EHR-Based Phenotyping: Bulk Learning and Evaluation

Supplementary Documentation

A. Code Selection in the Bulk Learning Set

The set of infectious diseases in the experiment was selected by taking into account the hierarchical structure of ICD-9 codes and the size of the data. Specifically, the ICD-9 system is a taxonomy of diagnostic codes organized in a tree structure with the first three digits representing disease category and additional digits after the decimal point indicating specific morbidity information. In principle, any set of clinical conditions that share common clinical factors in the diagnoses and treatments can be included in a bulk learning set as long as it is feasible to identify common variables with supporting data for the clinical cases in the set. In this study, we focus on the class of infectious (and parasitic) diseases that are known to share the predictive variables associated with the four phenotypic groups – microbiology, antibiotic, blood test, and urine test – which are by no means an exhaustive list. To better understand the applicability of the bulk-learning framework in a more generic setting, we selected the ICD-9 codes in a manner that encourages a higher diversity of infection categories, outlined in Table 1 in the paper. In particular, we excluded HIV infection from our training data due to its large data volume.

On the surface, it may seem that a pure random selection from among all infectious diseases is the most unbiased choice to form the bulk learning set; nonetheless, such strategy can lead to clusters of clinically similar conditions due to the hierarchical structure of the coding system. For instance, 036.0 and 036.1 represent *meningococcal meningitis* and *meningococcal encephalitis* respectively and share pathological similarity and common pathogens.

For a greater diversity in the clinical cases, we employ a two-step random selection process based on the *hash map* where the digits prior to the decimal point serve as a *key* that references a *bucket* of full ICD-codes. The random selection then proceeds by first selecting the key followed by selecting an ICD-9 code from the corresponding bucket where both steps follow a uniform distribution. In addition, the diagnostic codes are ranked according to their associated number of unique patients. In this manner, 100 different codes are selected from the “moderately ranked” in terms of their corresponding number of unique patients. The reason behind this strategy is to sufficiently exhibit the property of a varying data size distribution that occurs in practice and to reduce the size disparity of the training data across different diseases.

B. Feature Extraction

The process of the feature extraction highly depends on the data source format and the underlying data models. In this study, patient data in the CDR are organized in a structured format such that the feature set can be assembled by first identifying the target set of patients with the related diagnostic codes (which in this study are the 100 ICD-9 codes associated with infectious diseases). Once the relevant diagnostic entries are determined, we then cross-reference these entries with other tables containing the patient attributes (i.e. codified variables consistent with MED) associated with the target clinical concepts by matching the patient identifier (e.g. medical record number, MRN) and dates (e.g. admission dates and discharge dates).

Assuming that the MRNs are unique, in the simplest case, one may consider using both MRN and date attributes as the composite foreign key to uniquely identify the patients in every table. However, in a typical healthcare process, a clinical visit involves a date range, i.e. admission and discharge, and mostly do not coincide with the dates associated with laboratory tests and medical prescriptions if involved as part of the diagnostic and treatment procedure. To increase the flexibility of the matching process, we use a predefined threshold of error to account for the potential gap between the dates when diagnostic codes were given and those of the relevant clinical procedures. For instance, a patient could receive diagnostic tests prior to a clinical visit or after the visit as a follow-up procedure; on the other hand, medicinal treatments are typically given following the visit. In this study, we looked for the clinical records that fall within the range of 60 days prior to the mention of a target ICD-9 code and 30 days following the mention. The tolerable errors in times shall be considered as adjustable parameters.

C. Control Data

As mentioned in the paper, control data are the negative examples for the predictive unit, formulated as a binary classifier, created to serve as counterexamples for a target disease. For clarity, we denote the bulk learning set by $C\colon\{C_i \mid i = 1{\sim}N\}$ where C_i denotes the cohort of the *i-th* infectious disease (expressed as an ICD-9 code) and N is the total number of diseases in the bulk learning set (N =100 in this study). It is convenient to overload the notation in the context of statistical learning such that C_i also refers to the training data set associated with *i-th* infectious disease. Given this, one can simply denote C_{-i} as the set of all cases excluding those in C_i, i.e. patients without the *i-th* infectious condition. Further, let C' denote the set of clinical cases involving no infections. Thus, the control data for C_i would potentially consist of the cohort sampled from C_{-i} and C'. Intuitively, for a given cohort C_i, using the sampled data from C_{-i} as the control group can potentially identify the key clinical differences in the bulk learning set while using the cohort C' as the control helps to distinguish infections from other broader classes of clinical conditions.

Without judiciously selecting the control data, the chance of finding clinically meaningful predictors for a target disease may be greatly reduced. Recall from Section 2.2.4 that share variables are defined as the predictor variables that not only occur in the training instances of both class labels but also assume non-trivial values, where the non-triviality depends on the domain of the variable and its meanings. A training data set with few shared variables tends to result in overly simplistic models in which strong predictors (e.g. large coefficients in absolute values in logistic classifier) are predominantly those with non-trivial values exclusively in positive or in negative training instances. This situation occurs when cases and non-cases are too dissimilar in the sense of sharing a very small or even an empty set of active variables. For instance, mixing clinical cases from a non-infectious disease cohort in the control data is very likely to result in the lack of shared variables since these cases may not share similar treatments and laboratory results as those diagnosed with infectious conditions (e.g. patients without bacterial infections are unlikely to have received antibiotic treatments).

Similarly, care must also be taken in selecting the control group from any non-target infectious diseases. For instance, cases involving septicemia, a serious bloodstream infection, are very likely to share some commonality in blood tests as those in other bacterial infections that also tend to spread through the bloodstream. The caveat is that during the diagnosis of a septicemia case, other testing such as urine culture could also be involved in order to either evaluate the source of the original infection or to rule out potential sources. Hypothetically, if a clinical variable representing the urine culture order is active in a significant portion of the positive examples for septicemia whereas only few or none of the cases mixed in the control, or negative examples, are active in this variable, then inevitably, urine culture will appear to be one of the strong predictors for septicemia, even though on the surface, variables as such may not seem to be directly relevant to the case. In general, the diagnostic tests used for ruling out tend to be strong predictors when they are active with high probability predominantly in one condition but not the others, which happen to be chosen as the control data.

We now show a concrete numerical example. Consider a subset of training data in which 6 clinical variables $\{x_i \mid i = 1\sim 6\}$ are used to predict a given disease. We can then consider a hypothetical data set that consists of 6 training instances encoded by a 6-by-6 design matrix X and its corresponding labels encoded by **y**, a 6-by-1 column vector, where each row in X represents a training instance and the first 3 rows correspond to positive cases and last 3 rows are negative:

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Notice that only the 2nd and 3rd variable (i.e. x_2, x_3) has a non-zero values (1 in this simple example) that occur in both positive and negative examples while all the other variables (i.e. x_1, x_4, x_5) only assume non-zeros exclusively in either positive or negative examples. For convenience, we refer to the variable set like $\{x_2, x_3\}$ as shared variables since there exist, for both class labels, at least one or more training instances with "non-trivial values" for these variables. Note that the non-triviality certainly depends on the domain of the variable and the meaning of its allowable values. The assumption made in this example is simply that a value of 0 signifies a non-event (and hence trivial) whereas 1 signifies an event, in which case the variable is said to be non-trivial, or active, for the given training instance. Indeed, if one were to fit a parametric model such as a logistic classifier to this dataset, then the shared variables x_2 and x_3 would have relatively smaller coefficients (in absolute values). Intuitively, this can be concluded by observing that the variables in $\{x_1, x_4, x_5\}$ are active only in either positive or negative examples but not both and therefore, knowing the values of these variables provide significant information as for which label the corresponding training instances belong to; by contrast, knowing the values of the share variables still leaves much ambiguity to be resolved given that they are active in both labels of training instances. If we denote the coefficients for the variable set in terms of a weight vector **w**, then fitting the data above directly via an $\ell 2$-regularized logistic classifier (with $\ell 2$ penalty strength set to 1.0) results in coefficient/weight vector **w** approximately at (0.41, 0.40, 0.07, -0.07, -0.40, -0.41), where the coefficients $\{w_2, w_3\}$ for shared variables $\{x_2, x_3\}$ have relatively lower absolute values as expected.

Indeed, there is more to the interpretation of coefficients for the logistic classifier; however, this simple example suffices to demonstrate that the lack of shared variables can lead to an overly simplistic model in which strong predictors (i.e. those with high absolute coefficients) are predominantly those with non-trivial values exclusively in positive or in negative training instances. This situation can occur when the case and the non-case are too dissimilar in the sense that the case has a very different active variable set from that of the non-case. Using the data from the cohort C' (i.e. cases without infections) as the control is very likely to result in the lack of shared active variables since the set C' may not contain patients with matching treatments and laboratory results as those in C (e.g. patients without infections are unlikely to receive antibiotic prescriptions).

Similarly, care must also be taken in selecting the control group from within C_{-i} for a given cohort of positive cases C_i**.** For instance, cases involving septicemia, a serious bloodstream infection, are very likely to share some commonality in blood tests as those in other bacterial infections that also tend to spread through the bloodstream. The caveat is that during the diagnosis of a septicemia case, other testing such as urine culture could also be involved in order to either evaluate the source of the original infection or to rule out potential sources. Then, hypothetically if a clinical variable representing the urine culture order is active in a significant portion of the positive examples for septicemia whereas only few or none of the cases in the negative examples are active in this variable, then inevitably, urine culture will appear to be one of the strong predictors for septicemia, for reasons illustrated in the aforementioned example, even though on the surface, it may not seem to be directly relevant to the case. In general, the diagnostic tests used for ruling out tend to be strong predictors when they are active with high probability predominantly in one condition but not the others, which happen to be chosen as the control data. Note that although we have focused on binary explanatory variables so far, similar reasoning can be generalized to continuous or ordinal variables.

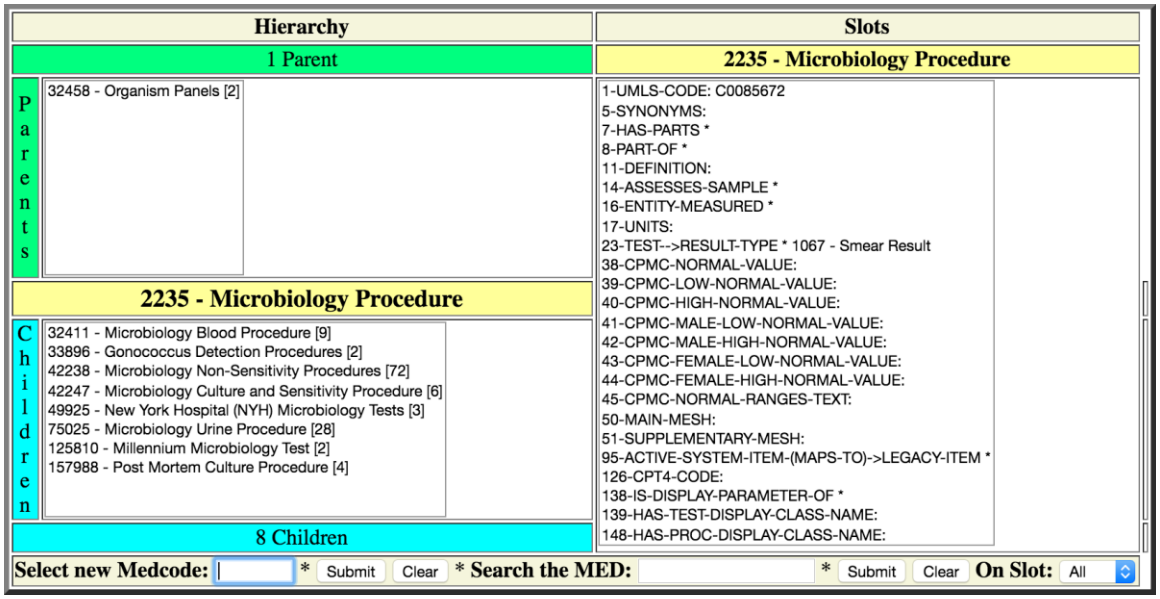
In order to match the positive and negative cases in a manner that maximizes the number of the shared variables while considering all the diseases in the bulk learning set, a systematic matching strategy is perhaps more preferable than an ad-hoc disease-dependent strategy. The idea of mixing appropriate control data to reach the largest set of shared variables can be recast as a problem of finding the greatest similarity in terms of the Jaccard coefficient, which is useful in measuring the degree of overlap between two (active) variable sets. Specifically, let F_i and F_j denote the sets of clinical variables associated any two different diagnostic codes ($i \neq j$), then their Jaccard coefficient can be expressed as:

$$J(F_i, F_j) = \frac{|F_i \cap F_j|}{|F_i \cup F_j|},$$

which is simply the ratio of the size of the intersection and the size of the union of the variable sets. Since the bulk learning set in this study consists of 100 diagnostic codes, the index ranges from 1 to 100 (i.e. we consider $\{F_i \mid i = 1 \sim 100\}$). In considering maximizing the number of shared variables, we are only concerned with the active variables (out of the entire variable set). As mentioned in the paper, an explanatory variable is said to be active if and only if its value is non-trivial, which in this study is assumed to be any non-zero values. Since a training data can consist of an arbitrary number of cases, it is useful to define the frequency of active occurrences – the number of times that a variable assumes a non-zero value in the training set. Given this, we first rank the frequency of active occurrences for all variables in a training set, and subsequently take the top 80% most frequent active variables, which are then used to define $\{F_i\}$. With all frequent active sets determined, we now have a basis to compare the similarity of any two training data sets by computing *J*. In particular, given a set of positive examples from C_i (associated with the *i-th* diagnostic code), we choose the top *N* most similar sets of "non-cases" from C_{-i} as the control data, where *N* was chosen to be 3 for the experiments in this study.

In the paper, we mentioned that the training data from both class labels are compared on the basis of the global feature set. Alternatively, a more accurate matching process would involve matching case by case, with one from C_i and the other from C_{-i}, whereby a top-ranked non-cases are obtained for each corresponding case; however, a brute-force procedure like this can be computationally prohibitive when the sizes of the training sets are large.

Moreover, the ratio between the positive and negative cases is another important factor that drives the prediction accuracy and stability. In particular, the control data obtained from the similarity-based criteria



32458 - Organism Panels [2]

32411 - Microbiology Blood Procedure [9]
33896 - Gonococcus Detection Procedures [2]
42238 - Microbiology Non-Sensitivity Procedures [72]
42247 - Microbiology Culture and Sensitivity Procedure [6]
49925 - New York Hospital (NYH) Microbiology Tests [3]
75025 - Microbiology Urine Procedure [28]
125810 - Millennium Microbiology Test [2]
157988 - Post Mortem Culture Procedure [4]

1-UMLS-CODE: C0085672
5-SYNONYMS:
7-HAS-PARTS *
8-PART-OF *
11-DEFINITION:
14-ASSESSES-SAMPLE *
16-ENTITY-MEASURED *
17-UNITS:
23-TEST-->RESULT-TYPE * 1067 - Smear Result
38-CPMC-NORMAL-VALUE:
39-CPMC-LOW-NORMAL-VALUE:
40-CPMC-HIGH-NORMAL-VALUE:
41-CPMC-MALE-LOW-NORMAL-VALUE:
42-CPMC-MALE-HIGH-NORMAL-VALUE:
43-CPMC-FEMALE-LOW-NORMAL-VALUE:
44-CPMC-FEMALE-HIGH-NORMAL-VALUE:
45-CPMC-NORMAL-RANGES-TEXT:
50-MAIN-MESH:
51-SUPPLEMENTARY-MESH:
95-ACTIVE-SYSTEM-ITEM-(MAPS-TO)->LEGACY-ITEM *
126-CPT4-CODE:
138-IS-DISPLAY-PARAMETER-OF *
139-HAS-TEST-DISPLAY-CLASS-NAME:
148-HAS-PROC-DISPLAY-CLASS-NAME:

Fig. S1. A concept node in MED, microbiology procedure (2235), one of the concept seeds for the microbiology group.

on active variables are inherently larger than the case data. We sample a subset of the negative examples to match comparably with the positives to obtain a balanced dataset. Situations may occur, however, when the positive examples are too few, making it difficult to enforce a balance in class labels. In such cases, existing methods range from generating synthetic samples and systematic oversampling for the minority class to systematic undersampling of the majority class, using unequal class weights, and using anomaly detection algorithms, among others, for which we refer the readers to [1–3] for more details.

D. Medical Entities Dictionary

In order to obtain features associated with various clinical concepts, we use Medical Entities Dictionary (MED) developed at New York Presbyterian Hospital with built-in ontological structures that organize various medical concepts/terms in the form of concept hierarchies. Specifically, MED is a semantic network [4] with medical concepts drawn from various sources such as UMLS [5], LOINC [6], and ICD-9-CM. Concept hierarchies are structured in directed acyclic graph (DAG) where each concept node can have multiple parents. Each concept node has an assigned code as an identifier, or MED code, with attributes such as the name, coding counterparts from various systems and textural information. Fig. S1 is a snapshot of MED network at the node of microbiology procedure (2235) with 8 children out of which, microbiology blood procedure and its descendants, for instance, can serve as a candidate concept node for feature grouping if the bulk learning set contains clinical cases of bloodstream infections such as sepsis.

E. Clinical Feature Overlaps

In accessing the clinical similarity from within the bulk learning set comprising multiple diseases, we examine their common clinical attributes characterized by shared (active) variables (see Section 2.2.4). That is, the larger the intersection of active variables between any pair of conditions, the more clinically similar they are to each other in the context of a phenotypic model. For instance, Ceftriaxone can used to treat multiple bacterial infections at different sites of the body including bloodstream, lungs, and urinary tract, etc. and therefore, its corresponding clinical variable would be active among those disease cohorts that share the same antibiotic treatment.

Moreover, since the size of the bulk learning set can be indefinitely large (perhaps in the order of hundreds), it is useful to inspect higher-order intersections across multiple diseases in parallel. Visualizing set intersections can be exponentially challenging, however, even when the number of sets becomes only trivially large due to the combinatory explosion of all possible set combinations. With 100 different diseases in the bulk learning set, conventional set visualization methods such as Euler and Venn diagrams become infeasible. To address this, we use the set visualization tool UpSet, developed by the Visual Computing Group at Harvard University [7], to understand the degree to which the active variables overlap across various diseases. Fig. S2a and Fig. S2b illustrate, in descending order, the degrees of intersections for the shared variables in the antibiotic model across 10 diseases identified by their ICD-9 codes. In particular, diseases in Fig. S2a were chosen to be relatively diverse according to the ICD-9 coding; by contrast, the diseases in Fig S2b are relatively similar.

Each diagram centers around the matrix/grid represented by dots and connecting lines combined as set indicators. The rows of the matrix correspond to the sets, which in our case represent the subset of the ICD-9 codes in the bulk training set, while the columns correspond to intersections. In particular, recall from Section 2 that each ICD-9 code has an associated classifier and a corresponding feature representation comprising the most frequent active shared variables that characterize the global property of the overall training data within the context of the same phenotypic model (Section 2.3.2). Within the matrix, a filled dot represents a participating set in the intersection and a vertical line connecting the dots serves to highlight the sets involved in the intersection. The cardinality of a set is encoded by the length of the bars: the vertical bars on top of the matrix indicate the sizes of the shared (active) variables of the participating sets whereas the horizontal bars off to the left indicate the total sizes of the active variables associated with the sets themselves.

Given these notations above, the histogram above the matrix essentially represents the sorted degrees of the intersecting active variables exclusive to and shared by the corresponding subset of the ICD-9 codes, marked by the filled dots. For instance, the first column in Fig. S2a indicates that the 10 selected diseases

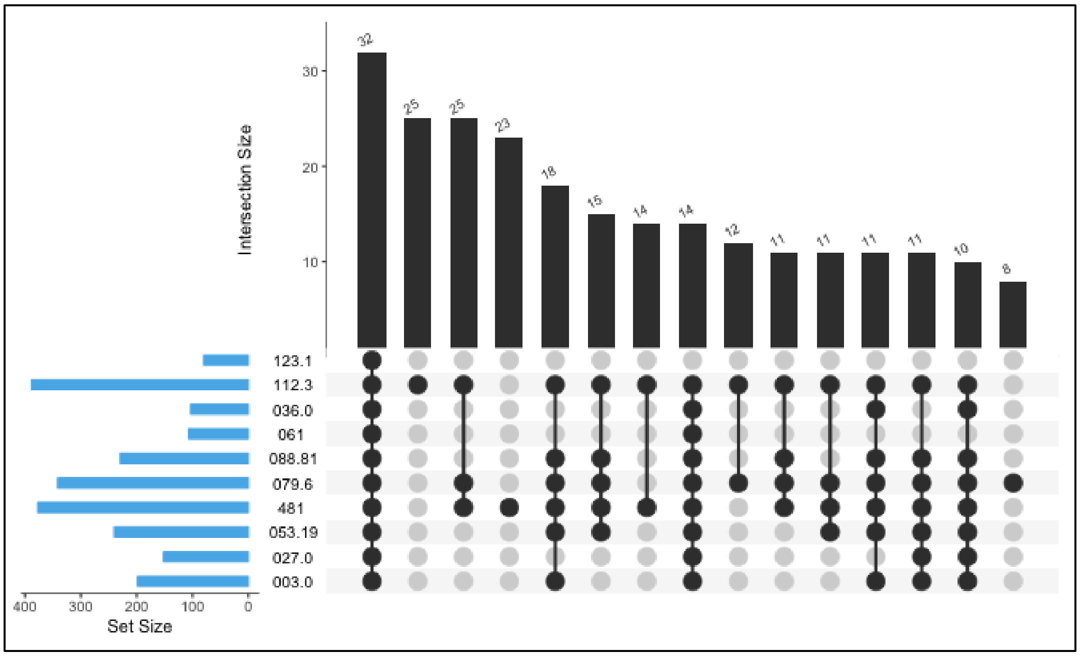


Fig. S2a. Active variables associated with the antibiotic group analyzed in terms of the degree of intersections, sorted in descending order, for the selected 10 diverse infectious diseases on the vertical axis: cysticercosis (123.1), candidiasis (112.3), meningococcal meningitis (036.0), dengue (061), Lyme disease (088.81), RSV (079.6), pneumococcal pneumonia (481), herpes zoster (053.19), listeriosis (027.0), salmonella gastroenteritis (003.0).

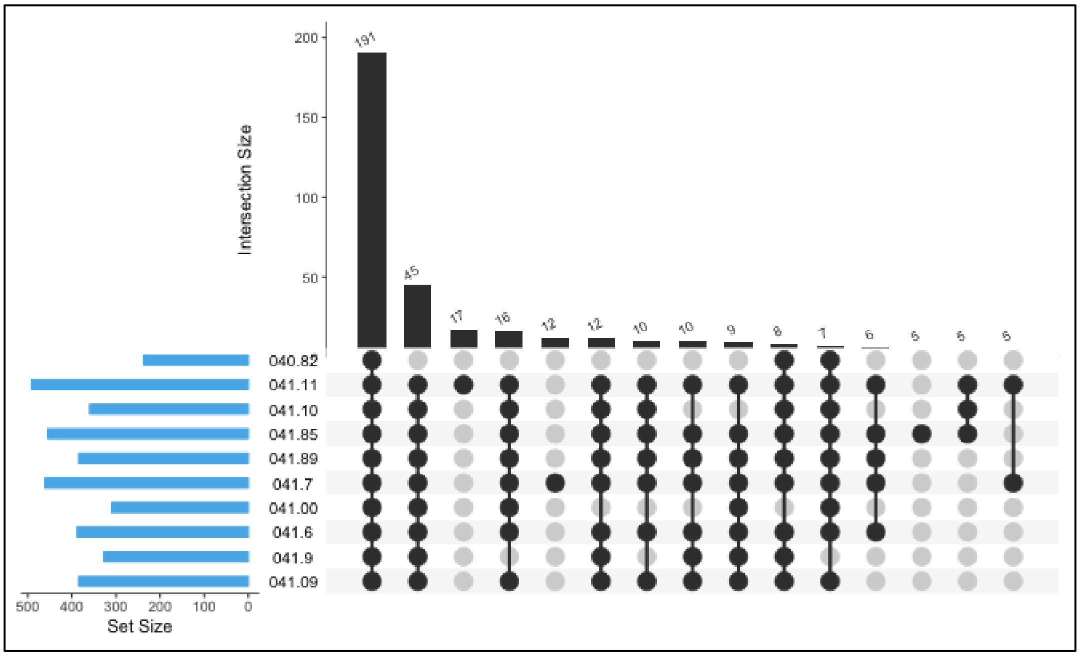


Fig. S2b. Active variables associated with the antibiotic group analyzed in terms of the degree of intersections, sorted in descending order, for the selected 10 similar infectious diseases according to the ICD-9 classification: toxic shock syndrome (040.82), staphylococcus infection of unspecified site (041.11, 041.10), gram-negative organism infection (041.85), unspecified bacterial infection (041.89, 041.9), pseudomonas infection of unspecified site (041.7), unspecified streptococcus infection (041.00), proteus (041.6), streptococcus infection of unspecified site (041.09).

have 32 active variables in common. By contrast, the second column has only 1 filled dot, which indicates that there exist 25 active variables exclusive to candidiasis (ICD-9 code 112.3); similarly, the third column shows another 25 active variables but shared by 3 diseases instead: Candidiasis (112.3), respiratory syncytial virus (079.6), and pneumococcal pneumonia (481), respectively. By comparing Fig. S2a and S2b, we see that similar diseases share higher degrees of intersecting variables. In particular, the first column in Fig. S2b indicates that there exist 191 active variables shared by all the diseases, a significant portion of the total variable sets. Additionally, most diseases except for toxic shock syndrome (040.82) share another 45 active variables, which perhaps is not surprising considering the fact that 040.82 is the only ICD-9 code that does not start with 041.

F. Ontological Feature Grouping

Knowledge representation of the clinical cases plays a decisive role in how the share phenotypic components of the bulk learning set can be effectively modeled. Decomposing, via medical ontology such as MED, the potentially gigantic EHR feature set into several coherent groups admits an initial feature (set) reduction such that multiple phenotypic models can be trained in parallel.

Specifically, clinical variables are grouped according to a set of desired clinical concepts, guided by MED, that are considered discriminatory for the diseases in the bulk learning set. For example, the microbiology feature group in this paper is derived from 3 concept seeds, as specified in Table 3, including microbiology procedure (MED code 2235), microbiology results (315), and microbiology sensitivity (41901). The MED codes derived from these seeds are then taken union to form a reference set, which effectively delineates the scope of candidate phenotypic features in the domain of microbiology. Finally, only the MED codes with supporting data in the CDR are retained as the final feature set for the microbiology group. The feature sets for the other three phenotypic groups – antibiotics, blood tests, urine tests – can be determined in a similar way summarized as follows:

1. Given a phenotypic group, select its appropriate concept seeds.
2. For each concept seed in a group, traverse its corresponding subgraph in the MED and collect the MED code for each concept node encountered; repeat the same graph traversal for each seed to obtain its associated MED codes and subsequently, take the union of all the MED codes from the entire concept seeds to get the reference set of the phenotypic group.
3. Take the intersection of the reference set obtained from step 2, and the codified patient attributes in the CDR, assumed to have mapped to MED codes prior to this routine, to finally arrive at the feature set associated with the phenotypic group for training statistical models later on.
4. Repeat steps 1 to 3 for each phenotypic group to obtain their associated feature sets.

Additionally, for each phenotypic group, we retain only the top 80% most frequently active variables (see Section 2.2.4) across the entire training data of the bulk learning set such that all diseases modeled by the same phenotypic group share the same feature set.

G. Design of Phenotypic Groups

In the paper, we have assumed that by locating concept seeds at appropriate levels, one can then obtain the desired reference set via graph traversals. Defining appropriate levels for the concept seeding, however, is somewhat of a design choice. A simple guideline for tracing the concept hierarchy is that medical concepts located towards the higher level of the hierarchy are more generic whereas downstream concepts become progressively more specialized. If concept seeds are too generic, the resulting reference set may end up covering too many features, which though conceptually related, may not provide a good basis for the purpose of feature decomposition since the corresponding feature set, after matching with the data source, can still be very large (relative to the number of training instances). For instance, the concept node, organism panels (32458), which is connected to two child nodes: microbiology procedure (2235) and urinalysis (33891), would be too generic as a seed to distinguish the contributing factors between microorganism tests and urine analyses in disease predictions. On the other hand, if a concept seed is too specialized, then the resulting set of MED codes can be either too small or having no supporting data at all in the CDR; however, this is less of an issue because one can simply use more than one concept seeds as a grouping unit as mentioned earlier.

In this study, we chose the phenotypic groups that are believed to exhibit predictive strength in the domain of infectious diseases. In particular, the microbiology group consists of features associated with microorganism tests of different sources (e.g. blood, urine samples, etc.), and pathogens (e.g. bacterial, fungal cultures, etc.). The microbiology group, as a concept unit, is useful in predicting infectious diseases as long as there exists at least some strong predictors within the group. For instance, a microorganism test for Streptococcus Pneumoniae (750) is expected to be a strong predictor for clinical cases of Streptococcus Pneumonia. Various antimicrobial susceptibility tests also fall under the microbiology group such as the test for beta-lactam antibiotics (36450), which determines potential drug resistance of common pathogens to this family of antibiotics and thus provides a signal to the bulk-learning model the likelihood of observing related antibiotic prescriptions as treatments. Other features in microbiology group include status information such as an indicator for heavily contaminated culture (426).

The antibiotic group, on the other hand, includes various antibiotic prescription drugs such as ceftriaxone (30475), commonly used to treat various bacterial infections included in the bulk learning set such as pneumonia, meningitis, and skin infections. Conceptually, the antibiotic group is closely related to the microbiology group, given that antibiotics are treatments for the infection of bacterial microorganisms; therefore, the features in antibiotic group can be significantly correlated to at least a subset of features in microbiology group such as those representing antimicrobial susceptibility tests. Consequently, the contributing factors for disease predictions between prescriptions and susceptibility tests could become obscured if these features were to mix together in a single phenotypic group. However, since specific drug prescriptions and susceptibility tests are now decoupled into two separate phenotypic groups, the two sets of variables will be trained independently via different base models such that their contributions to the disease prediction can be observed independently.

The intravenous chemistry group includes hematology results and intravenous chemistry tests. Hematology results include various blood level indicators such as those for white blood cells; for instance, an increase in lymphocyte concentration (with mentions of 42936 in the data) is usually a sign for viral infection. Intravenous chemistry tests identify the numerous chemical substances found in the blood such as glucose, creatinine, and uric acid. Taking urine sample is also a common diagnostic procedure for infectious diseases. The urinary chemistry group includes urine panels and urine chemistry tests. Kidney infection, for instance, is commonly diagnosed via urinalysis (33891), which is a child node of urine panels. In particular, a mention of urinalysis coupled with intravenous tests on creatinine level and positive microorganism culture results on leukocyte esterase, can be a strong indicator for kidney infection.

The phenotypic groups mentioned above are by no means an exhaustive list. Various body substance specimens, for instance, can be used to indicate specific sites and tissues of the body from which microorganism smears and cultures are obtained (although specimens as a concept is likely to be subsumed by culture results from the microbiology group). Antifungal and antiviral prescriptions are also examples of potential concept classes that can be used to define phenotypic groups for predicting fungal and viral infectious diseases respectively.

H. Feature Types

Features used for building statistical models come from the patient attributes. Corresponding to a MED code, each patient attribute in the CDR can assume a numerical value (e.g. laboratory measurements), a textual content (e.g. clinical notes) and at times assume no values when it corresponds to a MED code representing a state (e.g. a positive or negative culture result) or an indicator (e.g. confirmation for a laboratory order), in which case the value is implicit in the meaning of the MED code. For the purpose of this study, features are treated as either binary or numerical. Numerical features can assume any discrete or continuous values depending on the nature of corresponding MED codes. For instance, the code 36283 (aka

CPMC Laboratory Test: Urine pH) takes on a continuous value ranging from 5.5 to 8.5. MED codes that do not have values are considered as binary features and their mentions in the CDR simply signal the presence or absence of clinical events (e.g. a positive culture result). For simplicity, MED codes associated with textural data are considered as binary features as well by ignoring their textual contents. Although doing so results in a loss of information, however, for the purpose of illustrating the key concepts of the bulk-learning framework, we adopt this scheme as an approximation for modeling the disease assuming that each phenotypic group covers a sufficient range of clinical variables as predictors such that the prediction will still be reasonably accurate.

I. Feature Selection based on LASSO

In this study, we use $\ell 1$-regularized logistic regression to determine the best subset of features within each phenotypic group using AUC as an evaluation metric. For convenience of discussion, phenotypic groups are herein denoted by the set Φ: $\{\Phi_i \mid i = 1{\sim}4\}$. Regularization is widely used as a method to cope with overfitting, which often occurs when the training set is relatively small compared to the number of features. In particular, the $\ell 1$-norm regularization removes irrelevant features by shrinking their parameters to zeros and, when applied in logistic regression, has been shown to have good generalization performance [8–10]. The property of parameter shrinkage makes it a good option to address the fact that certain infectious diseases tend to have relatively fewer data points and therefore, a large feature set is undesirable in the predictive analytics for these diseases; by contrast, the phenotypic group comprising the total feature set remains invariant to the data.

In practice, the size of a phenotypic group (denoted by $|\Phi_i|$, or $|\Phi|$ for simplicity without referring to a specific group) can potentially change over time as they depend on the concept seeding as well as the medical ontology, which is subject to updates in response to the advancement of healthcare procedures. To ensure an upper bound on $|\Phi|$, we introduce a tunable parameter ρ such that the total number of selected features can be kept under control; that is, we want $|\Phi| \leq \rho$. For the sake of clarity, we first introduce a few useful notations. Consider for a given infectious disease with a training set of size M: $\left\{\left(\mathbf{x}^{(i)}, y^{(i)} \mid i = 1 \ldots M\right)\right\}$ where $\mathbf{x}^{(i)}$ represents the i-th feature vector associated with a phenotypic group Φ while $y^{(i)}$ is the corresponding binary label. The optimization of an $\ell 1$-regularized logistic regression problem can then be expressed as follows:

$$\min_{\theta} \sum_{i=1}^{M} -\log P\left(y^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}\right) + C\|\mathbf{w}\|_1,$$

where $P\left(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}\right)=\frac{1}{1+\exp\left(-\mathbf{w}^T\mathbf{x}^{(i)}\right)}$; i.e. a sigmoid function of $\mathbf{w}^T\mathbf{x}^{(i)}$, a linear combination of features with parameters $\mathbf{w}$.

Inside the minimization operator is essentially a cost function that consists of negative log likelihood and $\ell1$ penalty ($\mathrm{C}\|\mathbf{w}\|_1$). Observe that as the regularization strength *C* becomes higher, the $\ell1$ penalty term becomes more significant; therefore, in order to minimize the cost function, a larger set of the parameters has to decay to zeros, leading to a smaller feature set. The optimal *C* can be determined via cross validation. Note that ρ is only used to enforce the upper bound on $|\Phi|$, meaning that ρ has no effect if the logistic classifier results in a $|\Phi| \leq \rho$. Therefore, a very large ρ simply honors the feature selection result from the $\ell1$ regularization. Conversely, if ρ is too low, some of the informative features may be lost.

Moreover, for better stability and reliability in the result of feature selection, a resampling process is embedded into the $\ell1$-regularization; this version of Lasso is also known as Bolasso [11]. The resampling process refers to using sampling with replacements to induce multiple bootstrap samples out of the same training set such that each sample approximately follows the same data distribution and variability. When resampling is used as an ensemble learning approach in supervised learning, it is often referred to as bagging and has been shown to effectively reduce variance and improve accuracy. For our purpose here, the resampling step creates multiple bootstrap samples (one sample per iteration in a loop, algorithmically speaking) from the training set as inputs to the logistic classifier, which in turn selects the best feature subsets leading up to maximized AUCs. In the event where the resulting feature set becomes too large (i.e. $|\Phi| > \rho$), then only the most frequently selected features are kept. Conversely, if the resulting feature set is too small, then *C* can be chosen to increase incrementally (e.g. by 10-fold at a time) until the desired size of the feature set is reached (but no greater than ρ), the tradeoff of which is acquiring potentially redundant features.

J. Data Fusion

The process of data fusion for transforming the level-0 data to level-1 data is not unique and with different assumptions, numerous other strategies are possible in addition to the simpler one based on maximum and minimum probability predictions mentioned in the paper. For instance, instead of using the probability prediction of the base model as a conduit for choosing the most representative training instance from multiple clinical visits, one may wish to find temporally agreeable training instances across models. However, a clinical case may not always agree in the timestamp across different phenotypic models, each of which corresponds to different clinical activities during the diagnoses and treatments of a disease. As a result, it is not always straightforward to match one clinical case across the phenotypic models due to irregular time gaps between clinical activities.

One solution for matching timestamps is to fit a time-weighted regression model such that the temporal locality can be factored in the process of matching training instances for the same clinical case across different phenotypic models: the closer the time point, the more likely for the training instances to have come from the same clinical visit and hence, a higher weight would be given to its associated predictive probability for the disease. Another tunable parameter in the formation of the level-1 data is the sample weight. Having no evidence in a phenotypic model indicative of a negative case for a disease (i.e. (i.e. no evidence of ruling out a disease) may not provide a strong signal for determining the likelihood of having that disease given the enormous number of possible diseases. By contrast, having no evidence suggesting a positive case can be a strong signal in dismissing a positive case depending on the phenotypic model in action and its correlation with the target disease. For instance, diseases with pathological links to bloodstream infections would be highly correlated to the blood test model and thus, missing supporting evidence in blood tests would be a strong signal for ruling out. The key idea behind the level-1 feature representation is to leverage the diversity of probabilistic predictions from different base models; i.e. the more base models lack supporting data, the less reliable the combined prediction would become. The sample weight can therefore take into account of the number of models without supporting data. We refer interested readers to [12] for more information on sample-weighted regression and its implications.

K. More on Abstraction Feature Representation

Abstract features can be constructed via methods in relation to dimensionality reduction such as PCA, which essentially re-expresses the original feature set, potentially correlated, through their linear combinations as principle components, the first few of which account for most of the data variance, assuming that at least a subset of features are indeed (linearly) correlated. One can then project the original data points to these principle axes to obtain a new set of coordinates, giving rise to a new data representation, where principle components are a form of abstract features as they are functions of raw features. The notion of deriving abstract feature set via learning appropriate function mappings of raw features can also be traced from the literature of representational learning where the main objective is to capture useful explanatory factors inherent in the data, which not only minimizes efforts in feature engineering but the learned (abstract) feature representations can further assist in supervised learning tasks such as disease classifications in computational phenotyping. At times, through representation learning, dimensionality reduction is also achieved as a secondary result, which can be observed, for instance, in a typical deep learning architecture, where the number of hidden units is progressively made smaller as one goes further away from the input layer.

As mentioned in the paper, the autoencoder, as an instrument for unsupervised feature learning, has been shown to capture hidden temporal patterns in the longitudinal data of gout and acute leukemia [13]. In particular, these hidden temporal patterns reveal what underlies the time trajectories of uric acid concentration (e.g. measurement sequences in multiple 30-day segments) such as upward, downward ramps, and oscillatory patterns (i.e. Fourier components). These learned (abstract) features suggest that relative to the clinical cases of gout, leukemia cases tend to assume relatively wider variations, lower values and higher frequency in the measurements of uric acid. These temporal features are essentially transformations of the input feature vector (i.e. uric acid measurements) through applying non-linear functions (e.g. sigmoid) to the weighted sums of feature vector components, structurally similar to the internals of a logistic classifier. The property of autoencoders allows for a deep architecture to be constructed by stacking multiple layers of autoencoders, leading to higher-levels of feature abstractions via multiple stages of (non-linear) transformations.

Incidentally, model stacking bears resemblance to (stacked) autoencoders and other deep learning architectures in the sense that one could choose to cascade multiple layers of (meta-)models to achieve levels of feature abstractions (also known as cascade generalization [14]). As discussed in the paper, using the abstract feature representation on top of the phenotypic models helps to gain a better control over the dimensionality of the training data given its compact form through functional mappings of raw features. If the abstract feature space has sufficient discriminatory strength, then the requirement of annotated data can be greatly reduced due to its lower dimensionality. Specifically, the number of training instances needed to learn well from the data (i.e. sample complexity) grows with the VC dimension [15]. For most parametric models, the VC dimension grows linearly in the number of (free) parameters, which tends to increase the feature set becomes larger.

Although we have been focusing on the four example phenotypic models in the paper, it is possible to further extend the abstract feature set by introducing more phenotypic models (e.g. antiviral prescriptions) to account for more clinical aspects of infectious diseases. The control of feature dimensionality, in particular, can be achieved by varying the stacking architecture. The example stacking hierarchy depicted in Fig. 1 in the paper effectively has only a single level of combiners; i.e. from base models to level-1 meta-models for per disease. Alternatively, each base model can be formulated, instead, by various feature subsets of the chosen phenotypic groups, leading to a finer-grained feature abstraction with a higher flexibility in terms of the number of parameters for promoting diversity tradeoffs in model predictions. Moreover, it is also possible to mix abstract features with features at lower levels, as a form of partial abstraction, for potentially more accurate disease predictions at the cost of an increasing demand of labeled data due to a higher “mix-in” of raw features.

Further, the feature abstraction from one level to another can be made progressively by introducing intermediate meta-classifiers as model combiners, for which, the single-combiner model in this study actually results in a very sharp reduction of feature dimension. In practice, this sharp and perhaps abrupt reduction of feature dimension may not be necessary, if the size of labeled data is at least moderate, but nonetheless was chosen to be the stacking architecture that suffices for illustrating the key concepts involved in bulk learning. Generally speaking, the degree of feature abstraction depends on the available labeled data; the more labeled data become available, the less degree of feature abstraction is needed for the model to perform well (which refers to the property that the learned model generalizes well to unseen data). On the other hand, if labeled data were scarce due to the cost, then a higher-level abstraction would be more desirable. An example for such modeling choices in relation to "regularizing" the feature dimension is given in the paper in terms of the comparisons between the level-1 models and their level-2 counterparts.

In essence, feature abstraction used in bulk learning compresses the information coming from the base level, encoded by variables extracted directly from EHR, to a smaller and manageable set of variables. It is a form of lossy compression and hence, there is no guarantee that using the abstract features alone can always precisely distinguish all infectious diseases. This is the reason behind the need of evaluating the learned abstract feature representation (Section 3.2.2) from the perspective of how they can be used to reconstruct the original data through their use in statistical models that predict ICD-9 labels. In particular, by plotting the performance profiles in the form of Fig. 4a and Fig. 4b (the former for the global model and the latter for local models), one can then sort out which bulk learning subset are better candidates for phenotyping as a group.

L. Using Abstract Features to Predict Gold Standard

Continuing on the discussion in Section K, if the learned abstract features can reasonably differentiate individual diseases in the bulk learning set, then it is possible to move forward and use them to approximate the annotated sample with true labels that are assumed to be quite close to the surrogate labels like ICD-9. Since the true labels are independent from the surrogates (again by assumption), it is even possible to mix the learned abstract features and surrogate labels in statistical models as exemplified by the ICD-9 modulated level-1 models. The idea is to leverage their trade-offs to potentially induce a better generalization over the data than if one were to use surrogate labels alone, particularly for the data region where surrogate labels committed errors (referred to as type-FP and type-FN annotations in the paper). However, the question now becomes whether it is sufficient to examine only one type of surrogate labels.

The model performance using ICD-9-derived abstract features is comparable to the ICD-9. However, no significant improvement over the ICD-9 labeling itself has been observed using various abstract feature representations coupled with the ICD-9 as a feature (previously used as surrogate labels in deriving abstract features) in predicting the annotated sample. This is also true for higher-order models that allow abstract features to interact (see Section R). Nonetheless, training statistical models in the abstract feature space has an advantage of reducing the training requirement on labeled data.

ICD-9 is merely one form of surrogate labels among many others. We believe bulk learning has the potential to surpass the ICD-9's accuracy by taking into account multiple types of surrogate labels such as other diagnostic codes, if available, and keywords extracted from clinical notes. A practical extension would be to apply the same bulk learning method, again based on the model stacking methodology discussed in the paper, to multiple types of surrogate labels in parallel, from which their corresponding abstract feature sets can be derived and combined through another layer of model stacking that leverages their predictive diversity so as to go beyond the individual performances based on separate sets of surrogate labels.

M. 2-D Ensembles for Bulk Learning

The feature abstraction in this study is achieved through stacking meta-classifiers, as an ensemble learning method, which aggregates the predictive results from phenotypic models at the base level and transforms them into higher-level representations. If we consider the process of transforming the base-level feature representation to its level-1 counterpart as a *horizontal ensemble learning* on a per-disease basis, then the computation of the global level-1 model can be thought of as a *vertical ensemble learning* across all infectious diseases at the population scale.

Fig. S3 illustrates graphically the 2-D ensemble learning perspective of the bulk learning system. In particular, the base level, depicted on the left-most column, consists of four phenotypic models denoted by $\Phi\colon\{m, a, b, u\}$. These variables are conveniently named using the first letter of the four phenotypic models used in this study; namely, microbiology (*m*), antibiotic (*a*), blood test (*b*), and urine test (*u*). All base models share the same layout of a logistic regression unit, taking raw features as input, computing their linear combination, followed by a transformation through the sigmoid unit. The output of the base models serves as the input for the next level (i.e. level 1), which is marked by the subscript at each base model output. In the middle column of Fig. S3 is a grid structure wherein each row represents abstract features derived from the base models (e.g. class conditional probabilities). Each local level-1 model is associated with a disease in the bulk learning set and shares the same underlying horizontal ensembles of base models, the layout of which is depicted by the right-most column of Fig. S3. The learning process of the local models is referred to as horizontal ensemble learning since it reflects the grid layout and the fact that each local, per-disease level-1 model can be computed in parallel.

On the other hand, at the bottom of the center grid lies the global level-1 model whose input features are aggregates of those from the local level-1 models. In the paper, we simply combined the level-1 training instances from across all diseases to form a global training set, which, when fed as input to a logistic regression unit, gives rise to a global level-1 model. Since the data aggregation runs across all diseases, this unit of the ensemble learning is referred to as a vertical ensemble. Just as each abstract feature of the local level-1 model is a function of the base-level raw features, each abstract feature of the global level-1 model is in general a function of its local counterparts (making it a composite function of the raw features). It is also possible to weight per-disease level-1 sample in the training of the global model based on, for instance, the number of training instances for each disease, and data availability at the base level, among others.

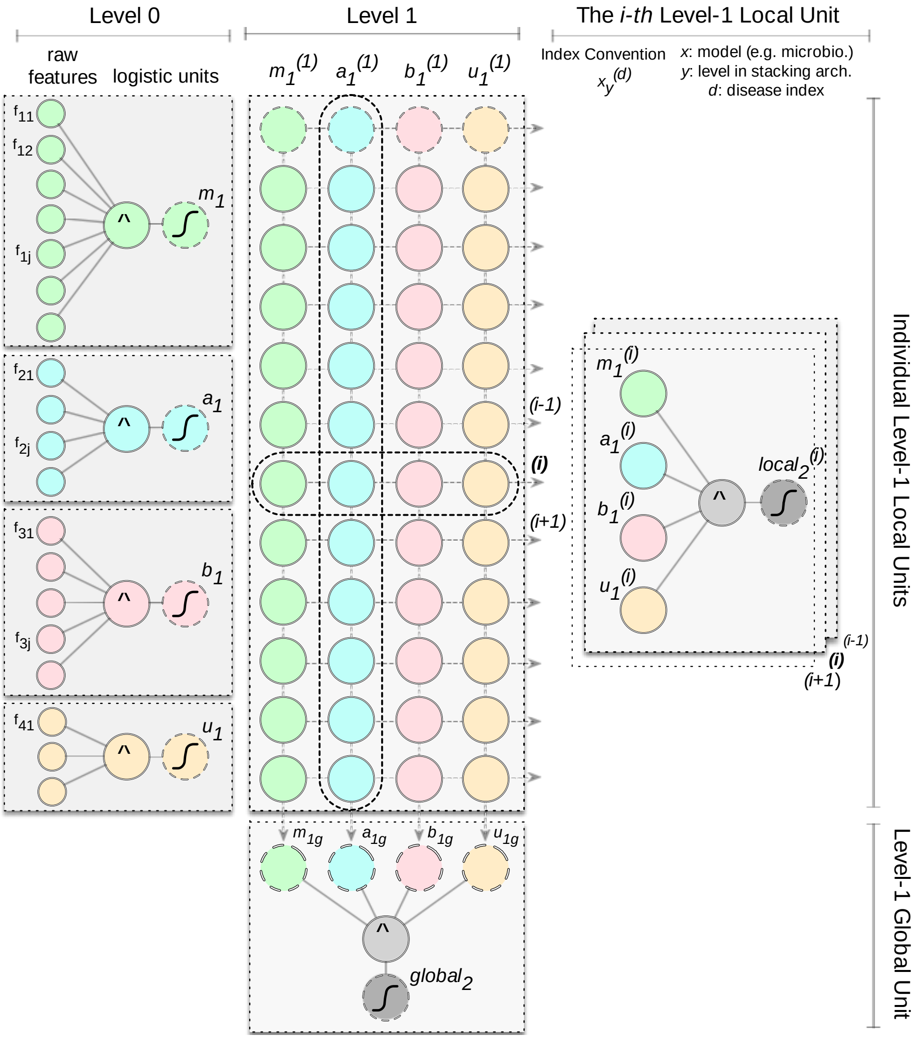


Fig. S3. A "2D-ensemble" perspective of the stacking architecture. One form of ensemble learning unfolds in the horizontal direction where base models (left-most column), consisting of four phenotypic models, are consolidated to form per-disease level-1 models. Each row in the center grid (middle column) represents an abstract feature set for a local level-1 model that predicts a particular disease in the bulk learning set. The internal layout of a local model is depicted on the right-most column. Another form of ensemble learning goes in the vertical direction where level-1 data are aggregated across diseases and subsequently used to train a global level-1 model, which is depicted below the center grid. Since the feature set for the global model is derived from aggregating those of the local models, each feature (e.g. $\boldsymbol{m_{1g}}$) is effectively a function of its local counterparts (e.g. $\{\boldsymbol{m_1^{(i)}}\}$).

N. Training Local and Global Level-1 Models

The evaluation of the level-1 local models were accomplished by a nested validation procedure wherein the outer loop allocates the training split (denoted as *g*) for computing each local model and the test split (denoted as *h*) for evaluating its performance at the end; meanwhile, as the outer-loop training process unfolds, the inner loop takes place behind the scene in computing base models, followed by the feature transformation from the base level to the level-1 representation. There is a subtle difference, however, between the feature transformation in the training split and in the test split. The feature transformation within the training split (*g*), as mentioned in Section 2.4.2, was accomplished via a *k*-fold CV procedure in which only one fold worth of level-0 training data is transformed into the level-1 representation in each iteration; therefore, it takes *k* iterations to finally transform all the data in the training split and in particular, the transformation simply means translating the predictive outputs of the base models to the predefined format of level-1 representation given previously in both Section 2.4.2 in the paper and Section K in this document. Effectively, the level-1 representation associated with the training split (*g*) only utilizes *(k-1)/k* portion of the data, which were used as the training data, i.e. a training set derived from the outer-loop training split, denoted *g-*, for the base models to generate their predictive outputs; the minus sign (with the *g*) is used to reflect a reduced training data. The transformation for the test split, however, utilizes all the data in the (outer-loop) training split (*g*) since they were all used as the training data for base models to generate their predictive outputs over the test split (*h*), resulting in its level-1 counterpart.

After the level-1 data are obtained for each member in the bulk learning set, they are combined as a single training set, which is then used to compute the level-1 global model. At this point, the size of the (level-1) training data for each bulk learning member indeed influences the model accuracy especially in the case of rare diseases with only few data points. If a subset of the selected diseases has an overwhelmingly large dataset compared to the rest of the members, the resulting global model would effectively be fitting towards the diseases with inherently more data points (e.g. pandemic and long-term infections). This is the reason behind excluding diseases with large data volume in bulk learning set as mentioned in Section A. Nonetheless, the bias induced by sample size disparity has much less impact in the context of annotated sample because the number of annotated instances for each disease was made comparable (i.e. 1 or 2 annotated instances per disease).

Similar to the idea of the subset sampling that makes up the annotated sample, an alternative method to even out sample size disparity is to employ methods that cope with imbalanced classes such as over-sampling on minority classes and/or under-sampling of majority classes [2]. The imbalanced classes in the aforementioned literature are analogous to the disease members in the bulk learning set having relatively larger and smaller sample sizes than the average or a predefined reference sample size.

O. Semi-supervised Learning

Semi-supervised learning aims to find potentially better classifiers by leveraging the unlabeled data on top of the existing labeled data, for which numerous methodologies have been proposed and they are largely based on i) cluster assumptions [16,17] and ii) manifold assumptions [18,19]. The main idea behind the cluster assumption is to identify the unlabeled data that are sufficiently similar to the labeled ones such that they can be connected through paths along high-density region of the same class label. On the other hand, the manifold assumption states that the classification function can be defined within a lower dimensional submanifold of the original feature space of higher dimension. Simply put, if feature abstractions can be identified in the lower dimensional space such that they accurately capture the manifold structure of the data, both labeled and unlabeled as a whole, then similar unlabeled data points within this submanfiold are assigned the same class labels as the corresponding labeled data. Semi-supervised learning will be an important component for bulk learning as demonstrated in the paper through virtual annotations.

P. More on the Performance Metric Calculation for the Imaginary ICD-9 Classifier

As mentioned in Section 3.3.1, the training set size is not always identical to the number of annotated cases; however, at level-1, they are consistent due to the policy chosen to consolidate predictive results from the base level (Section 2.4.2). In particular, one can compute, by referencing Table 4, the sensitivity (1) and specificity (0.66) for the ICD-9 system as a classifier from the perspective of level 1 data.

At the base level, however, the same performance metrics may turn out different since a single annotated case could correspond to multiple training instances spread out across different phenotypic models. Specifically, the 2nd and the 3rd column of Table 4 contrasts, by annotation types, the number of annotated cases (83 in total) with that of base-level training instances (254 in total). The sensitivity measure remains the same as before (i.e. 1.00) but the specificity is now 0.68, the ratio of 92 to 43+92.

Further, as a reference for comparing with meta-classifiers, it is possible to simulate an ROC curve for the ICD-9 classifier by assigning probabilities to its predictive labels under appropriate assumptions. Without a prior knowledge of exactly how clinical data were coded in ICD-9, one could assume that the coders were confident in assigning appropriate diagnostic codes to clinical cases; that is, if the ICD-9 were a probabilistic classifier, it would generate a high probability towards 1 to conclude a positive label and by symmetry, a low probability towards 0 to conclude a negative label. This can be simulated by sampling from a negatively skewed distribution between the interval [0.5, 1] and a positively skewed distribution between [0, 0.5). We performed this probability assignment for 30 cycles, each of which represents a possible outcome of this imaginary ICD-9 classifier. Fig. S4 illustrates the performance snapshot of the

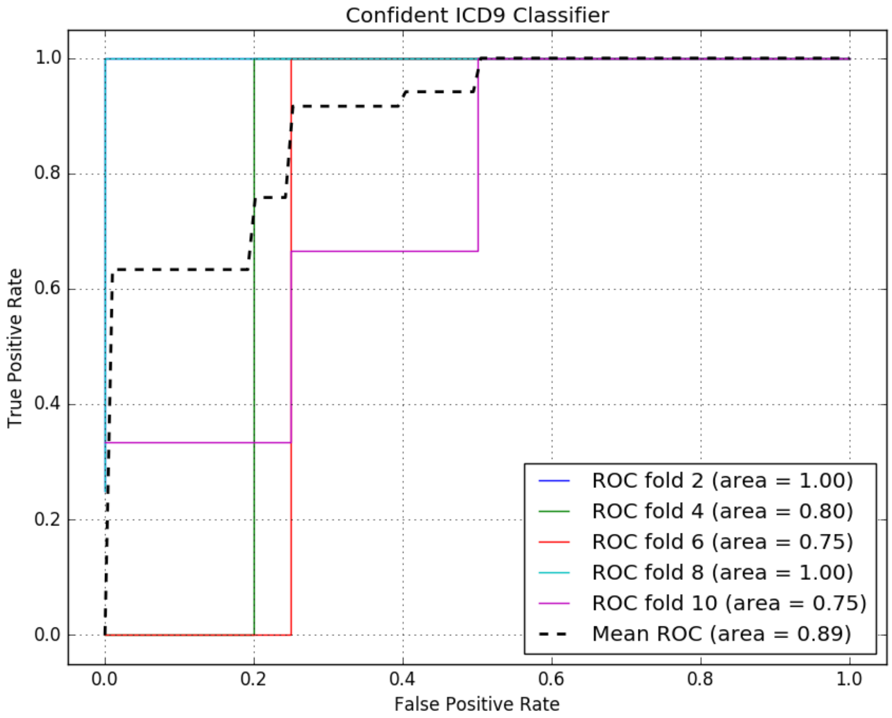


Fig. S4. Performance snapshot of the simulated ICD-9 classifier taken from 1 of the 30 cycles of simulation.

simulated ICD-9 classifier taken from 1 of the 30 cycles of simulation with a mean AUC at 0.89 (versus the grand mean at approximately 0.83). Notice the relatively low AUCs in the 6th and 10th fold, for which the training split would contain the type-FP data driving the performance down whereas the training split of the 2nd and the 8th fold would only contain the type TP and the type TN.

Q. More on the Annotation Set

The annotation set was created by sampling the training instances at the base level. As mentioned in the paper, the strategy for selecting clinical cases can influence the model performance. In this study, we did not take into account how representative these cases are for the rest of the population. The only control in the sampling process was to ensure that each positive annotation corresponds to a distinct ICD-9 code, which we had assumed to represent an infectious disease for simplicity.

The difference between the original training sample and the annotated sample therefore lies only in their labeling. During the phase of deriving abstract features, we had used ICD-9 codes as surrogate labels in the model stacking, assuming they were true labels but only subject to some small errors. Further, assuming that the process of data annotation and the diagnostic coding are independent processes, the ICD-9 codes, once abstract features are derived, can then be used as external explanatory variables in the model learning within the annotated sample. Also, by the assumption that the ICD-9 codes are reasonably accurate, the abstract features would be able generalize sufficiently well in the annotated sample if we used them (instead of raw features) as predictors in statistical models (e.g. logistic regression and SVM). To see if such assumption holds true, the predictive capacity of the ICD-9 derived abstract features would have to be first justified such that evidence indeed supports their use in generalizing the data originally represented in raw features. This is the main reason behind the experimental setup in Section 3.2.2 as a prerequisite step prior to evaluating the gold standard using abstract features and ICD-9 codes as an external predictor.

Incidentally, one subtle distinction between the positive and negative annotations is that there is a much lower probability for type-FN annotations to occur than the type-FP type. As evidenced by our annotated sample, 15 out of the 54 positive annotations are of type FP whereas there is no type-FN annotation found. This is partly due to the mixing strategy of the control data (Section 2.2.4), which allows for instances of any ICD-9 codes but the target ICD-9 to mix in. For a training instance of an ICD-9 code, say 481, to be a type-FN annotation, there must exist some control data that are not only incorrectly labeled but their true identities happen to be the target ICD-9 (i.e. 481) among all the other possible ICD-9 codes, which rarely occurs (unless highly correlated, easily confused ICD-9 codes exist for 481).

Table S1a. Comparison of different meta-classifiers trained by mixing virtual annotations.

Settings	Sensitivity	Specificity	Mean AUC (Repeated 10-fold with 30 cycles)
Level 1 (L1)	1029/1170 (0.88)	212/1320 (0.16)	0.59 (0.51 ~ 0.66)
Level 2 (L2)	812/1170 (0.69)	456/1320 (0.35)	0.52 (0.45 ~ 0.60)
L1 + ICD9	1158/1170 (0.99)	771/1320 (0.58)	0.85 (0.80 ~ 0.89)
L2 + ICD9	910/1170 (0.78)	836/1320 (0.63)	0.74 (0.67 ~ 0.82)
Interactive L1	1154/1170 (0.99)	839/1320 (0.64)	0.83 (0.78 ~ 0.87)

Table S1b. Comparison by annotation types among different meta-classifiers trained by mixing virtual annotations.

Settings	Type TP (39)	Type FP (15)	Type TN (29)	Type FN (0)
Level 1 (L1)	1029/1170 (0.88)	102/450 (0.23)	110/870 (0.13)	n/a
Level 2 (L2)	812/1170 (0.69)	158/450 (0.35)	298/870 (0.34)	n/a
L1 + ICD9	1158/1170 (0.99)	10/450 (0.02)	761/870 (0.87)	n/a
L2 + ICD9	910/1170 (0.78)	104/450 (0.23)	732/870 (0.84)	n/a
Interactive L1	1154/1170 (0.99)	4/450 (0.01)	834/870 (0.96)	n/a

R. Higher-Order Meta-Models

Empirically speaking, the type-FP annotation set is most difficult to predict correctly using only the abstract features derived from the learning hierarchy. The tradeoff between abstract features and the ICD-9 only permits limited improvement upon classification performance in the type-FP region. To further verify this, we introduced a second-order level-1 model, referred to as Interactive L1 in Table S1 (a minor extension to Table 6 in the paper) by allowing its probability attributes and the ICD-9 to interact, which leads to an additional 14 abstract features: 4 features from the set $\{p_i \cdot icd9\}$ and 10 features from $\{p_i \cdot p_j\}$, where both i and j range from 1 to 4. Interactive L1 produces a more expressive, non-linear decision boundary but yet its performance is comparable to L1+ICD9 and tends towards the ICD-9 itself with the type-FP accuracy barely above 0.

S. Towards the type-FP Prediction

As mentioned in Section 4, the abstract feature set plays a role of modulating the ICD-9 signal such that the meta-learner potentially exhibits a performance profile that shifts towards closing the gap between the ICD-9 and the gold standard (by reducing predictive errors in the type-FP region). Unfortunately, the result is not prominent with the current experimental settings in Section 3. The remaining challenge is a method of making educated decisions between either relying on the ICD-9 signal or instead, using the pattern within the abstract features to predict general clinical cases, for which the key lies in the capacity to predict the type-FP region (and the type-FN region if it exists).

Towards that end, we note that although both type-TN and type-FP data correspond to negative examples by the gold standard; however, their properties are very different. In particular, type-TN data assume negative labels according to the mixture strategy of the control data. For instance, a negative instance with respect to the ICD-9 code, 036.0, can only indicate that the underlying clinical case did not have 036.0 documented in their diagnosis records; yet in principle, any other codes can very well serve as possible candidates for its true label so long as the shared-variable criteria hold (Section 2.2.4). By contrast, the type-FP data are corrected to be negative by removing the error occurred in the medical coding process. Since such errors are typically more systematic than random, the data characteristic in the type-FP region is likely to correlate with the type-TP region more than the type-TN region (i.e. within the subset of the annotated sample for which the coder of ICD-9 labeled positive is assumed to have higher correlation with each other than that between the positive and the negative). For this reason, we further postulated that removing type-TN annotations would enable the classifier to identify true labels in the type-FP region with greater ease by using only the type-TP data as counterexamples.

However, the caveat is that the type-FP region in general is smaller than the type-TP region by a significant margin given the decent accuracy of the ICD-9 system (0.72 for positive annotations), which suggests that annotating only a small portion of the positive cases may not produce sufficient type-FP data points. Table S2 compares the performance (in accuracy by annotation types) of different abstract models, trained and evaluated only on the positive annotations (including the virtual annotation). In this case, including the ICD-9 no longer has a prominent effect on driving the performance profile given that its advantage in the type-TN region (with 100% accuracy) is completely removed. Consequently, there is no significant reduction as before in the type-FP accuracy, which remains congruent in the presence of the ICD-9 predictor and exhibits higher accuracy compared to the earlier experimental results in Table 6 in the paper. However, the absence of the type-TN data as counterexamples indeed compromised the accuracy in the type-TP region, suggesting yet another tradeoff in terms of the composition of annotated data for bulk learning.

Hypothetically, if one had access to the knowledge as to whether the ICD-9 and the gold standard agree on the labeling of a training instance, then it would be possible to introduce a moderator variable (denoted by u) for the ICD-9 predictor such that it sets to 1 if the labeling matches (between the ICD-9 and the gold standard) and 0 otherwise. In this case, the original level-1 feature set is augmented to $\{\boldsymbol{t}, \boldsymbol{p}, u \cdot icd9\}$ where the joint vector $(\boldsymbol{t}, \boldsymbol{p})$ remains the same as before while $u \cdot icd9$ represents the interaction between the label-matching indicator u and the ICD-9 as a predictor. Since ICD-9 is a strong predictor that, when included, gravitates the prediction towards itself, mixing ICD-9 with abstract features naturally leads to more errors in the type-FP region, which is undesirable; to address this, u plays the role of suppressing the ICD-9 predictor by setting itself to 0 in the type-FP region so that the only predictors remained therein are the abstract features. Specifically, the accuracy of this moderated level-1 model is specified in the 4th column of Table S2. Although this abstraction mechanism allows for a further increase in the type-FP accuracy without compromising the type-TP's, however, the moderation of ICD-9 relies on an oracle access to the type-FP region. Nonetheless, this result suggests that the type-FP region is not beyond the generalizability of the abstract features; however, it would require that they assume two different sets of weights (i.e. regression coefficients) in order to express two separate decision functions for the type-TP and type-TN regions respectively.

Table S2. Comparison of different feature abstractions with positive annotations only (including virtual annotations).

Annot. Types	Level 1 (L1)	L1 + ICD9	L1+u*ICD9	Level 2 (L2)	L2 + ICD9	ICD9
Type TP (+)	563/1170 (0.48)	573/1170 (0.49)	1095/1170 (0.94)	696/1170 (0.59)	836/1170 (0.71)	1.00
Type FP (−)	243/450 (0.54)	237/450 (0.53)	425/450 (0.94)	187/450 (0.42)	159/450 (0.35)	0.00

As can be seen in Table S2 as well as the earlier experimental settings in Section 3, sharing the same regression coefficients in all regions of annotated data leads to a degradation of the generalizability, a reflection of conflicting evidence that exists between the ICD-9 and the gold standard. This also suggests that an external explanatory factor beyond the ICD-9 itself would be needed in order to assign appropriate values to the variable *u*, which informs the model specifically when to trust ICD-9 and when to use abstract feature set alone. As mentioned earlier in Section L, using multiple surrogate labels is a candidate solution.

References

[1] N. V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357. doi:10.1613/jair.953.

[2] M.M. Rahman, D.N. Davis, Addressing the Class Imbalance Problem in Medical Datasets, Int. J. Mach. Learn. Comput. 3 (2013) 224–228. doi:10.7763/IJMLC.2013.V3.307.

[3] D. Pelleg, A. Moore, Active learning for anomaly and rare-category detection, Adv. Neural Inf. Process. Syst. 18 (2004) 1073–1080.

[4] J.J. Cimino, P.D. Clayton, G. Hripcsak, S.B. Johnson, Knowledge-based approaches to the maintenance of a large controlled medical terminology., J. Am. Med. Inform. Assoc. 1 (1994) 35–50. doi:10.1136/jamia.1994.95236135.

[5] O. Bodenreider, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic Acids Res. 32 (2004) 267D–270. doi:10.1093/nar/gkh061.

[6] C.J. McDonald, S.M. Huff, J.G. Suico, G. Hill, D. Leavelle, R. Aller, A. Forrey, K. Mercer, G. DeMoor, J. Hook, W. Williams, J. Case, P. Maloney, LOINC, a universal standard for identifying laboratory observations: A 5-year update, Clin. Chem. 49 (2003) 624–633. doi:10.1373/49.4.624.

[7] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, H. Pfister, UpSet: Visualization of intersecting sets, IEEE Trans. Vis. Comput. Graph. 20 (2014) 1983–1992. doi:10.1109/TVCG.2014.2346248.

[8] S. Society, S.B. Methodological, Regression Shrinkage and Selection via the Lasso Robert Tibshirani, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (2007) 267–288. doi:10.1111/j.1467-9868.2011.00771.x.

[9] S.S. Lee, H. Lee, P. Abbeel, A.Y.A. Ng, Efficient L1 Regularized Logistic Regression, Compute. 21 (2004) 401. doi:10.1.1.64.1993.

[10] P. Ravikumar, M.J. Wainwright, J.D. Lafferty, High-dimensional Ising model selection using ℓ 1 -regularized logistic regression, Ann. Stat. 38 (2010) 1287–1319. doi:10.1214/09-AOS691.

[11] F. Bach, Bolasso: model consistent Lasso estimation through the bootstrap., Proc. 25th Int. Conf. Mach. Learn. (2008) 33–40. doi:10.1145/1390156.1390161.

[12] G. Solon, S. Haider, J. Wooldridge, What Are We Weighting For?, Cambridge, MA, 2013. doi:10.3386/w18859.

[13] T.A. Lasko, J.C. Denny, M.A. Levy, Computational Phenotype Discovery Using Unsupervised Feature Learning over Noisy, Sparse, and Irregular Clinical Data, PLoS One. 8 (2013) e66341. doi:10.1371/journal.pone.0066341.

[14] J. Gama, P. Brazdil, Cascade Generalization, Mach. Learn. 41 (2000) 315–343. doi:10.1023/A:1007652114878.

[15] V.N. Vapnik, Estimation of Dependences Based on Empirical Data, Springer New York, 2006. doi:10.1007/0-387-34239-7.

[16] X. Zhu, J. Lafferty, Harmonic mixtures, in: Proc. 22nd Int. Conf. Mach. Learn. - ICML '05, ACM Press, New York, New York, USA, 2005: pp. 1052–1059. doi:10.1145/1102351.1102484.

[17] O. Chapelle, J. Weston, B. Schölkopf, Cluster kernels for semi-supervised learning, Adv. Neural Inf. Process. Syst. 15. 7 (2003) 1. doi:10.1016/S0090-3019(02)01037-6.

[18] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, Mach. Learn. Work. Then Conf. 20 (2003) 912. doi:10.1.1.5.68.

[19] M. Belkin, P. Niyogi, Semi-Supervised Learning on Riemannian Manifolds, Mach. Learn. 56 (2004) 209–239. doi:10.1023/B:MACH.0000033120.25363.1e.