Multitask matrix completion for learning protein interactions across diseases

Meghana Kshirsagar[1], Jaime G. Carbonell[2], Judith Klein-Seetharaman[3], and Keerthiram Murugesan[2]

[1]IBM T. J. Watson Research, Yorktown Heights, NY 10598, USA
[2]Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh PA 15213, USA
[3]Metabolic & Vascular Health, Warwick Medical School, University of Warwick, Coventry, UK
mkshirs@us.ibm.com {jgc,judithks,kmuruges}@cs.cmu.edu

Abstract. Disease causing pathogens such as viruses, introduce their proteins into the host cells where they interact with the host's proteins enabling the virus to replicate inside the host. These interactions between pathogen and host proteins are key to understanding infectious diseases. Often multiple diseases involve phylogenetically related or biologically similar pathogens. Here we present a multitask learning method to jointly model interactions between human proteins and three different, but related viruses: *Hepatitis C*, *Ebola* virus and *Influenza A*. Our multitask matrix completion based model uses a shared low-rank structure in addition to a task-specific sparse structure to incorporate the various interactions. We obtain upto a 39% improvement in predictive performance over prior state-of-the-art models. We show how our model's parameters can be interpreted to reveal both general and specific interaction-relevant characteristics of the viruses. Our code and data is available at: http://www.cs.cmu.edu/~mkshirsa/bsl_mtl.tgz

Keywords: host-pathogen protein interactions, multitask learning, matrix completion

1 Introduction

Infectious diseases such as H1N1 influenza, the recent Ebola outbreak and bacterial infections, such as the recurrent *Salmonella* and *E. coli* outbreaks are a major health concern worldwide, causing millions of illnesses and many deaths each year. Key to the infection process are host-pathogen interactions at the molecular level, where pathogen proteins physically bind to human proteins to manipulate important biological processes in the host cell, to evade the host's immune response and to multiply within the host. Very little is known about these protein-protein interactions (PPIs) between pathogen and host proteins for any individual disease. However, such PPI data is widely available across several diseases, and the central question in this paper is: *Can we model host-pathogen PPIs better by leveraging data across multiple diseases?* This is of particular interest for lesser known or recently evolved diseases where the data is particularly scarce. Furthermore, it allows us to learn models that generalize better across diseases by modeling global phenomena related to infection.

An elegant way to formulate the interaction prediction problem is via a graph completion based framework, where we have several bipartite graphs over multiple hosts and pathogens as illustrated in supplementary Figure S1. Nodes in the graphs represent host and pathogen proteins, with edges between them representing interactions (host protein $\overset{interacts}{\underline{\quad}}$ pathogen protein). Given some observed edges (interactions obtained from laboratory based experiments), we wish to predict the other edges in the graphs. Such bipartite graphs arise in a plethora of problems including: recommendation systems (user $\overset{prefers}{\underline{\quad}}$ movie), citation networks (author $\overset{cites}{\underline{\quad}}$ paper), disease-gene

networks (gene $\underline{influences}$ disease) etc. In our problem, each bipartite graph $\mathcal{G}$ can be represented using a matrix M, where the rows correspond to pathogen proteins and columns correspond to host proteins. The matrix entry M_{ij} encodes the edge between pathogen protein i and host protein j from the graph, with $M_{ij} = 1$ for the observed interactions. Thus, the graph completion problem can be mathematically modeled as a matrix completion problem [2].

Most of the prior work on host-pathogen PPI prediction has modeled each bipartite graph separately, and hence cannot exploit the similarities in the edges across the various graphs. Here we present a *multitask* matrix completion method that *jointly models* several bipartite graphs by sharing information across them. From the multitask perspective, a *task* is the graph between one host and one pathogen (can also be seen as interactions relevant to one disease). We focus on the setting where we have a single host species (human) and several related viruses, where we hope to gain from the fact that similar viruses will have similar strategies to infect and hijack biological processes in the human body. Our model is motivated by the following biological intuition governing protein interactions across diseases.

1. An interaction depends on the structural properties of the proteins, which are conserved across similar viruses as they have evolved from common ancestors. We use a component to capture these latent similarities, that is *shared* across tasks.
2. In addition to the shared properties discussed above, each pathogen has also evolved specialized mechanisms to target host proteins. These are unique to the pathogen and can be expressed using a *task-specific* parameter.

This leads us to the following model that incorporates the above ideas. The interactions matrix M_t of task t can be written as: $M_t = \mu_t *$ (shared component)$+(1-\mu_t)*$ (specific component), with hyperparameter μ_t allowing each task to customize its' amount of shared and specific components.

To incorporate the above ideas, we assume that the interactions matrix M is generated from two components. The first component has low-rank latent factors over the human and virus proteins, with these latent factors jointly learned over all tasks. The second component involves a task specific parameter, on which we additionally impose a sparsity constraint as we do not want this parameter to overfit the data. Section 3 discusses our model in detail. We trade-off the relative importance of the two components using task-specific hyperparameters. We can thus learn what is conserved and what is different across pathogens, rather than having to specify it manually.

The applications that we consider involve extremely sparse graphs with a large number of nodes and very few observed edges. There will be nodes i.e proteins that are not involved in any known interactions – the model should be able to predict links between such prior 'unseen' node pairs (this is called the *cold start problem* in the recommendation systems community). For instance, the host-pathogen PPI network of human-Ebola virus (column-3, Table 1) has ≈ 90 observed edges (equivalent to 0.06% of the possible edges) which involve only 2 distinct virus proteins. Any biologist studying virus-human interactions will be more interested in the virus proteins which have yet unknown interactions. The main contributions of this work are:
1. We extend the basic matrix decomposition framework from [1] to the multitask setting by incorporating the structure between the tasks and providing a mechanism for the tasks to share information.
2. We leverage node features which allows us to predict on unseen nodes.
3. We apply the model to an important problem – prediction of interactions in disease-relevant host-pathogen protein networks, for multiple related diseases and demonstrate significant gains in performance over prior state-of-the-art multitask models.
4. We use unlabeled data to initialize the parameters of our model, which gives us a modest boost in prediction performance.

1.1 Background: Host-pathogen protein interactions

The experimental discovery of host-pathogen protein interactions involves biochemical and biophysical methods such as co-immunoprecipitation (co-IP), yeast two-hybrid (Y2H) assays, co-crystallization. The most reliable experimental methods are often very time-consuming and expensive, making it hard to investigate the prohibitively large set of possible host-pathogen interactions – e.g. the bacterium *Bacillus anthracis* with about 2321 proteins when coupled with the 100,000 human proteins gives ≈232 million protein pairs to validate. Computational techniques complement laboratory-based methods by predicting highly probable PPIs. Supervised machine learning based methods use the few known interactions as training data and formulate the interaction prediction problem in a classification setting

1.2 Prior work

Most of the prior work in PPI prediction has focused on building models separately for individual organisms [23, 19] or on building a model specific to a disease in the case of host-pathogen PPI prediction [24, 6, 12]. There has been limited work on combining PPI datasets to learn joint models. [20] proposed a semi-supervised multi-task framework to predict PPIs from partially labeled reference sets. [13] develop a task regularization based framework that incorporates the similarity in biological pathways targeted by various diseases. [29] uses a collective matrix factorization based approach in a multi-task learning setting for within species PPI prediction. The methods used in all prior work on PPI prediction do not explicitly model the features of the proteins and cannot be applied on proteins which have no known interactions available. Our work addresses both these issues. A majority of the prior work in the relevant areas of collaborative filtering and link prediction includes single relation models that use neighbourhood based prediction [21], matrix factorization based approaches [11, 14] and bayesian approaches using graphical models [10, 17].

2 Bilinear low-rank matrix decomposition

In this section, we present the matrix decomposition model that we extend for the multitask scenario. In the context of our problem, at a high level, this model states that – protein interactions can be expressed as dot products of features in a lower dimensional subspace.

Let $\mathcal{G}_t$ be a bipartite graph connecting nodes of type υ with nodes of type ς. Let there be m_t nodes of type υ and n_t nodes of type ς. We denote by $M \in \mathbb{R}^{m_t \times n_t}$, the matrix representing the edges in $\mathcal{G}_t$. Let the set of observed edges be Ω. Let $\mathcal{X}$ and $\mathcal{Y}$ be the feature spaces for the node types υ and ς respectively. For the sake of notational convenience we assume that the two feature spaces have the same dimension d_t [1]. Let $\mathbf{x}_i \in \mathcal{X}$ denote the feature vector for a node i of type υ and $\mathbf{y}_j \in \mathcal{Y}$ be the feature vector for node j of type ς. The goal of the general matrix completion problem is to learn a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that also explains the observed entries in the matrix M. We assume that the function f is bilinear on $\mathcal{X} \times \mathcal{Y}$. This bilinear form was first introduced by [1] and takes the following form:

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^\intercal H \mathbf{y}_j = \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j \tag{1}$$

The factor $H \in \mathbb{R}^{d_t \times d_t}$ maps the two feature spaces $\mathcal{X}$ and $\mathcal{Y}$. This model assumes that H has a low-rank factorization given by $H = UV^\intercal$, where $U \in \mathbb{R}^{d_t \times k}$ and $V \in \mathbb{R}^{d_t \times k}$. The factors U and V project the two feature spaces to a common lower-dimensional subspace of dimension k. While the dimensionality of the feature spaces $\mathcal{X}$ and $\mathcal{Y}$ may be very large, the latent lower dimensional

[1] the dimensions being different does not influence the method or the optimization in any way

subspace is sufficient to capture all the information pertinent to interactions. To predict whether two new nodes (i.e nodes with no observed edges) with features $\mathbf{p}_i$ and $\mathbf{q}_j$ interact, we simply need to compute the product: $\mathbf{p}_i U V^\intercal \mathbf{q}_j$. This enables the model to avoid the cold start problem that many prior models suffer from. The objective function to learn the parameters of this model has two main terms: (1) a data-fitting term, which imposes a penalty for deviating from the observed entries in Ω and (2) a low-rank enforcing term on the matrix H.

The first term can be any loss function such as squared error, logistic-loss, hinge loss. We tried both squared error and logistic-loss and found their behaviour to be similar. The squared error function has the advantage of being amenable to adaptive step-size based optimization which results in a much faster convergence. The low-rank constraint on H (mentioned in (2) above) is NP-hard to solve and it is standard practice to replace it with either the trace-norm or the nuclear norm. Minimizing the trace norm (i.e. sum of singular values) of $H = UV^\intercal$, is equivalent to minimizing $\|U\|_F^2 + \|V\|_F^2$. This choice makes the overall function easier to optimize:

$$\mathcal{L}(U,V) = \sum_{(i,j)\in\Omega} c_{ij}\ \ell(M_{ij}, \mathbf{x}_i^\intercal U V^\intercal \mathbf{y}_j) + \lambda(\|U\|_F^2 + \|V\|_F^2) \quad \text{where } \ell(a,b) = (a-b)^2 \tag{2}$$

The constant c_{ij} is the weight/cost associated with the edge (i,j) which allows us to penalize the error on individual instances independently. The parameter λ controls the trade-off between the loss term and the regularizer.

3 The bilinear sparse low-rank multitask model (BSL-MTL)

In the previous section, we described the bilinear low-rank model for matrix completion. Note that in order to capture linear functions over the features, we introduce a constant feature for every protein (i.e $[\mathbf{x}_i 1]$). We now discuss the multitask extensions that we propose. Let $\{\mathcal{G}_t\}$ where $t = 1 \ldots T$ be the set of T bipartite graphs and the corresponding matrices be $\{M_t\}$. Each matrix M_t has rows corresponding to node type υ_t and columns corresponding to the node type ς_t. The feature vectors for individual nodes of the two types be represented by $\mathbf{x}_{ti}$ and $\mathbf{y}_{tj}$ respectively. Let Ω_t be the set of observed links (and non-links) in the graph $\mathcal{G}_t$. Our goal is to learn individual link prediction functions f_t for each graph. In order to exploit the relatedness of the T bipartite graphs, we make some assumptions on how they share information. We assume that each matrix M_t has a low-rank decomposition that is shared across all graphs and a sparse component that is specific to the task t. That is,

$$f_t(\mathbf{x}_{ti}, \mathbf{y}_{tj}) = \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}, \text{ where } H_t = \mu_t U V^\intercal + (1-\mu_t) S_t \tag{3}$$

As before, the shared factors U and V are both $\mathbb{R}^{d_t \times k}$ (where the common dimensionality d_t of the two node types is assumed for convenience). The matrix $S_t \in \mathbb{R}^{d_t \times d_t}$ is a sparse matrix. The objective function for the multitask model is given by:

$$\mathcal{L}(U,V,\{S_t\}) = \frac{1}{N}\sum_{t=1}^{T} \sum_{(i,j)\in\Omega_t} c_{ij}^t\ \ell\big(M_{t_{ij}}, \mathbf{x}_{ti}^\intercal H_t \mathbf{y}_{tj}\big) + \lambda(\|U\|_F^2 + \|V\|_F^2) + \sum_{t=1}^{T} \sigma_t \|S_t\|_1 \tag{4}$$

Here $N = \sum_t |\Omega_t|$, is the total number of training examples (links and non-links included) from all tasks. To enforce the sparsity of S_t we apply an ℓ_1 norm. In our experiments, we tried both ℓ_1 and ℓ_2 norms and found that the ℓ_1 norm works better.

Optimization: The function $\mathcal{L}(U,V,\{S_t\})$ is non-convex. However, it is convex in every one of the parameters (i.e when the other parameters are fixed) and a block coordinate descent method

called alternating least squares (ALS) is commonly used to optimize such functions. To speed up convergence we use an adaptive step size.

Convergence: The ALS algorithm is guaranteed to converge only to a local minimum. There is work showing convergence guarantees to global optima for related simpler problems, however the assumptions on the matrix and the parameter structure are not very practical and it is difficult to verify whether they hold for our setting.

Initialization of U and V: We tried random initialization (where we randomly set the values to lie in the range [0 1]), and also the following strategies that initialize: $U^0 \leftarrow$ top-k left singular vectors, and $V^0 \leftarrow$ top-k right singular vectors from the SVD of $\sum_{(i,j)\in\Gamma} \mathbf{x}_i \mathbf{y}_j^\intercal$. We set Γ to (a) training examples from all tasks, or (b) a random sample of 10000 unlabeled data from all tasks. We found that using the unlabeled data for initialization gives us a better performance.

3.1 Handling the 'curse of missing negatives'

For the MC algorithm to work in practice the matrix entries M_{ij} should represent interaction scores (range [0 1]) or take binary values (1s for positives and 0s for negatives). Our experiments with PPI probabilities (obtained using the MINT-scoring algorithm) gave bad models. The binary matrix setting requires some observed 0s. However non-interactions are not available as they cannot be verified experimentally for various reasons. Hence we derived a set of 'probable negatives' using a heuristic often used in PPI prediction work [19, 18, 5, 13]. We pair up all virus proteins with all human proteins and sample a random set to be negatives. This heuristic works in practice as the interaction ratio (i.e number of positives in a large random set of protein pairs) is expected to be very low: $\approx 1/100$ to $1/500$. That is, the probability that our negatives contain true positives is negligible.

High class imbalance: We incorporate the prior on the interaction ratio by setting the size of our randomly sampled negatives set equal to 100 times the number of gold standard positives.

4 Experimental setup

4.1 Datasets and features

Task →	***Influenza A***	***Hepatitis C***	***Ebola***
number of HP PPIs (positives)	848	981	90
# of distinct virus proteins in PPIs	54	151	2
# of distinct human proteins in PPIs	362	385	88
total # of virus proteins across strains	542	163	150
number of negatives	84800	98100	9000
density of observed graph‡ (as %)	.15	.60	.06

HP PPI: host-pathogen protein protein interactions

‡: considering all proteins from the two tasks involved.

Note: considering the total number of human proteins to be ≈ 100,000.

Table 1: Tasks and their sizes. Each column corresponds to one bipartite graph between human proteins and the pathogen indicated in the column header. All pathogens are single stranded RNA viruses. The interactions and the protein count both includes data across various strains of each pathogen

We use three human-virus PPI datasets from the PHISTO [25] database (version from 2014), the characteristics of which are summarized in Table 1. The *Influenza A* task includes various strains of flu: H1N1, H3N2, H5N1, H7N3. Similarly, the *Hepatitis* task includes various strains of the virus. [2] All three are single-strand RNA viruses, with *Hepatitis* being a positive-strand ssRNA whereas *Influenza* and *Ebola* are negative-strand viruses. The density of the known interactions is quite small when considering the entire proteome (i.e all known proteins) of the host and pathogen species (last row in Table 1).

Features: Since the sequence of a protein determines its structure and consequently its function, it may be possible to predict PPIs using the amino acid sequence of a protein pair. [22] introduced the "conjoint triad model" for predicting PPIs using only amino acid sequences. They partitioned the twenty amino acids into seven classes based on their electrostatic and water affinities.[3] A protein's amino acid sequence is first transformed to a class-sequence (by replacing each amino acid by its class). For k=3, they count the number of times each distinct trimer (set of three consecutive amino acids) occurred in the sequence. Since there are 343 (7^3) possible trimers (with an alphabet of size 7), the feature vector containing the trimer frequency counts will have 343 elements. To account for protein size, they normalized the counts by linearly transforming them to lie between 0 and 1. Thus the value of each feature in the feature vector is the normalized count for each of the possible amino acid three-mers. We use di-, tri- and four-mers thus leading to a total of 2793 features ($7^2 + 7^3 + 7^4$). Such features have been successfully applied in prior work [6, 13].

4.2 Competing methods

We compare BSL-MTL to various single-task and multitask methods, which includes conventional multitask methods and other recent low-rank and sparse models, and prior work on HP PPI prediction. Wherever appropriate, we concatenated the features of the two node types into a single feature vector. Let $W \in \mathbb{R}^{T \times d_t}$ be the matrix with the task-specific weight vectors $\mathbf{w}_t$. For a uniform comparison we used least squared loss in all the methods. The MALSAR package was used for some of the baselines.

Single task (STL): Ridge regression with ℓ_2 regularization (which performed better than ℓ_1).

MMTL: The mean regularized multitask learning model from [7].

Low rank model (TraceNorm): A low-rank structure is enforced on W by minimizing the nuclear norm $\|W\|_*$.

Sparse + low-rank (SpLowRank) [3]: W is assumed to have the decomposition: $W = P + Q$, where P is sparse and Q has a low-rank structure.

IMC [9, 16]: This is the link-prediction model from Section 2, where data from all tasks is combined without incorporating any task relationships (comparable to the 'union' setting from [28]). U and V are shared by all tasks. We use the same initialization for this method as we do for our model. A comparison to this model tells us how much we gain from the task-specific sparsity component S_t.

MTPL [13]: A biologically inspired regularizer is used to capture task similarity.

BSL-MTL: This work, Bilinear sparse low-rank multitask learning.

4.3 Evaluation setup

We first compare all the methods in two settings, where a small proportion of the available labeled data is randomly sampled and used to train a model which is then evaluated on the remaining

[2] Since we use data from several strains for each task, the PPI data contains some interactions that are interologs. Please see the supplementary section S2 for details

[3] For details of these classes, please refer to the supplementary or the original paper

	10% training			**30% training**		
	Ebola	*Hep-C*	*Influenza*	*Ebola*	*Hep-C*	*Influenza*
STL (Ridge Reg.)	0.189±.09	0.702±.08	0.286±.02	0.220±.03	0.802±.03	0.428±.03
MMTL [7]	0.113±.04	0.767±.03	0.321±.02	0.129±.02	0.802±.04	0.430±.03
Trace-norm	0.199±.11	0.767±.03	0.318±.02	0.207±.02	0.808±.02	0.409±.03
SpLowRank [3]	0.144±.07	0.767±.02	0.318±.02	0.153±.02	**0.814**±.01	0.414±.03
MTPL [13]	**0.217**±.08	0.695±.02	0.345±.02	**0.260**±.05	0.713±.01	**0.496**±.03
IMC [16]	0.087±.04	**0.779**±.02	**0.362**±.01	0.122±.02	0.801±.01	0.410±.03
BSL-MTL	**0.233**±.10	**0.807**±.02	**0.486**±.02	**0.361**±.03	**0.842**±.01	**0.560**±.02

Table 2: Area Under the Precision-Recall curve for each task in the two settings. X% training indicates the fraction of the labeled data used for training and tuning the model with the rest (100-X)% used as test data. We report the average AUC-PR over 10 random train-test splits (stratified splits that maintain the class-skew of 1:100). The standard deviation is also shown. The performance of the best baseline and the overall best method (BSL-MTL) is highlighted in bold. The first row is the only single-task method and all others are multitask models.

data. For the first setting we randomly split the labeled data from each task into 10% training and 90% test, such that the class-skew of 1:100 is maintained in both splits (i.e stratified splits). The second setting uses a 30% training, 70% test split. We use identical splits for all algorithms. In each setting we generate ten random splits and average the performance over the ten runs. Next, we do a standard 10-fold cross validation (CV) experiment (8 folds to train, 1 fold as held-out, 1 fold as test data). In this setting, each algorithm has access to a much larger training set but a significantly smaller test set. The two prior settings (10% and 30%) portray a more realistic multitask scenario where we have access to little training data from each task.

We report the area under the precision recall curve (AUC-PR) along with the standard deviation. AUC-PR has been shown to give a more informative picture of an algorithm's performance than ROC curves in high class imbalance datasets [4] such as ours.

4.4 Parameter tuning

We tune the hyper parameters using a 3 fold cross-validation (CV) on the training split. For all baseline regularization parameters we tried the range: $[100, 50, 10, 1, 0.1, 0.01, 0.05, 10^{-3}]$. To address the class-skew we assign a higher weight to the positives. For BSL-MTL, to tune the rank parameter 'k' we tried: [1, 5, 10, 25, 40, 60, 100] and the regularization parameter controlling the norm of U and V was tuned over the range λ=$\{1...10^{-3}\}$. For each task t, σ_t was varied over the values $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$ and μ_t was varied over $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The optimal setting was: $k = 10$, $\lambda = 0.01$, $\mu_{ebola} = 0.6$, $\mu_{flu} = 0.4$, $\mu_{hepc} = 0.4$, $\sigma_{ebola} = 10^{-5}$, $\sigma_{flu} = \sigma_{hepc} = 10^{-6}$.

5 Results

Table S1 has the AUC-PR for all methods. Note that the AUC-PR of a random classifier model is ≈ 0.01. The first row (STL) is the single-task baseline and all others are multitask models. In general, we notice that multitask learning benefits all tasks. The first three columns show the results in the 10% setting. Our model (last row) has significant gains for *Influenza* (1.3 times better than the next best) and modest improvements for the other tasks. The variance in the performance

is high for the Ebola task (column 1) owing to the small number of positives in the training splits (8 positives). The most benefits for our model are seen in the 30% setting for all tasks, with improvements of 39%, 3% and 12% on the *Ebola*, *Hepatitis* and *Influenza* tasks, respectively. *Ebola*, the data-poorest task, benefits the most. 10 fold CV results are in the supplementary section S1.

5.1 Biological significance of the model

The model parameters U, V and S are a source of rich information which can be used to further understand host-pathogen interactions. Note that our features are derived from the amino acid sequences of the proteins which provide opportunities to interpret the parameters.

Clustering proteins based on interaction propensities We analyze the proteins by projecting them using the model parameters U and V into a lower dimensional subspace (i.e computing $XU^\intercal$ and $YV^\intercal$ to get projections of the virus and human proteins respectively). The principal component analysis (PCA) of this lower dimensional representation is compared with PCA in the original feature space (protein sequence features) in Fig 1. Firstly, the projected data has a much better separation than the original data. Secondly, Fig 1 (right) tells us that Hepatitis-C and Influenza have many proteins with similar binding tendencies, and that these behave differently than most Ebola virus proteins. This observation is not obvious in the PCA of the original feature space (Fig 1 left), where proteins with similar sequences cluster together. These clusters of proteins can be analyzed further for enrichment of Gene Ontology (GO) annotations.

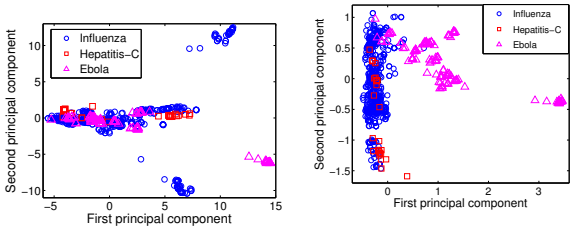

Fig. 1: Principal component analysis (PCA) of virus proteins in the original feature space (left) and the projected subspace (right). The first two principal components are shown. Shape of the points indicates which virus that protein comes from.

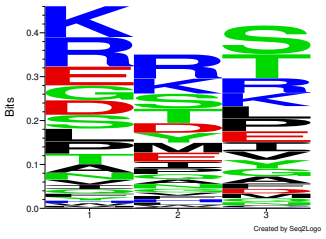

Fig. 2: Top trimer sequence motifs from virus proteins across all three viruses.

Sequence motifs from virus proteins In Figures 2-4, we show sequence motifs derived from the top k-mers that contribute to interactions. The significant entries of the model parameters U, V and $\{S_t\}$ were used to compute these motifs. The top positive-valued entries from the product UV^T indicate which pairs of features: $((f_v, f_h)$: virus protein feature, human protein feature) are important for interactions across all the virus-human PPI tasks. Analogously, the entries from S_t give us pairs of features important to a particular virus-human task 't'. We find that most of the top entries from UV^T correspond to linear virus features, whereas those from the various S_t involve bilinear features. We analyze the k-mers corresponding to the top 20 features from each of the matrices.

Note that our features do not directly correspond to a unique amino-acid k-mer (see Section 4.1): the virus feature f_v will map to several amino-acid sequences (for instance KKCC, KRCC, RRCC

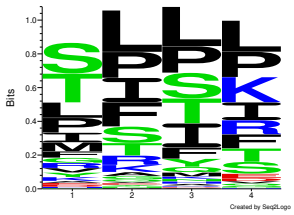

Fig. 3: Sequence motif constructed from the top four-mer features of virus proteins across all three viruses.

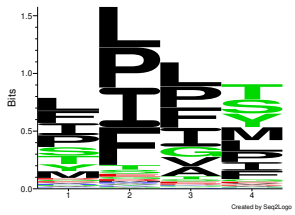

Fig. 4: Sequence motif constructed from the top four-mer features of human proteins

etc all map to a single feature due to the molecular similarity between the amino acids K and R being both positively charged). Given the set of top virus features we can obtain the corresponding set of amino-acid k-mers, say AA_v, by reversing the feature-generation step. However most of the possible k-mers do not appear in the training data (ex: out of the 160,000 ($=20^4$) possible 4-mers $\approx$24,000 appear). Let AA_{tr} be the set of amino-acid k-mers that appear in the training data. Then, the intersection $I_v = AA_v \cap AA_{tr}$ gives us the important amino-acid k-mers from virus proteins w.r.t interaction prediction. To summarize I_v, we use a popular tool Seq2Logo [26] to generate a sequence motif. The logos for the two-, three-, four-mers from I_v are generated independently. Since we only want to summarize, we use the Shannon logo type (which does not consider any background amino-acid distribution) with the following settings: clustering-method=None, weight-on-prior=1 (pseudo-counts do not make sense in our analysis). Figures 2 and 3 show the motif that is common across viruses.

This procedure described above is used to analyze the most significant human protein features, obtained from the matrix UV^T, which are shown in Figure 4. We observe that the shared trimer motif for virus proteins in Figure 2 is dominated by hydrophilic amino acids (K, R, T, D, E). All other motifs seem to be dominated by hydrophobic residues (I, P, L, V, A, G) though S and T do appear in some motifs as well. The human protein motifs are shown in Fig 4. Further analysis with trimers and tetramers specific to the pathogens is in the supplementary, sections S3 and S4. These task-specific features (i.e k-mers) are obtained from the matrices S_{ebola}, S_{hepc} and S_{flu} respectively. In most cases, the first position of the trimer was less significant than the second and third, while for the tetramer all four positions show clear preferences.

Phosphorylation sites: We found the frequent occurrence of S and T and sometimes Y in the motifs striking and suspected this may be related to the amino acids being frequent targets of phosphorylation. Phosphorylated sites often serve as PPI sites, and databases such as Phosphosite [8] are repositories for known sites in human proteins. Since these are sites in human proteins, we searched for the patterns from the 4-mer motif in Fig 4 and found several to be flanking known phosphorylation sites in human proteins: `LLLs`, `LLLt`, `ILLs`, `PPPs`, `PIPs`, `PIPt`, `LIPs`, `PLLt` (lowercase indicates the putative phosphorylation site). This observation also supports the notion that the motifs predictive of interaction are biologically significant.

Novel interactions with Ebola proteins The top four Ebola-human PPI are all predictions for the Ebola envelope glycoprotein (GP) with four different human proteins (Note: GP is not in the gold standard PPIs). There is abundant evidence in the published literature [15] for the critical role played by GP in virus docking and fusion with the host cell. A list of interactions will be provided on the corresponding authors website.

6 Conclusions and future extensions

This work developed and tested a new method based on low-rank matrix completion for sharing information across tasks The method was evaluated in the host-pathogen protein interaction domain for three pathogens (three tasks) and exhibited significant increases in prediction accuracy. The model parameters provide several avenues for studying host-pathogen interactions for biologists that can lead to interesting observations and insights. Finally, the model we present is general enough to be applicable on other problems such as: gene-disease relevance prediction across organisms or disease conditions.

References

1. J. Abernethy, F. Bach, T. Evgeniou, and J. P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research (JMLR)*, 2009.
2. E. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2008.
3. Jianhui Chen, Ji Liu, and Jieping Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):22, 2012.
4. Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.
5. M.D. Dyer et al. Supervised learning and prediction of physical interactions between human and hiv proteins. *Infect., Genetics and Evol.*, 11:917–923, 2011.
6. M.D. Dyer, T.M. Murali, and B.W. Sobral. Computational prediction of host-pathogen protein-protein interactions. *Bioinf.*, 23(13):i159–66, 2007.
7. T. Evgeniou and M. Pontil. Regularized multi-task learning. *ACM SIGKDD*, 2004.
8. Peter V Hornbeck, Bin Zhang, Beth Murray, Jon M Kornhauser, Vaughan Latham, and Elzbieta Skrzypek. Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic acids research*, 43(D1):D512–D520, 2015.
9. Prateek Jain and Inderjit S. Dhillon. Provable inductive matrix completion. 2013.
10. Rong Jin, Luo Si, and ChengXiang Zhai. Preference-based graphic models for collaborative filtering. In *UAI*, pages 329–336, 2002.
11. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
12. M. Kshirsagar, J. G. Carbonell, and J. Klein-Seetharaman. Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics*, 2012.
13. M. Kshirsagar, J. G. Carbonell, and J. Klein-Seetharaman. Multi-task learning for host-pathogen protein interactions. *Bioinformatics*, 2013.
14. A. K. Menon and C. Elkan. Link prediction via matrix factorization. *ECML*, 2011.
15. Asuka Nanbo, Masaki Imai, Shinji Watanabe, et al. Ebolavirus is internalized into host cells via macropinocytosis in a viral glycoprotein-dependent manner. *PLoS pathogens*, 6(9):e1001121, 2010.
16. Nagarajan Natarajan and Inderjit S. Dhillon. Inductive matrix completion for predicting genedisease associations. *Bioinformatics*, 2014.
17. Dinh Q Phung, Svetha Venkatesh, et al. Ordinal boltzmann machines for collaborative filtering. In *UAI*, pages 548–556. AUAI Press, 2009.
18. Y. Qi, H.K. Dhiman, Z. Bar-Joseph, et al. Systematic prediction of human membrane receptor interactions. *Proteomics*, 23(9):5243–55, 2009.
19. Y. Qi et al. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins*, 63(3):490–500, 2006.
20. Y. Qi, O. Tastan, J. G. Carbonell, J. Klein-Seetharaman, and J. Weston. Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins. *Bioinformatics*, 2010.
21. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

22. J. Shen et al. Predicting protein-protein interactions based only on sequences information. *PNAS*, 104:4337–4341, 2007.
23. R. Singh, J. Xu, and B. Berger. Struct2net: Integrating structure into protein-protein interaction prediction. *Pacific Symposium on Biocomputing*, 2006.
24. O. Tastan et al. Prediction of interactions between hiv-1 and human proteins by information integration. *Pac. Symp. Biocomput.*, (14):516–527, 2009.
25. S. D. Tekir, Ali S., Tunahan C., and Kutlu O. U. Infection strategies of bacterial and viral pathogens through pathogen-host protein protein interactions. *Frontiers in Microbial Immunology*, 2012.
26. Martin Christen Frølund Thomsen and Morten Nielsen. Seq2logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research*, 40(W1):W281–W287, 2012.
27. Randi Vita, James A Overton, Jason A Greenbaum, et al. The immune epitope database (iedb) 3.0. *Nucleic acids research*, 43(D1):D405–D412, 2015.
28. C. Widmer, J. Leiva, Y. Altun, and G. Ratsch. Leveraging sequence classification by taxonomy-based multitask learning. *RECOMB*, 2010.
29. Q. Xu, E. W. Xiang, and Q. Yang. Protein-protein interaction prediction via collective matrix factorization. *International Conference on Bioinformatics and Biomedicine*, 2010.

Supplementary Note

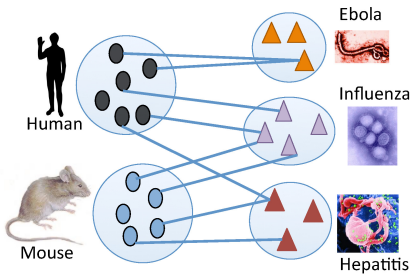

Fig. S1: Multiple interaction graphs, each one is bipartite. On the left are host proteins and on the right are proteins from different viruses. Edges represent protein interactions

S1 10 fold cross validation

In the 10 fold cross-validation (CV) experiments, a much larger training set is available. So the single task baseline becomes much harder to beat as it can independently tune parameters for each task. All methods have a higher variance in their performance (we think this is due to the smaller and hence more variable test data in a 10 fold CV setting). Our method improves on only the *Hep-C* task.

	10 fold CV		
	Ebola	*Hep-C*	*Influenza*
STL	0.28±.11	0.74±.05	0.66±.08
MMTL	0.28±.11	0.68±.04	0.40±.03
Norm	0.20±.09	0.70±.06	0.44±.05
Rank	0.18±.08	0.70±.06	0.45±.04
MTPL	0.27±.08	0.67±.05	0.48±.06
IMC	0.16±.08	0.72±.06	0.45±.06
BSL-MTL	0.28±.11	0.82±.05	0.62±.02

Table S1: Area Under the Precision-Recall curve for 10 fold CV. The first row is the only single-task method and all others are multitask models.

S2 Homologous interactions

Since we use data from several strains for each task, the PPI data contains some interactions that are interologs. We observed this for two of the tasks: *Hepatitis-C* and *Influenza A*. Note that we

did not find any interologs *across* tasks. These homologous interactions in the various strains are reported by different experimental studies and we believe their presence suggests the confidence of such interactions. Since we use only PPI derived from experimental methods (and not electronically inferred), we do not exclude any homologous interactions from our training.

Removing homologs from the evaluation:
The number of distinct PPI for which interologs or homologs exist is very small (≈20), but there are several homologs for each such PPI. Note that any benefits achieved from the presence of homologous interactions will be available to all the methods (since the models were built using identical train/test data). Here we present results on the 10% setting by removing all homologs from the test data (note: the *Ebola* task did not have any homologs). For lack of space we only mention the trend for our method (BSL-MTL) - *Hep-C*: 0.85 and *Influenza:* 0.45. The other methods show a similar trend; our method continues to outperform by significant margins.

S3 Biological significance of interactions

Evidence in IEDB[4]: We found experimental evidence for the significance for the virus motifs in the Immune Epitope[5] database (IEDB) [27]. The pattern `IVGG` from the *Hepatitis-C* motif in Fig. S3 is found in 53 epitopes. From the *Ebola* motifs in Fig S2, we find that `TLAT` is part of six different epitopes, `SLTT` appears in three epitopes. `PLIK`, `SLLL` from the *Influenza* motif are also found in many epitopes. Finding that the virus k-mer patterns predicted by our method are recognized by human antibodies is a further validation of its performance. Further, using higher dimensional k-mers (where k=7, 8, 9) as features in our model will give motifs from which complete epitopes can be derived. Our model thus has applications in epitope prediction as well, where conventional methods consist of scanning all possible k-mers from protein sequences to identify likely epitopes.

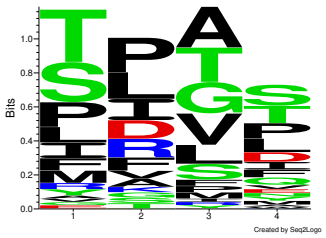

Fig. S2: Sequence motif from top four-mer features specific to *Ebola* proteins

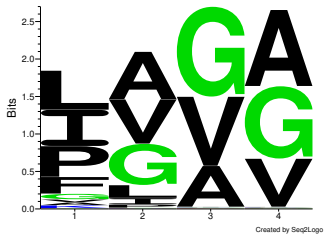

Fig. S3: Sequence motifs specific to *Hepatitic-C* proteins that are also important to interactions.

[4] `www.iedb.org`
[5] An epitope is a very short sequence from the virus that binds to human antibodies

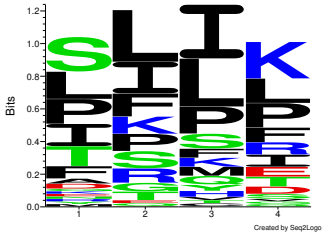

Fig. S4: Sequence motifs specific to *Influenza* proteins that are also important to interactions.

S4 Tri-mers

Below, we show the sequence motifs from the tri-mers found to be highly relevant to predicting interactions between human and viral proteins.

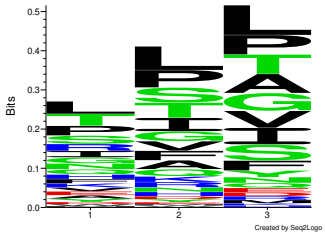

Fig. S5: Sequence motifs specific to *Ebola* proteins.

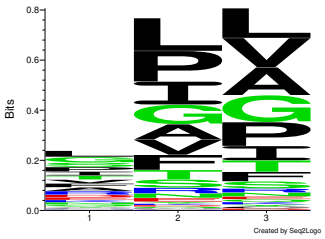

Fig. S6: Motifs specific to *Hepatitic-C* proteins.

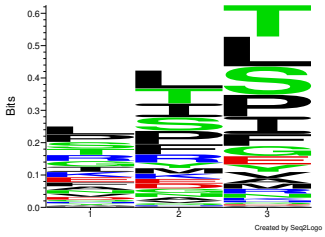

Fig. S7: Sequence motifs specific to *Influenza* proteins.

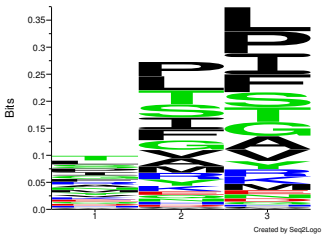

Fig. S8: Sequence motif constructed from the top tri-mer features of human proteins