes on average (Fig. 5A). To test this, we compared the distribution of Euclidean distances in the UMAP projection between (i) pairs of genes that are in the same genomic neighborhood (i.e. apart from each other in the genome by a fixed cutoff genomic distance) and (ii) pairs of genes chosen at random from the full *Mtb* genome. We

indeed find that neighboring genes tend to have more similar TnSeq profiles than expected by chance ($p=10^{-14}$, Kolmogorov-Smirnov test).

Next, using k-means clustering, we grouped genes by the similarity of their TnSeq profiles as captured by their coordinates in the UMAP projection. An example UMAP projection with 19 different clusters is shown in (Fig. 5B). Using the COG [33] and Tuberculist [34] *Mtb* genome annotations, we find that several clusters are significantly enriched for a number of different functional categories, including, "nucleotide transport and metabolism", "intermediary metabolism and respiration", and "secondary metabolites and biosynthesis", among others. Importantly, our observation of clusters enriched for functional categories is statistically significant ($p<0.05$, permutation test), and on average, zero categories were enriched when annotations were shuffled and re-analyzed for overlap with the UMAP clusters (Fig. S2). Taken together, these analyses serve as evidence that the TnSeq profiles capture functional relationships between genes, and encode information about the underlying operonic structure of the *Mtb* genome.

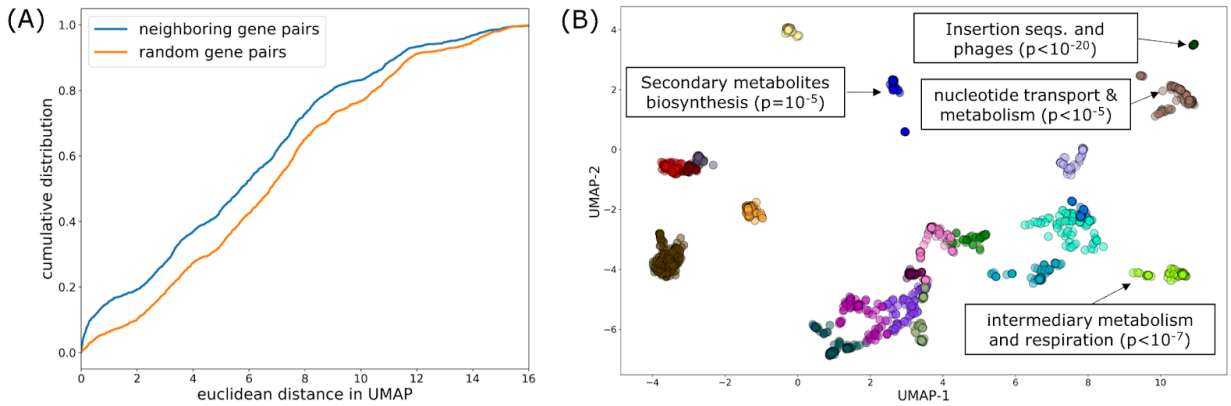

Fig. 5: Genes with similar TnSeq profiles tend to have similar functions. Neighboring genes in the genome have more similar TnSeq profiles, and clusters of genes with similar TnSeq profiles are enriched for annotated genes belonging to the same functional categories. (A) Distribution of Euclidean distances (in UMAP projection) for pairs of random genes (orange) and neighboring genes (i.e. less than 3 genes away from each other in the genome). The two distributions are significantly different ($p<10^{-6}$, Kolmogorov-Smirnov test). (B) Uniform manifold approximation and projection (UMAP) of TnSeq profiles. Genes with at least one conditional essentiality call in the MtbTnDB were clustered using k-means according to their UMAP coordinates and color-coded. Four clusters enriched for gene functional categories are shown.

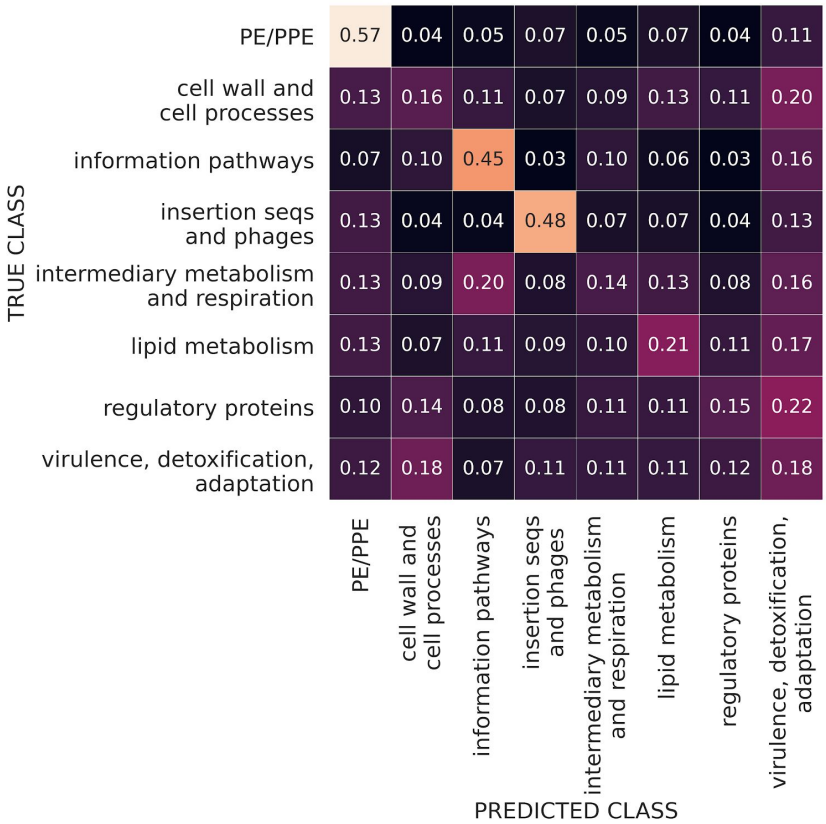

Fig. 6: Prediction of gene function with machine learning. Confusion matrix showing the fraction of observations predicted to be classified in each functional category. Predictions were generated using an ensemble machine learning model consisting of a logistic regression and random forest classifier. The number of genes in each functional category are: PE/PPE (166); cell wall and cell processes (771); information pathways (242); insertions sequences and phages (146); intermediary metabolism and respiration (934); lipid metabolism (271); regulatory proteins (198); virulence, detoxification, adaptation (220).

Building on the unsupervised analysis described above, we reasoned that since genes with similar physiological functions have similar essentiality profiles, the information encoded in the Mtb TnSeq Matrix could be used to help computationally annotate orphan genes. Towards this end, we trained a supervised classification model which takes as input a gene's log2FC values from the 64 screens in the database and predicts the functional category from the Tuberculist genome annotation that the gene belongs to. Our model (see Methods), a stacked meta-learner that combines predictions from logistic regression and random forest sub-models, achieved an accuracy of 45-60% (3-fold cross-validation) for three Tuberculist categories: 'PE/PPE', 'information pathways', and 'insertion sequences and phages', as

assessed by cross-validation (Fig. 6). We note that (i) our prediction is completely blind to the coding sequence, and that (ii) as a comparative baseline, the accuracy of an unbiased random classifier would be 12.5% for each of the 8 categories. Thus, while improvements in predictive accuracy can be achieved by incorporating other sources of data, TnSeq essentiality profiles can potentially help guide the functional annotation of unknown genes in Mtb. In addition, the predictive power of machine learning models trained on TnSeq data will likely improve as the database is populated with more conditional essentiality screens.

Discussion

To facilitate access to and the ability to gain insight from the vast amount of information stored in *M. tuberculosis* genetic essentiality screens, we compiled a large set of publicly available TnSeq screens into a single, standardized, and open-access database, the *Mtb* transposon sequencing database (MtbTnDB). While other TnSeq databases exist [35], these contain significantly fewer *Mtb* specific screens and do not address the issue of standardization of raw sequencing data. We envision periodically updating the MtbTnDB by adding publicly available TnSeq screens as they become available in the literature: researchers wishing to contribute their screen to the database would simply need to provide the sequence files along with a basic description of the screening conditions.

Although we aimed to maximize the degree of standardization, we note that there are upstream experimental factors that introduce variability into the screens, placing inherent limits in our ability to standardize the data. For instance, the transposon libraries differ in saturation (the number of unique insertion sites with one or more mutants present in the library), or in protocol details, such as whether barcoding was utilized, or the specific sequencing technology used to obtain the reads. These differences may introduce artifacts, such as PCR over-amplification, or simply reduce the amount of data that is available (in the case of libraries with low coverage) which ultimately will affect the statistical power of downstream analyses.

One limitation of the current set of TnSeq screens included in the MtbTnDB is a bias towards a subspace of experimental conditions. Future efforts should therefore be directed towards covering a wider and diverse region of experimental condition space. Towards this end, a recent TnSeq screen assaying iron uptake in *M. tuberculosis* [36], as well as recent technical developments that couple TnSeq with random DNA barcoding (RB-TnSeq) allowing parallel screening across dozens of conditions [37] are very encouraging.

It is worthwhile to point out that the information afforded by TnSeq experiments is quite different and complementary from that of RNAseq (transcriptomics). Transcriptomics identifies sets of genes whose expression changes in response to changes in conditions, which can be clustered into sets of co-regulated genes (or regulons) and sometimes correlated with transcription factors [2,38]. There is a large amount of transcriptomic data (DNA microarrays and RNAseq) that has been collected for Mtb in different conditions over the years, including stress conditions like hypoxia and antibiotic exposure, in macrophages and mouse models, knockout strains, among others [39]. TnSeq identifies the essentiality of genes in a given condition. In the extreme, essential genes are those which are indispensable for growth in the condition and hence cannot tolerate transposon insertions. Additionally, there might be some genes that exhibit a quantitative reduction in transposon insertion counts, which is interpreted to indicate a fitness defect (growth impairment) caused by the disruption of the ORF [24]. Hence, TnSeq reveals information about which genes play functional roles in pathways required for survival in the condition. While both technologies can be efficiently scaled up to profile genome-wide cellular responses to changes in environmental conditions, the assays assess different information about genes (essentiality versus expression), and the lists of hits (conditionally essential genes vs. differentially regulated genes) often exhibit little overlap [40]. A gene can be differentially expressed without being conditionally essential and, conversely, a gene can become conditionally essential without a significant change in expression in the new condition. Efforts to combine TnSeq and RNAseq to enhance function prediction are on-going.

The prospect of combining high-throughput phenotypic screening with machine learning to illuminate orphan gene function is promising. In this work, we tested how well the MtbTnDB could functionally classify unannotated *Mtb* genes using supervised machine learning models and found limited prediction accuracy. One possible explanation for this could be the quality of the functional annotations themselves, and future directions would include combining the MtbTnDB with complementary sources of data, such gene expression datasets, in order to improve the prediction of orphan gene function. We envision that the essentiality patterns encoded in the collected and standardized TnSeq screens will help researchers unravel the higher-order, physiological function of genes in the *Mtb* genome.

Methods

Standardization of TnSeq datasets

To analytically standardize and process available TnSeq raw sequencing data, we selected TRANSIT, a Python-based TnSeq analysis tool, and TPP, a software tool for processing raw reads [9].

We were able to obtain either the original raw sequencing data or read-counts in a format suitable for analysis, such Wiggle Track Format (.wig), for 64 TnSeq screens. For conditions where raw reads were available, we first used Tn-Seq Pre-Processor (TPP), a preprocessing tool included with TRANSIT, to map them all to the genome sequence of the standard H37Rv reference strain and obtain .wig formatted files.

Those datasets for which we could not obtain raw sequencing data were not included for analysis but are referenced in the MtbTnDB website to allow users to look at their results. As an additional quality control step, we only included screens for which the transposon insertion density was higher than 25%. Raw reads were mapped with TPP to the H37Rv genome using default settings. We note that for screens that did not use H37Rv, genes that are deleted in the experimental strain but not in the control strain - or vice versa - may cause false positive conditional essentiality calls. The resulting .wig files produced by TPP were analyzed using the resampling in TRANSIT, which compares two experimental conditions to find significantly different read-counts in a gene. We also included pseudocounts in the resampling pipeline, using the default of PC=1 in TRANSIT. Whenever possible, a condition (experiment) was compared to the reference dataset (control) specified in a TnSeq screen's source publication. For publications that did not specify a reference or control condition, a highly-saturated in-vitro dataset [29] was used. Resampling was run using default settings, using TTR normalization and 10,000 permutations. P-values were automatically adjusted for multiple hypothesis testing by TRANSIT using the Benjamini-Hochberg procedure. The resulting adjusted p-values and log2 fold-changes estimated by resampling were compiled together for all comparisons (SI Tables 3 and 4).

We labelled genes as conditionally essential in a given TnSeq screen the following two conditions are satisfied: p-val (adjusted) <= 0.05 and *abs*(log2FC)>=1, where $log2FC = log_2(\frac{experiment+pseudocounts}{control+pseudocounts})$, i.e. the logarithm base 2 of the ratio of insertion counts (adding the fixed so-called pseudocounts parameter), and *abs* is the absolute value. We note that the use of the absolute value in the threshold condition results in some selected genes having positive log2FC values, meaning they are detected as significantly *less* essential than expected by chance in that particular condition, with their loss providing a fitness *advantage*.

To inspect the TnSeq screens for potential batch effects, each TnSeq screen was represented by an N-dimensional vector of log2FC (where N is the number of genes in the Mtb genome) and we performed dimensionality reduction of these vectors using the uniform manifold approximation and projection (UMAP) algorithm.

MtbTnDB website and open-source code

The MtbTnDB online portal was built using Dash, a tool for web-based analytical apps in Python. All source code used to build the online portal as well as all analyses mentioned in this work are available at https://github.com/ajinich/mtb_tn_db_demo.

Classification of genes into annotation categories

We classified genes in the H37Rv genome into a set of five categories ranging from least to most well characterized using annotations scores from the UniProt Knowledgebase (UniProtKB) [41], which evaluates available experimental evidence for each gene at the protein level.

Clustering of genes with UMAP

To perform the unsupervised learning analysis of the standardized TnSeq profiles using UMAP, we first removed all genes (2453) that have zero conditional essentiality calls across all 64 standardized screens. We then used the standardized binary TnSeq profiles for the remaining number of genes input into UMAP (with 3 components), as implemented in Python. Having obtained the UMAP components, we performed K-means clustering in the 2-dimensional projection, selecting the optimal number of clusters that maximizes the Silhouette Coefficient [42]. We used Fischer's (hypergeometric) enrichment test to look for potential enrichments of COG or Tuberculist annotation functional categories within each cluster. Finally, to further evaluate the statistical significance of the observed enrichments within clusters, we generated randomized versions of the UMAP dataset by fixing the UMAP coordinates and cluster identities, shuffling the gene identifiers and annotation information associated to each point in the UMAP projection, and repeating the statistical enrichment test.

Prediction of gene function:

To fit a supervised machine learning model to the standardized TnSeq profiles, we used the log2FC values from the standardized matrix as inputs. Genes containing missing values in some screens were dropped (n=84). We used Tuberculist annotation functional categories as classification labels and dropped genes belonging to the 'unknown' (n=14) or 'conserved hypothetical' (n=1020) categories. The resulting matrix containing 2937 genes was used for prediction.

We used two sub-models to predict functional categories: a logistic regression classifier with lasso regularization and a random forest classifier [43]. Both methods were used as implemented in the scikit-learn library in Python. We used SMOTE [44] to correct for class imbalance along with 3-fold cross-validation. The outputs of each of the sub-models were subsequently fit to a logistic regression classifier to create an ensemble model for final predictions.

References

1. Griffin JE, Gawronski JD, Dejesus MA, Ioerger TR, Akerley BJ, Sassetti CM. High-resolution phenotypic profiling defines genes essential for mycobacterial growth and cholesterol catabolism. PLoS Pathog. 2011;7: e1002251.

2. DeJesus MA, Nambi S, Smith CM, Baker RE, Sassetti CM, Ioerger TR. Statistical analysis of genetic interactions in Tn-Seq data. Nucleic Acids Res. 2017;45: e93.

3. van Opijnen T, Lazinski DW, Camilli A. Genome-Wide Fitness and Genetic Interactions Determined by Tn-seq, a High-Throughput Massively Parallel Sequencing Method for Microorganisms. Current Protocols in Microbiology. 2017. pp. 1E.3.1–1E.3.24. doi:10.1002/9780471729259.mc01e03s36

4. Jacobs MA, Alwood A, Thaipisuttikul I, Spencer D, Haugen E, Ernst S, et al. Comprehensive transposon mutant library of Pseudomonas aeruginosa. Proc Natl Acad Sci U S A. 2003;100: 14339–14344.

5. Salama NR, Shepherd B, Falkow S. Global transposon mutagenesis and essential gene analysis of Helicobacter pylori. J Bacteriol. 2004;186: 7926–7935.

6. Chao MC, Abel S, Davis BM, Waldor MK. The design and analysis of transposon insertion sequencing experiments. Nat Rev Microbiol. 2016;14: 119–128.

7. Cain AK, Barquist L, Goodman AL, Paulsen IT, Parkhill J, van Opijnen T. A decade of advances in transposon-insertion sequencing. Nat Rev Genet. 2020;21: 526–540.

8. Barquist L, Mayho M, Cummins C, Cain AK, Boinett CJ, Page AJ, et al. The TraDIS toolkit: sequencing and analysis for dense transposon mutant libraries. Bioinformatics. 2016;32: 1109–1111.

9. DeJesus MA, Ambadipudi C, Baker R, Sassetti C, Ioerger TR. TRANSIT--A Software Tool for Himar1 TnSeq Analysis. PLoS Comput Biol. 2015;11: e1004401.

10. Rubin EJ, Akerley BJ, Novik VN, Lampe DJ, Husson RN, Mekalanos JJ. In vivo transposition of mariner-based elements in enteric bacteria and mycobacteria. Proc Natl Acad Sci U S A. 1999;96: 1645–1650.

11. Zhang YJ, Ioerger TR, Huttenhower C, Long JE, Sassetti CM, Sacchettini JC, et al. Global assessment of genomic regions required for growth in Mycobacterium tuberculosis. PLoS Pathog. 2012;8: e1002946.

12. DeJesus MA, Ioerger TR. A Hidden Markov Model for identifying essential and growth-defect

regions in bacterial genomes from transposon insertion sequencing data. BMC Bioinformatics. 2013;14: 303.

13. Zhang YJ, Reddy MC, Ioerger TR, Rothchild AC, Dartois V, Schuster BM, et al. Tryptophan Biosynthesis Protects Mycobacteria from CD4 T-Cell-Mediated Killing. Cell. 2013. pp. 1296–1308. doi:10.1016/j.cell.2013.10.045

14. Kieser KJ, Baranowski C, Chao MC, Long JE, Sassetti CM, Waldor MK, et al. Peptidoglycan synthesis in Mycobacterium tuberculosis is organized into networks with varying drug susceptibility. Proc Natl Acad Sci U S A. 2015;112: 13087–13092.

15. Mendum TA, Wu H, Kierzek AM, Stewart GR. Lipid metabolism and Type VII secretion systems dominate the genome scale virulence profile of Mycobacterium tuberculosis in human dendritic cells. BMC Genomics. 2015;16: 372.

16. Nambi S, Long JE, Mishra BB, Baker R, Murphy KC, Olive AJ, et al. The Oxidative Stress Network of Mycobacterium tuberculosis Reveals Coordination between Radical Detoxification Systems. Cell Host Microbe. 2015;17: 829–837.

17. Korte J, Alber M, Trujillo CM, Syson K, Koliwer-Brandl H, Deenen R, et al. Trehalose-6-Phosphate-Mediated Toxicity Determines Essentiality of OtsB2 in Mycobacterium tuberculosis In Vitro and in Mice. PLoS Pathog. 2016;12: e1006043.

18. Xu W, DeJesus MA, Rücker N, Engelhart CA, Wright MG, Healy C, et al. Chemical Genetic Interaction Profiling Reveals Determinants of Intrinsic Antibiotic Resistance in Mycobacterium tuberculosis. Antimicrob Agents Chemother. 2017;61. doi:10.1128/AAC.01334-17

19. Mishra BB, Lovewell RR, Olive AJ, Zhang G, Wang W, Eugenin E, et al. Nitric oxide prevents a pathogen-permissive granulocytic inflammation during tuberculosis. Nat Microbiol. 2017;2: 17072.

20. Carey AF, Rock JM, Krieger IV, Chase MR, Fernandez-Suarez M, Gagneux S, et al. TnSeq of Mycobacterium tuberculosis clinical isolates reveals strain-specific antibiotic liabilities. PLoS Pathog. 2018;14: e1006939.

21. Rittershaus ESC, Baek S-H, Krieger IV, Nelson SJ, Cheng Y-S, Nambi S, et al. A Lysine Acetyltransferase Contributes to the Metabolic Adaptation to Hypoxia in Mycobacterium tuberculosis. Cell Chem Biol. 2018;25: 1495–1505.e3.

22. Bellerose MM, Baek S-H, Huang C-C, Moss CE, Koh E-I, Proulx MK, et al. Common Variants in the Glycerol Kinase Gene Reduce Tuberculosis Drug Efficacy. MBio. 2019;10. doi:10.1128/mBio.00663-19

23. Minato Y, Gohl DM, Thiede JM, Chacón JM, Harcombe WR, Maruyama F, et al. Genomewide Assessment of Mycobacterium tuberculosis Conditionally Essential Metabolic Pathways. mSystems. 2019;4. doi:10.1128/mSystems.00070-19

24. Sassetti CM, Boyd DH, Rubin EJ. Genes required for mycobacterial growth defined by high density mutagenesis. Mol Microbiol. 2003;48: 77–84.

25. Sassetti CM, Rubin EJ. Genetic requirements for mycobacterial survival during infection. Proc Natl

Acad Sci U S A. 2003;100: 12989–12994.

26. Rengarajan J, Bloom BR, Rubin EJ. Genome-wide requirements for Mycobacterium tuberculosis adaptation and survival in macrophages. Proc Natl Acad Sci U S A. 2005;102: 8327–8332.

27. Joshi SM, Pandey AK, Capite N, Fortune SM, Rubin EJ, Sassetti CM. Characterization of mycobacterial virulence genes through genetic interaction mapping. Proc Natl Acad Sci U S A. 2006;103: 11760–11765.

28. FLUTE: Main Page. [cited 9 Feb 2020]. Available: https://orca2.tamu.edu/U19/

29. DeJesus MA, Gerrick ER, Xu W, Park SW, Long JE, Boutte CC, et al. Comprehensive Essentiality Analysis of the Mycobacterium tuberculosis Genome via Saturating Transposon Mutagenesis. MBio. 2017;8. doi:10.1128/mBio.02133-16

30. Domenech P, Reed MB. Rapid and spontaneous loss of phthiocerol dimycocerosate (PDIM) from Mycobacterium tuberculosis grown in vitro: implications for virulence studies. Microbiology. 2009;155: 3532–3543.

31. De Voss JJ, Rutter K, Schroeder BG, Su H, Zhu Y, Barry CE 3rd. The salicylate-derived mycobactin siderophores of Mycobacterium tuberculosis are essential for growth in macrophages. Proc Natl Acad Sci U S A. 2000;97: 1252–1257.

32. McInnes L, Healy J, Saul N, Großberger L. UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software. 2018. p. 861. doi:10.21105/joss.00861

33. Tatusov RL, Galperin MY, Natale DA, Koonin EV. The COG database: a tool for genome-scale analysis of protein functions and evolution. Nucleic Acids Res. 2000;28: 33–36.

34. Lew JM, Kapopoulou A, Jones LM, Cole ST. TubercuList – 10 years after. Tuberculosis. 2011. pp. 1–7. doi:10.1016/j.tube.2010.09.008

35. Luo H, Lin Y, Liu T, Lai F-L, Zhang C-T, Gao F, et al. DEG 15, an update of the Database of Essential Genes that includes built-in analysis tools. Nucleic Acids Res. 2021;49: D677–D686.

36. Zhang L, Hendrickson RC, Meikle V, Lefkowitz EJ, Ioerger TR, Niederweis M. Comprehensive analysis of iron utilization by Mycobacterium tuberculosis. PLoS Pathog. 2020;16: e1008337.

37. Price MN, Wetmore KM, Waters RJ, Callaghan M, Ray J, Liu H, et al. Mutant phenotypes for thousands of bacterial genes of unknown function. Nature. 2018;557: 503–509.

38. Rustad TR, Minch KJ, Ma S, Winkler JK, Hobbs S, Hickey M, et al. Mapping and manipulating the Mycobacterium tuberculosis transcriptome using a transcription factor overexpression-derived regulatory network. Genome Biol. 2014;15: 502.

39. Moretto M, Sonego P, Dierckxsens N, Brilli M, Bianco L, Ledezma-Tejeida D, et al. COLOMBOS v3.0: leveraging gene expression compendia for cross-species analyses. Nucleic Acids Res. 2016;44: D620–3.

40. Jensen PA, Zhu Z, van Opijnen T. Antibiotics Disrupt Coordination between Transcriptional and

Phenotypic Stress Responses in Pathogenic Bacteria. Cell Rep. 2017;20: 1705–1716.

41. UniProt Consortium. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res. 2019;47: D506–D515.

42. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987. pp. 53–65. doi:10.1016/0377-0427(87)90125-7

43. Breiman L. Random Forests. Mach Learn. 2001;45: 5–32.

44. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research. 2017;18(1):559-63.

Supplementary Figures

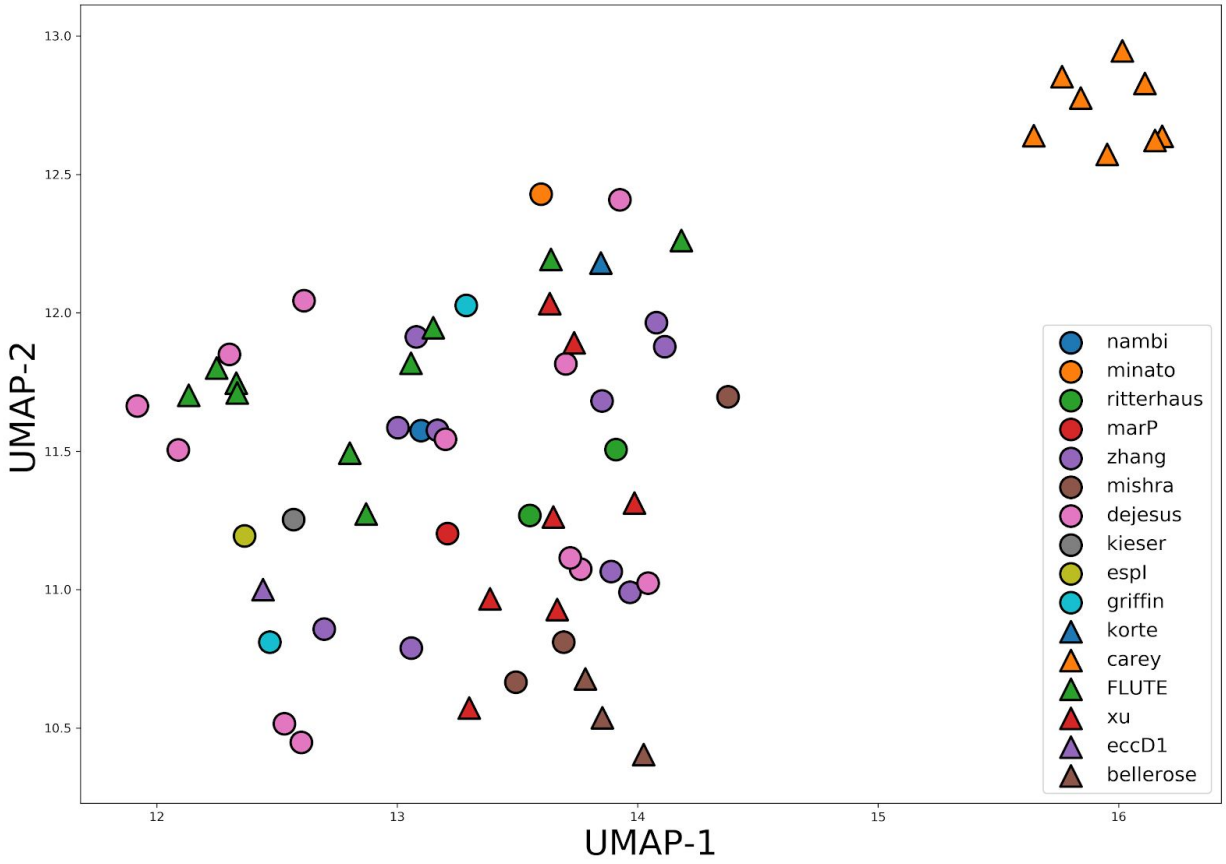

Fig S1: Unsupervised clustering of standardized TnSeq data by screen reveals a subset of screens with potential batch effects. Two-dimensional projection of TnSeq screens was performed using the

Uniform Manifold Approximation and Projection (UMAP) algorithm. Each point corresponds to a single TnSeq screen in the MtbTnDB, and screens are color-coded according to the reference publication or data-source.

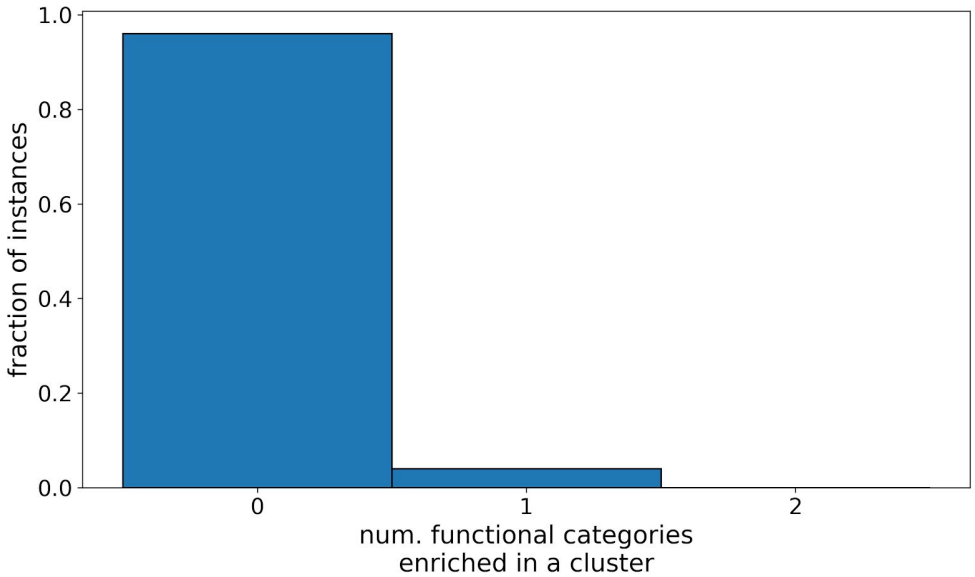

Fig S2: Lack of enrichment of gene functional categories in randomized versions of gene essentiality data. To evaluate the statistical significance of the observed enrichments of COG and Tuberculist gene functional categories in UMAP projection of TnSeq essentiality profiles, we generated 200 randomized versions of the UMAP dataset. Specifically, the positions of genes in the 2-dimensional UMAP projection were fixed, while the gene information (Rv-ID, name, annotation) were randomly shuffled. The normalized histogram shows that in more than 95% of the shuffled datasets, no enrichment of functional categories was detected in any cluster.

Table S1: Consensus of conditional essentiality calls between standardized datasets and original published datasets.

		standardized	
		conditional essential	not conditional essential
original	conditional essential	1157	1205
	not conditional essential	1292	107952