Gao, Junning; Yao, Shuwei; Mamitsuka, Hiroshi; Zhu, Shanfeng

AiProAnnotator

Published in:
Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018

DOI:
10.1109/BIBM.2018.8621517

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Gao, J., Yao, S., Mamitsuka, H., & Zhu, S. (2018). AiProAnnotator: Low-rank Approximation with network side information for high-performance, large-scale human Protein abnormality Annotator. In *Proceedings - 2018 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2018* (pp. 13-20). Article 8621517 IEEE. https://doi.org/10.1109/BIBM.2018.8621517

AiProAnnotator: Low-rank Approximation with network side information for high-performance, large-scale human Protein abnormality Annotator

Junning Gao* Shuwei Yao* Hiroshi Mamitsuka‡§ Shanfeng Zhu*†
*School of Computer Science and Shanghai Key Lab of Intelligent Information Processing, Fudan University
†Center for Computational System Biology, Fudan University, Shanghai 200433, China
‡Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji 611-0011, Japan
§Department of Computer Science, Aalto University, Finland
Email: zhusf@fudan.edu.cn

***Abstract*—Annotating genes/proteins is a vital issue in biology. Particularly we focus on human proteins and medical annotation, which both are important. The most proper data for our annotation is human phenotype ontology (HPO), which are sparse but reliable (well-curated). Existing approaches for this problem are feature-based or network-based. The feature-based approach can incorporate a variety of information, by which this approach is more appropriate for noisy data than reliable data, while the network-based approach is not necessarily useful for sparse data. Low-rank approximation is very powerful for both sparse and reliable data. We thus propose to use matrix factorization to approximate the input annotation matrix (proteins $\times$ HPO terms) by factorized low-rank matrices. We further incorporate network information, i.e. protein-protein network (PPN) and network from HPO (NHPO), into the framework of matrix factorization as graph regularization over the two low-rank matrices. That is, the input annotation matrix is factorized into two low-rank factor matrices so that they can be smooth over PPN and NHPO. We call our software of implementing the above method "AiProAnnotator", which in this paper has been empirically examined using the latest HPO data extensively under various experimental settings, including performance comparison under cross-validation, computation time and case studies, etc. Experimental results showed the high predictive performance and time efficiency of AiProAnnotator clearly.**

I. Introduction

Annotating gene/protein function is a fundamental issue in biology which has been well-considered in a variety of domains in bioinformatics, such as sequence comparison, gene function prediction and text mining for cooccurrent genes, etc. A typical example is annotating genes by Gene Ontology (GO) terms [1], which turns into a massive-scale multilabel classification problem with noisy, imbalanced labels, covering a wide range of species and label types. We address a similar issue, but our focus is more specific and twofold: 1) human proteins, because they are most important among those of all species. Also human genes have an ethical issue on opening sequencing results, which in general would make computationally annotating human proteins more important. 2) Medical side of annotation, such as abnormalities, disorders and diseases, which would be also practically the most important aspect in function assignment. These two focuses would be reasonable, since an ontology called human phenotype ontology (HPO) has been generated by [2] and well-developed, particularly emphasizing diseases, more generally abnormalities [3].

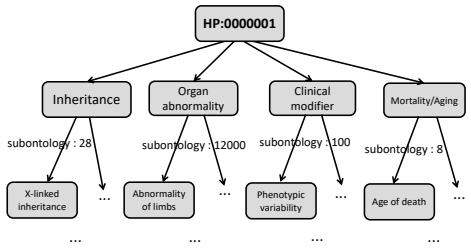

Fig. 1: HPO has four subontologies, in which terms are related with each other as a directed acyclic graph, mainly keeping the hierarchical tree structure.

HPO collects data from major databases of human hereditary disorders, by which HPO currently contains all clinical descriptions in OMIM [4], all annotated entries of Orphanet [5] and over 60 recurrent syndromes in DECIPHER [6]. HPO has four subontologies (see Fig. 1): organ abnormality, inheritance, clinical modifier and mortality/aging, which have around 12,000, 28, 100 and 8 terms, respectively, meaning that organ abnormality is highly weighed. The HPO terms are connected to each other, basically holding a hierarchical tree structure, keeping more general classes closer to the root and vice versa, while strictly the structure is a directed acyclic graph (DAG), since specialized terms can be related with more than one more general terms. A noteworthy point of the HPO is that annotations are sparse such that only 261,183 annotations (1.2%) are made for the entire matrix of 3,354 proteins and 6,229 HPO terms (January 2017 release), with increasing around 23,448 annotations (9% of 261,183) in 2017. The current number of annotated proteins, i.e. around 3,500, is expected to increase in the future, implying that computational predictive approaches would be urgent.

Existing computational approaches for HPO annotation

would have two directions: 1) feature-based and 2) network-based approaches. The feature-based approach uses information of genes/proteins as features to predict appropriate terms to be annotated. In general the feature-based approach is very useful for sparse data and also noisy cases, because this approach can incorporate auxiliary information into the original sparse, noisy input data, by which data can be augmented to improve the predictive performance. In fact, this type of approach, particularly learning to rank, has been proved to be very useful for GO annotation [1]. However comparing with GO annotation, HPO annotation are much more curated, reliable and stable. Also HPO annotations are sparse as shown above, but this sparseness is much less than GO, by focusing on human proteins and abnormality terms only. Besides, existing feature-based approaches rarely take advantage of HPO information, e.g. hierarchical structure and co-occurrence of HPO terms. The network-based approach is currently more prevalent than the feature-based approach for HPO annotation. In general a method in this direction uses at least two inputs: the network from HPO (hereafter NHPO) and HPO annotation (the matrix of HPO terms $\times$ proteins) as both networks, and sometimes one more input: the relationships over proteins/genes (hereafter protein-protein network (PPN)). One way is that these networks are combined together to form one large-scale network for HPO annotation. For example, random-walk [7] or some simple score computation [8] have been used. Also optimization-based methods use the input matrix directly [9], [10]. However these methods cannot work well enough for sparse data, in which nodes are disconnected so often, even if they can be relevant to each other.

From these observations, we use matrix factorization to approximate the HPO annotation matrix (of proteins $\times$ HPO terms) by factorized low-rank matrices, which can capture the factors in HPO term annotation of proteins. Since the HPO annotation matrix is binary (zero or one), our matrix factorization should be reasonably non-negative matrix factorization (NMF), which has proved to be useful for solving sparse problems in biology [11]–[14]. Also we transform our other two networks, NHPO and PPN, into adjacency matrices or graphs, and then generate two graph Laplacians. We use these two graph Laplacians to regularize the NMF over the main HPO annotation matrix, meaning that the main matrix is factorized into two matrices so that each of the two low-rank matrices should be smooth over each of the given two graphs, i.e. NHPO and PPN. We call our model "AiProAnnotator" (AiPA for short). We empirically tested the performance of AiPA, comparing with three network-based approaches, which are raised in the related work section, using the latest large-scale HPO data with around 300,000 annotations. Experimental results showed that AiPA outperformed competing methods in a variety of settings of data usage, such as cross-validation and using test data independent of training data, etc., indicating the effectiveness of low-rank approximation and network information on the human phenotype ontology problem. Finally empirical case studies confirmed the speed and practical usefulness of AiPA.

II. Related work

As mentioned in introduction, existing work can be divided into two approaches: feature-based and network-based.

The two notable methods of the feature-based approach are PHENOstruct [15] and Clus-HMC-Ens [16]. Feature-based methods, in general, generate a feature vector and HPO annotations (labels) for each gene as an input of a classifier, and then the trained classifier is used for prediction. This procedure is true of these two methods. Also both these methods are used for both GO and HPO annotations (or originally they are used for GO annotations and then applied to HPO annotation). In this sense, we can say that the difference of HPO annotation from GO annotation has not been necessarily considered so well in these methods. Clus-HMC-Ens uses decision tree ensembles, while PHENOstruct (originally GOstruct for GO annotation) uses a modified support vector machine (SVM).

The network-based approach uses two networks (HPO annotation matrix and NHPO) and in some cases one more network (PPN and totally three) and run some algorithm over these networks for HPO annotation. The assumption behind the network-based approach is that properties of nodes in networks should be similar more as the connected nodes are more similar. We raise three methods as representatives below, which are all used in our experiments for comparison.

Xie et al. conduct random walk over the Kronecker product graph between PPN and NHPO [17], [18]. We call this method BiRW, standing for Bi–network Random Walk. Petegrosso et al. [9] use all three networks, HPO annotation, NHPO and PPN, as well as two more networks, GO annotation and network from GO. The idea behind this approach is to transfer information of GO annotation to HPO annotation through GO, PPN and NHPO, particularly regularizing two annotation matrices by these three networks. This method is called tlDLP, standing for transfer learning dual label propagation [9]. However transferring GO annotation to HPO annotation is not our setting and out of our scope. So we focus on a model, called DLP, standing for dual label propagation, which is generated while developing the final model using GO annotation. In DLP, the objective function is the weighted error loss between the target matrix of HPO annotation, being regularized by both NHPO and PPN. DLP is very similar to our model, particularly using two graph regularizers (which are the same as those in our model), while a significant difference is DLP uses the target HPO annotation matrix directly to be regularized by NHPO and PPN, while our model factorizes the target HPO annotation matrix into two low rank matrices, each being separately regularized by NHPO and PPN.

The last method is called ontology-guided group lasso (OGL), which uses, instead of the graph regularizer for HPO in DLP, an ontology-guided group norm for HPO [10]. Except this point, OGL is the same as DLP. So OGL is also different from our model, in the sense that low rank matrices are not generated.

The biggest disadvantage of the network based method is data sparseness directly affects the performance heavily, and

particularly, as mentioned in introduction, HPO annotation is still sparse. Also all of them use the input HPO annotation matrix directly, which is rather large, by which the computational burden is usually very heavy, which is another big problem.

III. Methods

A. Notation

Let N_p and N_h be the number of proteins and HPO terms, respectively. Let $\boldsymbol{Y}$ be the $(N_p \times N_h)$ HPO annotation matrix, where $\boldsymbol{Y}_{ij} = 1$ if protein i is annotated by j; otherwise $\boldsymbol{Y}_{ij} = 0$. Let $\boldsymbol{S}^p$ be PPN, i.e. protein-protein network, where $\boldsymbol{S}^p_{i,j}$ is the score of the relationship between protein i and protein j. Similarly let $\boldsymbol{S}^h$ be the network of HPO terms, generated from HPO, where $\boldsymbol{S}^h_{i,j}$ is the similarity between term i and term j. Our goal is to estimate $\hat{\boldsymbol{Y}}$ by using $\boldsymbol{Y}$, $\boldsymbol{S}^p$ and $\boldsymbol{S}^h$.

B. Proposed method

1) Preprocessing: generating a network from HPO: We generate NHPO, i.e. a network from HPO, by measuring the similarity between two HPO terms in HPO, we use a similarity defined in [19]. This similarity has been used extensively as a semantic similarity in natural language processing to define the similarity between two labeled nodes by how many times these labels co-occur in a corpus.

For HPO, this similarity between two HPO terms s and t is defined as:

$$\boldsymbol{S}^h_{s,t} = \frac{2 \cdot I(\text{mca}(s,t))}{I(s) + I(t)} \tag{1}$$

where $I(s) = \log(p(s))$, $p(s) = \frac{\text{count}(s)}{N_p}$, $\text{count}(s)$ is the number of proteins annotated by s and $\text{mca}(s,t)$ is given as follows:

$$\text{mca}(s,t) = \arg \min_{k \in \text{A}(s,t)} p(k),$$

where $\text{A}(s,t)$ is the set of all common ancestors of s and t.

This similarity between s and t is used as a weight attached to the edge between nodes s and t of the network of HPO. The similarity would be larger as annotations by s and t are shared by a larger number of proteins. Also this would happen more likely as the common ancestor of s and t is closer. This means that $\boldsymbol{S}^h$ considers both the number of proteins annotated by two HPO terms at the same time and also the distance between the two HPO terms in the hierarchical structure.

2) Nonnegative matrix factorization (NMF): NMF aims to approximate the original input matrix by two low-rank matrices, which turns into capturing the factors of the original matrix. Mathematically input matrix $\boldsymbol{Y} \in \mathbb{R}_+^{N_p \times N_h}$ can be factorized into two rank K matrices, $\boldsymbol{U} \in \mathbb{R}_+^{N_p \times K}$ and $\boldsymbol{V} \in \mathbb{R}_+^{N_h \times K}$ to minimize the objective function of the Flobenius norm:

$$\begin{aligned} J &= ||\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^T||_F^2 \\ &s.t. \quad \boldsymbol{U} \geq 0, \boldsymbol{V} \geq 0. \end{aligned} \tag{2}$$

Usually L2 (Tikhonov) regularization is added to Eq. (2) to avoid overfitting of $\boldsymbol{U}$ and $\boldsymbol{V}$ to training data.

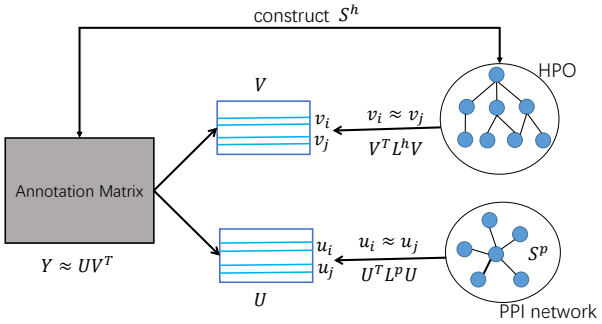

Fig. 2: Model

The input $\boldsymbol{Y}$ has unknown (missing) values and we filter out them by using weight matrix $\boldsymbol{W} \in \{0,1\}^{N_p \times N_h}$, where $\boldsymbol{W}_{ij} = 1$ if the annotation between protein i and HPO terms j ($\boldsymbol{Y}_{ij}$) is known ; otherwise zero[1]. Then the formulation is given as follows:

$$\begin{aligned} J_{\text{NMF}} &= ||\boldsymbol{W} \odot (\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^T)||_F^2 + \lambda(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) \\ &s.t. \quad \boldsymbol{U} \geq 0, \boldsymbol{V} \geq 0, \end{aligned} \tag{3}$$

where $\odot$ is Hadamard product (element-wise product), and λ is a regularization coefficient.

In prediction, we define $\hat{\boldsymbol{Y}}$ by using the estimated $\boldsymbol{U}$ and $\boldsymbol{V}$, where $\hat{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{V}^T$, by which unknown missing annotations can be estimated.

3) Network regularization: From (1), we have similarity matrix $\boldsymbol{S}^h$ of HPO which we can think is equivalent to the adjacency matrix or network. We then regularize $\boldsymbol{V}$ by using $\boldsymbol{S}^h$, which results into the standard graph regularizer using graph Laplacian as follows:

$$\begin{aligned} &\frac{1}{2}\sum_{i,j} \boldsymbol{S}^h_{i,j}||\boldsymbol{V}_i - \boldsymbol{V}_j||^2 \\ &= \text{trace}(\boldsymbol{V}^T(\boldsymbol{D}^h - \boldsymbol{S}^h)\boldsymbol{V}) \\ &= \text{trace}(\boldsymbol{V}^T\boldsymbol{L}^h\boldsymbol{V}), \end{aligned} \tag{4}$$

where $\boldsymbol{V}_i$ is the i-th row vector of $\boldsymbol{V}$, $\boldsymbol{D}^h$ is a diagonal matrix the corresponding diagonal is the node degree and $\boldsymbol{L}^h = \boldsymbol{D}^h - \boldsymbol{S}^h$ is the graph Laplacian of S_h .

Similarly we can have the graph regularizer for PPN from $\boldsymbol{S}^p$

$$\text{trace}(\boldsymbol{U}^T\boldsymbol{L}^p\boldsymbol{U}), \tag{5}$$

where $\boldsymbol{L}^p = \boldsymbol{D}^p - \boldsymbol{S}^p$ is the graph Laplacian of S_p and similarly $\boldsymbol{D}^p$ is the graph degree diagonal matrix.

Minimizing this graph regularization term means the fidelity of the low rank matrix $\boldsymbol{V}$ (or $\boldsymbol{U}$) to the smoothness over $\boldsymbol{S}^h$ (or $\boldsymbol{S}^p$). Note that this graph regularization is standard and has been used in a variety of applications already [20].

[1]Note that $\boldsymbol{W}$ is also an input of our method.

Algorithm 1 The training algorithm of AiProAnnotator

Require: Annotation matrix, $\boldsymbol{Y} \in \mathbb{R}^{n_p \times n_h}$;
protein-protein network (PPN), $\boldsymbol{S}^p \in \mathbb{R}^{n_p \times n_p}$;
Hierarchical structure of HPO terms
Ensure: $\boldsymbol{U}$, $\boldsymbol{V}$.
1: Generate the network of HPO terms, i.e. NHPO, $\boldsymbol{S}^h$ by (1).
2: **repeat**
3: Update $\boldsymbol{V}$ by (10) .
4: Update $\boldsymbol{U}$ by (11).
5: **until convergence**
6: Return : $\boldsymbol{U}$, $\boldsymbol{V}$ and $\hat{\boldsymbol{Y}} = \boldsymbol{U}\boldsymbol{V}^T$

4) Model formulation: Combining (3), (4) and (5), our model formulation is given as follows:

$$\min_{\boldsymbol{U}\geq 0, \boldsymbol{V}\geq 0} \quad \|\boldsymbol{W} \odot (\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^T)\|_F^2 + \lambda(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2) + \alpha \text{trace}(\boldsymbol{U}^T \boldsymbol{L}^p \boldsymbol{U}) + \beta \text{trace}(\boldsymbol{V}^T \boldsymbol{L}^h \boldsymbol{V}), \quad (6)$$

where α and β are the regularization coefficients, balancing between the approximation loss and graph smoothness.

5) Model optimization: Minimizing Eq. (6) is a biconvex problem regarding $\boldsymbol{U}$ and $\boldsymbol{V}$, and so we take a regular manner for solving this problem: alternating least square (ALS), which alternately optimizes one of the two parameters, fixing the other, until convergence.

We first show the derivation of the updating rule of $\boldsymbol{V}$. Fixing $\boldsymbol{U}$, the objective function can be written as follows:

$$\begin{aligned} J(\boldsymbol{V}) &= \|\boldsymbol{W} \odot (\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^T)\|_F^2 + \lambda \|\boldsymbol{V}\|_F^2 \\ &+ \beta \text{trace}(\boldsymbol{V}^T \boldsymbol{L}^h \boldsymbol{V}) \end{aligned} \quad (7)$$

The derivative of $J(\boldsymbol{V})$ with respect to $\boldsymbol{V}$ is

$$\begin{aligned} \frac{\partial J(\boldsymbol{V})}{\partial \boldsymbol{V}} &= -2(\boldsymbol{W} \odot \boldsymbol{Y})^T \boldsymbol{U} + 2(\boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{V}^T)^T \boldsymbol{U} \\ &+ 2\lambda \boldsymbol{V} + 2\beta \boldsymbol{L}^h \boldsymbol{V} \end{aligned} \quad (8)$$

The Karush-Kuhn-Tucker complementary condition leads to

$$[(\boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{V}^T)^T \boldsymbol{U} - (\boldsymbol{W} \odot \boldsymbol{Y})^T \boldsymbol{U} + \lambda \boldsymbol{V} + \beta \boldsymbol{L}^{(h)} \boldsymbol{V}]_{ij} \boldsymbol{V}_{ij} = 0 \quad (9)$$

We can write $\boldsymbol{L}^h = \boldsymbol{L}^{h+} - \boldsymbol{L}^{h-}$ (where $\boldsymbol{L}^{h+} = (|\boldsymbol{L}^h| + \boldsymbol{L}^h)/2$ and $\boldsymbol{L}^{h-} = (|\boldsymbol{L}^h| - \boldsymbol{L}^h)/2$) and then derive the following multiplicative updating rule:

$$\boldsymbol{V}_{ij} \longleftarrow \boldsymbol{V}_{ij} \sqrt{\frac{(\boldsymbol{W} \odot \boldsymbol{Y})^T \boldsymbol{U} + \beta \boldsymbol{L}^{h-} \boldsymbol{V}}{(\boldsymbol{W} \odot \boldsymbol{U}\boldsymbol{V}^T)^T \boldsymbol{U} + \lambda \boldsymbol{V} + \beta \boldsymbol{L}^{h+} \boldsymbol{V}}} \quad (10)$$

The problem given by (6) is simply symmetric between $\boldsymbol{U}$ and $\boldsymbol{V}$. So the derivation of the updating of $\boldsymbol{U}$, is simply reverse of the above case, and the multiplicative rule of $\boldsymbol{U}$ can be given as follows:

$$\boldsymbol{U}_{ij} \longleftarrow \boldsymbol{U}_{ij} \sqrt{\frac{(\boldsymbol{W} \odot \boldsymbol{Y})\boldsymbol{V} + \alpha(\boldsymbol{L}^{p-} \boldsymbol{U})}{(\boldsymbol{W} \odot (\boldsymbol{U}\boldsymbol{V}^T))\boldsymbol{V} + \lambda \boldsymbol{U} + \alpha \boldsymbol{L}^{p+} \boldsymbol{U}}} \quad (11)$$

TABLE I: Statistics of two datasets: Data-201706 and Data-201712.

Dataset	Data-201706	Data-201712
#proteins	3,459	3,644
#HPO terms	6,407	6,642
#leaves of HPO	4,092	4,274
#annotations	284,621	317,443
Ave. #annotations per protein	82.28	87.11
Ave. #annotations per HPO	44.42	47.79

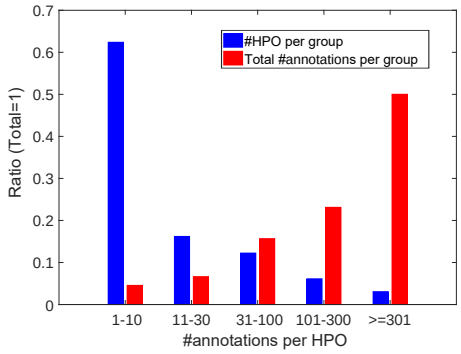

Fig. 3: HPO terms were divided into five groups by the number of annotations per HPO term. The number of HPO terms per group (left-hand side of each group) and the total number of annotations per group (right-hand side of each group) are shown from Data-201706.

6) Pseudocode and implementation: The pseudocode of our optimization process is presented in Algorithm 1. We implemented the optimization by MATLAB, using the code provided by [20].

IV. Experimental setting and results

A. Data

1) HPO annotation: We generated two matrices of HPO annotation, by using two recent versions of HPO annotation, which were downloaded from the website of HPO project: June 2017 release and December 2017 release, which we call Data-201706 and Data-201712, respectively. Table I shows the statistics of these two matrices.

We then divided HPO terms into five groups, due to the number of annotations per HPO term: 1 to 10, 11 to 30, 31 to 100, 101 to 300 and more than 300. Fig. 3 shows the two distributions of "the number of HPO terms per group" and "the total number of annotations per group" in blue and red, respectively, over the five groups from Data-201706.

2) NHPO (network of HPO): The hierarchical structure of HPO was downloaded from the website[2].

3) PPN (protein-protein network): We obtained PPN from the STRING database [21] ver 10.0a, as integrated protein-protein relationship data for human proteins. We used STRING, because the information from this database was most

[2]https://raw.githubusercontent.com/obophenotype/human-phenotype-ontology/master/hp.obo

powerful for predicting HPO annotation in [15]. STRING uses various sources, such as co-expression, co-occurrence, fusion, neighborhood, genetic interactions, physical interactions, to assign a score to each relationship, indicating the reliability. The obtained PPN for Data-201706 has 214,410 edges by 3,459 nodes (proteins), where the average number of edges per node reaches 62.

B. Evaluation Criteria

Annotation-centric measure: We use each annotation (protein-HPO term pair) as one instance and evaluate the compared methods by using Area Under the receiver operator characteristics Curve (AUC) [22]. Considering the sparseness of the (binary) annotation matrix of proteins versus HPO terms, we compute Area Under the Precision-Recall curve (AUPR) as well.

Protein-centric measure: For each protein, prediction scores by all available HPO terms are computed and sorted, and AUC (AUPR) was computed from the sorted scores. The computed AUCs (AUPRs) were averaged over all proteins, resulting in micro-AUC (micro-AUPR).

HPO Term-centric measure: We think that term-centric measure is important, since typically scientists or biologists first focus on a certain HPO term and are interested in obtaining genes/proteins which can be annotated by the focused HPO term. The HPO term-centric measure can be computed in the totally reverse manner of the protein-centric measure, having the following two steps: 1) AUC (AUPR) was first computed for each HPO term. 2) The computed AUCs (AUPRs) were averaged over all HPO terms, which resulted in macro-AUC (macro-AUPR). Additionally, we averaged the computed AUCs (AUPRs) over HPO terms at only leaves of the HPO hierarchical structure, and we call the obtained AUC (AUPR) leaf-AUC (leaf-AUPR).

We further checked macro-AUC (macro-AUPR) for each of the five groups, which were generated by focusing on the number of annotations per HPO term in Sec. IV-A1 (see Fig. 3). So entirely (from annotation-, protein-, and HPO term-centric measures) we had eight criteria to check the performance.

C. Experimental procedures

1) Parameter setting: We compared with three network-based methods: BiRW [18], DLP [9] and OGL [10], which are described in Section II. We also take Logistic Regression (LR) as a feature based baseline method which means training a collection of LR classifiers on each single HPO term, independently. Specifically, the features for LR has been constituted by PPN. In this comparative experiment, we used a grid-search for finding the optimum set of parameters.

The parameter of BiRW was selected out of $\{0.1, 0.2, ..., 0.9\}$. β and γ, regularization coefficients (i.e. hyperparameters) of DLP and OGL, were selected out of $\{10^{-6}, 10^{-5}, ..., 10^{6}\}$. We note that these parameter ranges are decided, following [9]. Our model has four parameters: K, α, β and λ, which are determined by internal five-fold cross-validation, where the training data is further randomly divided into five folds (one for parameter evaluation and the rest for training). Also in this parameter evaluation step, values are selected out of all combinations of the following values: $\{100, 200\}$ for K, $\{2^{-3}, 2^{-2}, ..., 2^{2}, 2^{3}\}$ for λ, $\{2^{-7}, 2^{-6}, ..., 2^{6}, 2^{7}\}$ for α and β.

Our model has hyperparameters, α and β, where our model was changed by modifying the values of these parameters. We evaluate each of these settings as different methods as follows:

1) **NMF**: $\alpha = 0$ and $\beta = 0$
   Our model is exactly (3), which we call NMF.
2) **NMF-PPN**: $\alpha \neq 0$ and $\beta = 0$
   Under this setting, the regularization term of NHPO is gone, while that of PPN is remained. So we call this model NMF-PPN.
3) **NMF-NHPO**: $\alpha = 0$ and $\beta \neq 0$
   This setting is reverse to NMF-PPN. That is, the regularization term of PPN is gone, while that of NHPO remains.
4) **AiProAnnotator** (**AiPA** for short): $\alpha \neq 0$ and $\beta \neq 0$
   This is our final model with the two network regularization terms. That is, both PPN and NHPO are used.

2) Two data settings: We conducted two different settings to check the performance of the compared methods from two different viewpoints:

1) Cross-validation over Data-201706
   We conducted 5 × 5-fold cross-validation over all annotations in Data-201706. That is, we repeated the following procedure five times: all known annotations are randomly divided into five, equal folds, from which four are for training and the remaining one fold is for testing. In particular, to avoid any overlap between data for training and for testing, after selecting the test annotation between protein p and HPO term h, all annotations between protein p and HPO terms, which are descendants of HPO term h in the hierarchical structure of HPO are removed from the training data. This means that we predict the annotation of protein p out of all unknown HPO terms, which is we think fair and strict evaluation.
2) Independent test using Data-201712
   HPO annotation is incomplete, due to various factors, such as slow curation. The way of annotation might be changed as time passes. So we conducted more severe experiment than regular cross-validation, using data obtained in different time period. That is, the training data were obtained before June 2017. All annotations in Data-201706 were used for training, where internal five-fold cross-validation was done for setting up parameter values, and then after this training, annotations obtained from June to December 2017 was used for testing.

D. Experimental results

1) Predictive performance in cross-validation over Data-201706: Table II shows the values of eight criteria obtained by averaging over 5×5 cross-validation (totally 25 runs) on

TABLE II: The results of eight criteria obtained by 5×5-fold cross validation over Data-201706 for totally eight competing methods.

method	AUC	AUPR	micro-AUC	micro-AUPR	macro-AUC	macro-AUPR	leaf-AUC	leaf-AUPR
LR	0.775	0.028	0.760	0.072	0.579	0.052	0.532	0.020
BiRW	0.875	0.066	0.826	0.096	0.732	0.056	0.597	0.031
OGL	0.785	0.051	0.776	0.078	0.603	0.034	0.536	0.014
DLP	0.902	0.073	0.875	0.100	0.736	0.094	0.659	0.055
NMF	0.961	0.496	0.900	0.273	0.753	0.139	0.701	0.089
NMF-PPN	0.963	0.525	0.902	0.281	0.756	0.142	0.703	0.089
NMF-NHPO	0.965	0.541	0.903	0.290	0.756	0.144	0.702	0.094
AiProAnnotator (AiPA)	**0.970**	**0.559**	**0.905**	**0.295**	**0.760**	**0.146**	**0.705**	**0.096**

TABLE III: Macro-AUC obtained by 5×5 cross-validation over Data-201706 for eight competing methods.

method	[1-10]	[11-30]	[31-100]	[101-300]	[≥301]
LR	0.526	0.553	0.633	0.735	0.755
BiRW	0.608	0.854	0.875	0.835	0.815
OGL	0.586	0.670	0.788	0.812	0.806
DLP	0.622	0.880	0.914	0.863	0.834
NMF	0.649	0.908	0.942	0.948	0.911
NMF-PPN	0.651	0.911	0.943	0.951	0.916
NMF-NHPO	0.653	0.919	**0.946**	0.947	0.919
AiPA	**0.654**	**0.922**	0.943	**0.957**	**0.931**

TABLE IV: Training time of a single run in 5 × 5 cross-validation (average over 25 runs)

method	computation time
LR	$\sim$ 3.5 hours
BiRW	$\sim$ 1.5 hours
OGL, DLP	$\geq$ 4 hours
NMF, NMF-PPN, NMF-NHPO, AiPA	$\sim$ 30 minutes

TABLE V: AUC from Independent test using Data-201712

BiRW	DLP	OGL	NMF	NMF-PPN	NMF-NHPO	AiPA
0.7971	0.8298	0.7322	0.8527	0.8923	0.8959	**0.9187**

Data-201706. In this experiment we compared totally eight methods, in which four are existing methods (LR, BiRW, OGL and DLP) and four are modifications of our own method (NMF, NMF-PPN, NMF-NHPO and AiPA). The table shows our four methods clearly better than the four existing methods. For example, in AUPR, our four methods achieved around 0.5 to 0.55, while all the values by the existing methods are less than 0.1. In fact in all eight criteria, this performance of our four methods were kept over the existing methods. Thus the difference is very clear, and we can say that low-rank approximation is useful for HPO annotation problem. Furthermore, among our four methods, AiPA outperformed other setting always in eight conditions, indicating that network information was well incorporated into our formulation.

Tables III shows AUC obtained for five groups divided by the number of annotations. Again these tables show the same conclusion as those in Table II. That is, AiPA outperformed all other methods in all cases, except only one case, where the group with the moderate number of annotations, 31 to 100. So in summary, we can say that in terms of cross-validation, our approach can achieve high performance for HPO annotation problem. In particular, one noteworthy, interesting point is our method worked well for the HPO terms with a very small number of annotations, i.e. only one to ten annotations per HPO term. In fact, this situation is usually hard for low-rank approximation, while AiPA achieved the best performance. This result indicates that low-rank approximation is useful for all types of groups including HPO terms with a very small number of annotations in the problem of HPO annotation.

2) Computation time in cross-validation over Data-201706: We checked the computation (training) time of the seven methods compared in cross-validation, where the time was averaged over the totally 25 runs (5 × 5 folds). Table IV shows computation time under the same machine setting. Our four models are faster than others, particularly being more than eight times faster than OGL and DLP. Training data is updated periodically, and so the software must be trained by the updated data often, by which this advantage would be a sizable difference. In addition to this, OGL and DLP need much more memory spaces than the other methods, which would be another serious problem.

3) Predictive performance in independent test using Data-201712: Table V shows AUC obtained by the experiment of using independent data for seven competing methods. Among the three existing methods, DLP achieved the best performance, e.g. 0.8298 by AUC. NMF outperformed DLP with AUC of 0.8527, and two modifications of NMF with one network regularizer further achieved a better performance with AUC of around 0.89. Finally AiPA gave the best performance, AUC of more than 0.9.

After the independent test, we then obtained top 30 annotations after sorting all the predicted combinations by their scores obtained by the estimated $\hat{\boldsymbol{Y}}$. Among the thirty annotations, six annotations are found in Data-201712. Table VI shows those six annotations.

For example, the top annotation in Table VI shows that gene COL7A1 (gene id: 1294), which is annotated by HPO term HP:0001072/HP:0000962 (HPO name: thickened skin/hyperkeratosis). This was not in our training data, while this appeared in the December 2017 release of HPO. Also the second annotation in Table VI shows that gene ATP6V0A2 (gene id: 23545) which encodes protein Q9Y487 is newly annotated by HPO term HP:0001263 (global developmental delay) in the recent release, while this was not in the previous

TABLE VI: Six new true annotations in top 30 annotations (by AiProAnnotator) out of all new combinations of proteins and HPO terms in Data-201706. These six annotations were not in training data but found in the latest release of HPO

rank	protein ID	protein name	gene name	HPO ID	HPO name
4	Q02388	Collagen alpha-1(VII) chain (Long-chain collagen) (LC collagen)	COL7A1	HP:0001072	Thickened skin
15	Q9Y487	V-type proton ATPase 116 kDa subunit a isoform 2 (V-ATPase 116 kDa isoform a2) (Lysosomal H(+)-transporting ATPase V0 subunit a2) (TJ6) (Vacuolar proton translocating ATPase 116 kDa subunit a isoform 2)	ATP6V0A2	HP:0001263	Global developmental delay
22	Q9H5I5	Piezo-type mechanosensitive ion channel component 2 (Protein FAM38B)	PIEZO2	HP:0000422	Abnormality of the nasal bridge
24	O43175	D-3-phosphoglycerate dehydrogenase (3-PGDH) (EC 1.1.1.95) (2-oxoglutarate reductase) (EC 1.1.1.399) (Malate dehydrogenase) (EC 1.1.1.37)	PHGDH	HP:0000366	Abnormality of the nose
25	Q02388	Collagen alpha-1(VII) chain (Long-chain collagen) (LC collagen)	COL7A1	HP:0000962	Hyperkeratosis
26	Q04656	Copper-transporting ATPase 1 (EC 3.6.3.54) (Copper pump 1) (Menkes disease-associated protein)	ATP7A	HP:0002650	Scoliosis

TABLE VII: Predicted HPO terms of P23434 (gene name: GCSH) by four our methods based on NMF. Correctly predicted HPO terms are in boldface.

method	predicted HPO terms	#correctly predicted
NMF	HP:0002079, HP:0001276, **HP:0000007**, HP:0007256, HP:0003287, **HP:0000718**, HP:0000729, HP:0002167, HP:0001268, HP:0002360	2
NMF-NHPO	**HP:0000007**, HP:0002079, **HP:0001250**, HP:0001276, **HP:0000718**, HP:0000729, HP:0012444, HP:0007256, HP:0002360, HP:0000478	3
NMF-PPN	**HP:0000007**, HP:0001276, HP:0007256, HP:0000729, **HP:0000718**, HP:0000478, HP:0003287, HP:0001268, **HP:0001298**, **HP:0001250**	4
AiPA	**HP:0001250**, **HP:0000007**, **HP:0001522**, HP:0001276, HP:0002167, **HP:0001298**, HP:0000478, **HP:0000718**, HP:0007256, HP:0012444	5
True	HP:0000007, HP:0000711, HP:0000718, HP:0001250, HP:0001298, HP:0001522, HP:0002086, HP:0002795, HP:0100247, HP:0100710	

TABLE VIII: Performance results, focusing on the subontology of organ abnormality of Data-201706. First three rows of methods with "Organ" are those trained by HPO terms on organ abnormality, while four rows with "All" are those trained by using all HPO terms.

method	AUC	AUPR	micro-AUC	micro-AUPR	macro-AUC	macro-AUPR	leaf-AUC	leaf-AUPR
NMF-Organ	0.955	0.507	0.883	0.250	0.745	0.127	0.682	0.077
NMF-PPN-Organ	0.962	0.555	0.889	**0.276**	0.755	0.144	0.701	**0.091**
NMF-NHPO-Organ	0.962	0.535	0.888	0.264	0.756	0.141	**0.702**	0.089
NMF-All	0.956	0.512	0.884	0.258	0.755	0.129	0.685	0.083
NMF-PPN-All	0.962	0.553	0.889	0.273	0.755	0.143	0.698	0.089
NMF-NHPO-All	0.962	0.556	0.889	0.274	0.755	0.144	0.699	0.090
AiPA-All	**0.963**	**0.558**	**0.891**	0.275	**0.759**	**0.145**	**0.702**	**0.091**

data. Similarly all other four cases in the table show those not in previous release of HPO and so not in training data but appeared in the current 2017 December release of HPO.

In more detail, for the top annotation of protein Q02388 by HPO HP:0001072, there are ten proteins (O43897, P07585, P08123, P08253, P12111, P20849, P20908, P25067, P53420, Q13751) which are annotated by HP:0001072 and also have the similarity score (with Q02388) of more than 0.9 in PPN. This indicates that PPN between Q02388 and those ten proteins imply a strong possibility of annotating Q02388 by HP:0001072. Similarly, for the second annotation of protein Q9Y487 by HP:0001263, there exist three proteins (O00203, O75787, P02786) which are annotated by HP:0001263 and at the same time have the similarity score (with Q9Y487) of larger than 0.9. These would be a good example of confirming that using protein-protein network information is useful for annotating proteins by unknown HPO terms.

4) Typical example showing the performance advantage of AiProAnnotator: We here show one typical example of the results obtained by our four methods, to illustrate the real performance difference in annotating proteins by HPO terms. We focus on protein P23434 (gene name: GCSH), which has true ten annotations, i.e. ten HPO terms to be annotated, shown in the bottom row of Table VII. Other rows of Table VII show the top ten HPO terms predicted by our four methods to annotate P23434. The most right-hand column of this table shows the number of correctly predicted HPO terms out of the true ten HPO terms. Interestingly the correctly predicted number was incrementally increased, starting with two by NMF, three by NMF-NHPO, four by NMF-PPN and finally five by AiPA. This result also indicates that using network information is useful for improving the performance of annotating proteins by HPO terms.

5) Performance comparison, focusing on organ abnormality: Most of the past work on annotating proteins by HPO terms show the performance in each subontology of HPO. However, focusing on part of data will lose the advantage of using the entire network information which can connect proteins or HPO terms even beyond the boundary of two or more subontology in the network space. Thus we avoided conducting experiments in each of all subontologies, and instead we focused on the most major subontology, organ abnormality (HP:0000118), with 6,370 HPO terms, 3,446 proteins and totally 269,420 annotations. We conducted 5×5 cross validation on the subontology of organ abnormality, using the same split for experiments over Data-201706. Table VIII

shows the values of eight evaluation criteria obtained by all compared methods, i.e. totally seven cases of our models. The table shows that the performance difference among the seven cases was very slight. For example, AiProAnnotator using all data achieved the best performance for all cases except one, and in AUC, AiProAnnotator was 0.963, being followed by four other cases with 0.962. From this result, we can imagine that network information is useful but using both networks, i.e. PPN and NHPO, might not be so useful for this case. Also organ abnormality is the major subontology, by which using all training data seems not so effective for improving the performance obtained by using organ abnormality only.

V. Conclusion and discussion

We have presented to use low-rank approximation to the problem of large-scale annotation of human proteins. Also we have proposed to use network information, which can be derived from the both sides of annotation, i.e. protein-protein network, and the hierarchical structure of the ontology side. In particular, we have formulated the low-rank approximation into the optimization problem of matrix factorization with network-derived regularization. We then have empirically examined the effectiveness of our approach by using the current HPO database. Experimental results clearly show the performance advantage of the proposed method under various settings, including cross-validation, independent test, focusing on the major subontology of organ abnormality, and detailed case studies, etc. In particular, our approach of using matrix factorization or more generally low-rank approximation was effective to improve the performance of annotation, even for the group of HPO terms with a very small number of annotations. These results indicate the validity of using low-rank approximation and also network information regularize the approximation for the problem of annotating human proteins by ontology with the hierarchical structure.

Important findings on our approach are: 1) low-rank approximation works very well for large-scale HPO annotation or more generally, multilabel classification even for predicting labels with an extremely small number of instances, i.e. proteins, at least for HPO annotation, 2) protein-protein network and hierarchical ontology structure were very helpful as side information for improving the performance of low-rank approximation, and 3) multiplicative parameter updating of low-rank approximation (matrix factorization) was time-efficient, particularly around eight times faster than network-based approaches, which use the original annotation matrices directly and so need huge memory spaces also.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 61572139 and 61872094).

References

[1] R. You, Z. Zhang, Y. Xiong, F. Sun, H. Mamitsuka, and S. Zhu, "Golabeler: Improving sequence-based large-scale protein function prediction by learning to rank," *Bioinformatics*, vol. 34, no. 14, pp. 2465–2473, 2018.

[2] N. Freimer and C. Sabatti, "The human phenome project," *Nature genetics*, vol. 34, no. 1, pp. 15–21, 2003.

[3] S. Köhler, S. C. Doelken, C. J. Mungall, S. Bauer, H. V. Firth, I. Bailleul-Forestier, G. C. Black, D. L. Brown, M. Brudno, J. Campbell *et al.*, "The human phenotype ontology project: linking molecular biology and disease through phenotype data," *Nucleic acids research*, vol. 42, no. D1, pp. D966–D974, 2013.

[4] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, "Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D514–D517, 2005.

[5] S. Aymé and J. Schmidtke, "Networking for rare diseases: a necessity for europe," *Bundesgesundheitsblatt-Gesundheitsforschung-Gesundheitsschutz*, vol. 50, no. 12, pp. 1477–1483, 2007.

[6] E. Bragin, E. A. Chatzimichali, C. F. Wright, M. E. Hurles, H. V. Firth, A. P. Bevan, and G. J. Swaminathan, "Decipher: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation," *Nucleic acids research*, vol. 42, no. D1, pp. D993–D1000, 2013.

[7] M. Xie, T. Hwang, and R. Kuang, "Reconstructing disease phenome-genome association by bi-random walk," *Bioinformatics*, vol. 1, no. 02, pp. 1–8, 2012.

[8] P. Wang, W.-F. Lai, M. J. Li, F. Xu, H. K. Yalamanchili, R. Lovell-Badge, and J. Wang, "Inference of gene-phenotype associations via protein-protein interaction and orthology," *PloS one*, vol. 8, no. 10, p. e77478, 2013.

[9] R. Petegrosso, S. Park, T. H. Hwang, and R. Kuang, "Transfer learning across ontologies for phenome–genome association prediction," *Bioinformatics*, vol. 33, no. 4, pp. 529–536, 2016.

[10] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," *Journal of Machine Learning Research*, 2010.

[11] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.

[12] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[13] J. J.-Y. Wang, X. Wang, and X. Gao, "Non-negative matrix factorization by maximizing correntropy for cancer clustering," *BMC bioinformatics*, vol. 14, no. 1, p. 107, 2013.

[14] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based stratification of tumor mutations," *Nature methods*, vol. 10, no. 11, pp. 1108–1115, 2013.

[15] I. Kahanda, C. Funk, K. Verspoor, and A. Ben-Hur, "Phenostruct: Prediction of human phenotype ontology terms using heterogeneous data sources," *F1000Research*, vol. 4, 2015.

[16] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC bioinformatics*, vol. 11, no. 1, p. 2, 2010.

[17] M. Xie, T. Hwang, and R. Kuang, "Prioritizing disease genes by bi-random walk," *Advances in Knowledge Discovery and Data Mining*, pp. 292–303, 2012.

[18] M. Xie, Y. Xu, Y. Zhang, T. Hwang, and R. Kuang, "Network-based phenome-genome association prediction by bi-random walk," *PloS one*, vol. 10, no. 5, p. e0125138, 2015.

[19] D. Lin *et al.*, "An information-theoretic definition of similarity." in *International Conference on Machine Learning*, vol. 98, no. 1998, 1998, pp. 296–304.

[20] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[21] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork *et al.*, "The string database in 2011: functional interaction networks of proteins, globally integrated and scored," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D561–D568, 2010.

[22] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," *International Conference on Machine Learning*, pp. 3780–3788, 2017.