Protease target prediction via matrix factorization

Simone Marini[1,*,eq.], Francesca Vitali[2,eq.], Sara Rampazzi[3], Andrea Demartini[4], and Tatsuya Akutsu[5,*]

[1] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA, [2](Center for Biomedical Informatics and Biostatistics, BIO5 Institute, Department of Medicine), University of Arizona, Tucson, [3]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA, [4]Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Pavia, Italy, [5]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto, Japan.

* To whom correspondence should be addressed.

eq. authors equally contributed

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Protein cleavage is an important cellular event, involved in a myriad of processes, from apoptosis to immune response. Bioinformatics provides in silico tools, such as machine learning-based models, to guide the discovery of targets of the protease responsible for protein cleavage. State-of-the-art models have a scope limited to specific protease families (such as Caspases), and do not explicitly include biological or medical knowledge (such as the hierarchical protein domain similarity, or gene-gene interactions). To fill this gap, we present a novel approach for protease target prediction based on data integration.

Results: By representing protease-protein target information in the form of relational matrices, we design a model that: (a) is general, i.e., not limited to a single protease family; and (b) leverages on the available knowledge, managing extremely sparse data from heterogeneous data sources, including primary sequence, pathways, domains, and interactions. When compared to other algorithms on test data, our approach provides a better performance even for models specifically focusing on a single protease family.

Availability: https://gitlab.com/smarini/MaDDA/ (Matlab code and utilized data.)

Contact: smarini@med.umich.edu, or takutsu@kuicr.kyoto-u.ac.jp

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Protein cleavage is a pivotal process in cell metabolism of both cellular and extracellular matrix. Among other processes, protein cleavage is involved in cell differentiation and cycle control, stress and immune response, removal of abnormally folded proteins and cell death (Rawlings *et al.*, 2016). The proteins responsible for cleavage, i.e. the proteases, account for about 2% of all gene products (James, 1999). As a consequence, wrongly regulated proteolytic activity may result in diseases (Rawlings *et al.*, 2016; Wang *et al.*, 2014). Caspase activity, for example, plays a role in Alzhemier's Disease (Chu *et al.*, 2015; Zhao *et al.*, 2016), Autoimmune lymphoproliferative syndrome (Lenardo *et al.*, 1999), and cancer (Oh *et al.*, 2010; Hosgood *et al.*, 2008; Lee *et al.*, 2009).Several computational methods have been proposed to tackle the key-lock machinery of the protease-protein target recognition (Song *et al.*, 2010; Wang *et al.*, 2014; Boyd *et al.*, 2005; Song *et al.*, 2012; Wilkins *et al.*, 1999; Barkan *et al.*, 2010; Singh and Su, 2016; Bao *et al.*, 2018). Cleavage target models, for example, aim at extracting sequence patterns or frequency matrices from known protein target structures/primary sequences in order to predict the likely cleavage points for proteases to act. Design of such algorithms goes back to more than a decade, confirming protease target prediction as a main research focus in Bioinformatics. A basic but effective approach to predict the targets of a certain protease consists of a simple BLAST sequence search (Wang *et al.*, 2014). The BLAST approach relies on the following assumption: if it is known that a protease P_p cleaves a protein target P_x, then the more a candidate target P_y shares primary sequence similarity to P_x, the more likely P_y will be a target as well. Therefore, in order to infer new targets, researchers can score a candidate protein P_y against its similarity (i.e., the BLAST eValue) with the target proteins $[P_x, \ldots, P_X]$ of a known protease P_p. However, even if there is a strong BLAST similarity between P_x, known target of P_p, and the query protein P_y, this does not guarantee that P_y will be a target of P_p as well. In fact, the cleavage mechanism does not

depend on general protein similarity, but on the similarity of specific primary structure patches, influenced by sequential and spatial patterns of specific amino acids in key positions (Song *et al.*, 2012; James, 1999). In other words, even if two proteins share a very high sequence similarity overall, the cleavage dichotomy between target and non-target might be due by a difference in only a few amino acids pivotal positions of the sequence. For example, the peptidase thrombin cleaves the target if an arginine is present next to the scissile bond, in N-terminal direction, but only if an aspartate or a glutamate is *not* juxtaposed to it in the carboxyl-terminal direction. The pioneering efforts in cleavage target prediction were based on the analysis of primary sequence only, for example with PeptideCutter (Wilkins *et al.*, 1999) and PoPS (Boyd *et al.*, 2005). The peculiarity of these methods, differentiating them from generic protein interaction prediction algorithms (Marini *et al.*, 2011; Planas-Iglesias *et al.*, 2013), is the embedding of specific protease knowledge. For example, PoPS searches for candidate cleavage spots through the primary sequence with a short sliding window, considering both cleavage-specific position-specific scoring matrix (PSSM) and cleavage-specific weight vectors. On the other hand, PeptideCutter predicts cleavage sites by applying enzyme-specific rules exploiting amino acid-specific cleavage probability tables. More recently, novel machine learning techniques have been applied to solve the cleavage target prediction problem. These approaches are mostly based on support vector machines (SVMs) and trained on complex protein features, such as structural and physicochemical features, including solvent accessibility or disordered regions (Song *et al.*, 2012; Barkan *et al.*, 2010; Song *et al.*, 2010; Wang *et al.*, 2014). Despite improving the quality of the predictions, these recent algorithms still suffer from two major limitations. First of all, they are specific for single proteases, i.e. each SVM model is trained to predict the cleavage protein target of a specific protease. While the known human proteases are hundreds (Rawlings *et al.*, 2016), prediction algorithms end up focusing on the most known ones, such as the ones belonging to the Caspase family, or the HIV-1 protease (Singh and Su, 2016). Therefore, these models are protease-specific, and lack of generalization, i.e. a different model needs to be trained for each protease. This led to a well-known bias toward "superstar" proteases, leaving most of other proteases orphan of adequate prediction models. Note that the problem encompasses not just proteases-protein target prediction, but Proteomics prediction in general (Orlowski *et al.*, 2007; Lam *et al.*, 2016). Furthermore, state-of-the-art algorithms do not leverage the implicit knowledge crosslinks available from biological and medical ontologies and databases, even if trained on fine-grained structural information. For example, Cascleave2 (Wang *et al.*, 2014) exploits Gene Ontology (Gene Ontology Consortium, 2015), InterPro (Finn *et al.*, 2017), and KEGG (Kanehisa *et al.*, 2017), respectively for GO term, domain, and pathway information. However, these data are elaborated and converted into numerical attributes describing single samples, and then fed to SVMs. With this traditional feature encoding approach, other *indirect*, although relevant, knowledge is not embedded in the prediction model. Examples of ontological relationships not encoded as features are: the domain interactions; the hierarchical relationships between domains and genes; or the overlapping of the same gene across different pathways. Intuitively, it is possible to numerically link the instances of different data sources with *interacts-with* (e.g. protein P_x interacts with protein P_y) or *is-part-of* (e.g. domain D is found in protein P_x) type of relations, as was done, for example, in the hierarchically structured Gene Ontology. Biological knowledge-bases are indeed replete with relational knowledge, and yet this knowledge is not exploited in a typical feature encoding. For example, let's assume that the single instance of a prediction model is a gene. We could list proteins from UniProt as gene features by mapping in a binary fashion the proteins encoded by gene. In addition, we know that proteins interact, and we can track their interactions with, for example, the STRING (Szklarczyk et al., 2017) database. In this way, we can also associate to each interaction a numeric value, i.e., the STRING confidence score. However, we might not include these scores directly as gene features. In other words, knowledge about the feature relations (the protein interactions), coming from a second data source (STRING) could not be explicitly included in the model.

Here we present an approach to the prediction of protease-to-protein target interactions, overcoming these limits, and provide a general model for protease target prediction. Note that our model does not predict the cleavage sites, but scores the probability of a protein to be a feasible target for a given protease. Our approach is based on the representation of different data sources through matrices allowing to directly integrate ontologies and other knowledge sources in the learning model. In other words, our approach accounts for multiple, heterogeneous data sources, without altering the knowledge data structure. For cleavage prediction we exploited the repositories MEROPS, STRING,, Interpro, Domine (Raghavachari *et al.*, 2008), 3did (Mosca *et al.*, 2014), UNIProt, BioGRID (Chatr-aryamontri *et al.*, 2017), and KEGG. Matrix decomposition-based methods demonstrated their power and versatility by being used for clustering, discovery, prioritization, and classification, for example in the prediction of disease subtypes alignment (Gligorijević et al., 2016), drug repositioning (Vitali et al., 2016), and estimating patient similarity (Vitali et al., 2018). The main advantage of this method, compared to other data integration techniques, is the preservation of the original data structure through the block matrices. This has been shown to enhance the interpretability of the data integrated (Vitali et al., 2018). On the other hand, the tri-factorization approach requires the definition of initial parameters (e.g., ranks), and might have a high computational demand due to the operations with large, sparse matrices. In this work we apply a data integration method based on non-negative matrix tri-factorization (Žitnik and Zupan, 2015) to predict protein targets for human proteases. Our approach is a *general-purpose model*, and the results confirmed its ability in providing predictions not limited to specific, well-studied proteases. We further show the proposed method outperforms five existing approaches in terms of both range of application and results (ROC area).

2 Methods

The algorithm we utilized is a variant of the non-negative matrix tri-factorization data integration approach (Vitali *et al.*, 2018, 2016) the main difference consisting in cross-validating the training set in order to find the optimal parameter permutation. Whereas in other works factorization ranks have been fixed (Vitali *et al.*, 2018, 2016), or selected by optimization, e.g., with the use of a score such as the cophenetic index (Žitnik and Zupan, 2015), in our work we cross-validated sparseness-related scaling factors. Note that we also cross-validated the very thresholds utilized for classification. In this way, the matrix factorization process itself becomes a supervised classification process (see Section 2.7, and Supplementary data). Cross-validation was chosen in preference to other methods (such as bootstrap) to reduce the computational burden. Relations between proteases, protein targets, genes, pathways and domains, harvested from knowledge databases, are first represented by different matrices. All the resulting matrices are then combined into a single block matrix. An iterative algorithm, guided by relational constraints, decomposes the original matrix into three smaller ones. Novel relations (i.e. novel protease-protein targets) are finally inferred by comparing the original and the reconstructed matrices.

2.1 Feature encoding by relational matrices

The basic assumption of tri-factorization is that every data instance (i.e. a relation) is represented by the value of a relational matrix cell. Here, rows and columns are sets of *data types*, such as proteases, target proteins, or genes. For example, given a set of N proteases encoded by M genes, the fact that a protease P_n is encoded by gene G_m is indicated by the value "1" of the cell corresponding to P_n row and G_m column of a $N \times M$ matrix. In this work, we considered five data types: *proteases*, *protein targets*, *genes*, *domains* and *pathways*. All the relationships between the elements of these data types are represented through two types of relational matrices: the *input matrices* R_{ij}, and the *constraint matrices* θ_i. The *input matrices* $R_{ij} \in R^{n_i \times n_j}$, represent different data types relations (e.g. protease-gene); while the *constraint matrices* $\theta_i \in R^{n_i \times n_i}$ represent relations among data of the same type (e.g. protease-protease). Figure 1 depicts the specific R and θ matrices utilized in this work.

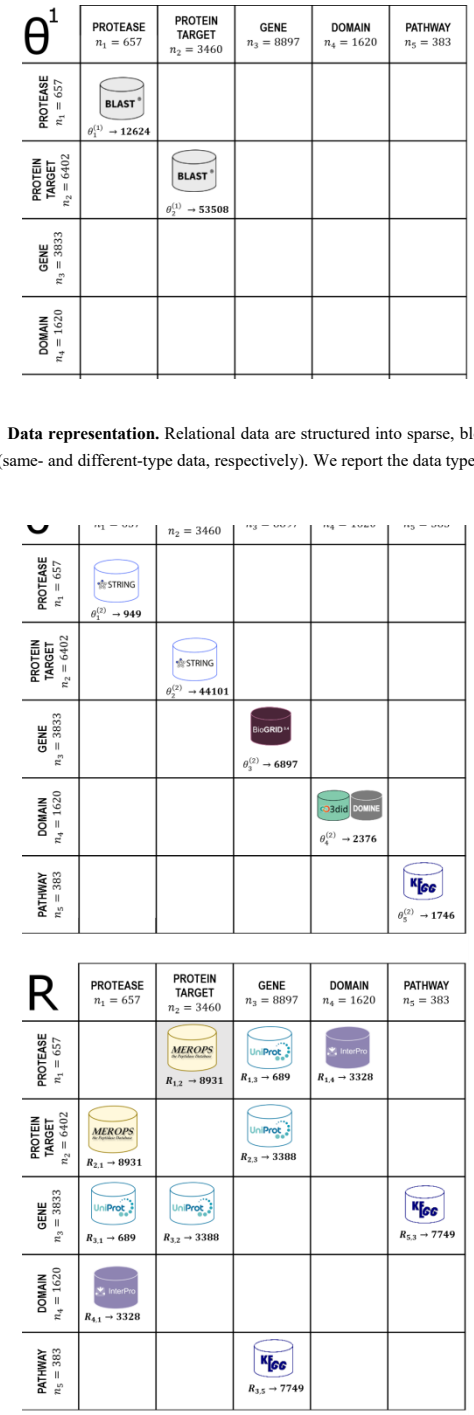


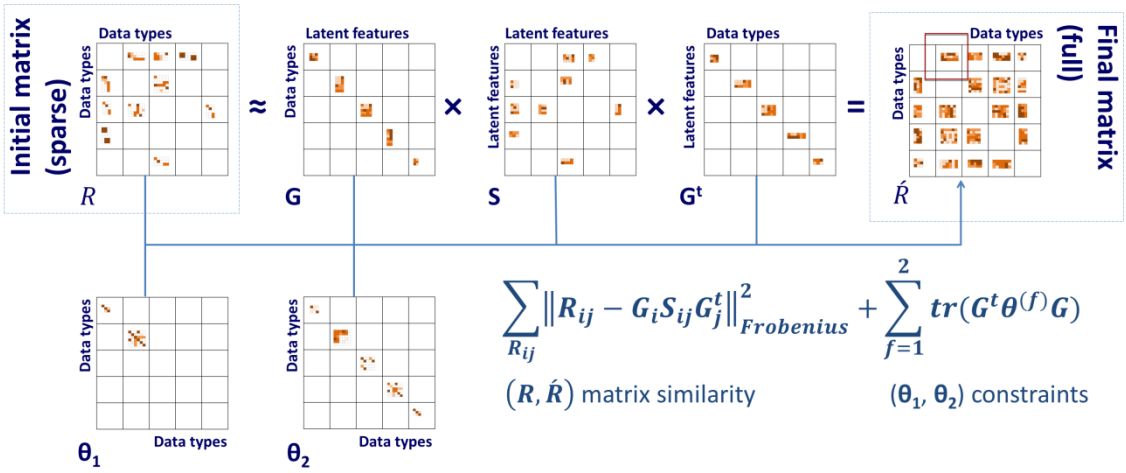
Fig. 1. Data representation. Relational data are structured into sparse, block matrices R and $\boldsymbol{\theta}$ (same- and different-type data, respectively). We report the data types and sources.

2.2 R matrices: different data-type relations

Input matrices R_{ij} are then combined into a block matrix $R \in R^{N \times N}$, with $N = \sum_i \quad n_i$, containing all the available associations of r data types:

$$R = [0\ R_{12} \cdots R_{1r}\ R_{21}\ 0 \cdots R_{2r} \vdots \vdots \ddots \vdots\ R_{r1}\ R_{r2} \cdots 0\ ]$$

This matrix representation does not compress or alter the original structure of the data, which are simply juxtaposed in a block matrix. Note that R is a hollow matrix, being composed of input matrices only (i.e. same data types relations are not considered). Blocks on the diagonal are empty. Also note that some blocks can be null due to missing (or meaningless) associations between the *i-th* and *j-th* data types. Furthermore, R is typically very sparse, since the known relations are just a small fraction of all the possible combinations of data types. Values of input matrices must be bound to the [0, 1] interval, where 1 indicates the strongest association, while 0 represents the absence of relation or the lack of knowledge about it.



2.3 θ matrices: same data-type relations (constraints).

θ_i matrices account for the data constraints in the tri-factorization algorithm, and record the associations between elements of the same kind (such as protease-protease or domain-domain relations). Multiple θs can describe constraint for the same data type, according to the following schema:

$$\theta^{(f)} = \begin{bmatrix} \theta_1^{(f)} & 0 & \cdots & 0 \\ 0 & \theta_2^{(f)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \theta_r^{(f)} \end{bmatrix}$$

For example, in order to model the protein-protein relationships, we can use both a primary sequence similarity from BLAST and protein interactions described in STRING. In this way, we can derive two constraint (θs) matrices to be used simultaneously. Being f the maximum cardinality of the θ_i matrices, we define f block diagonal $\theta^{(f)}$ matrices, with the same size of R. Each $\theta_i^{(f)}$ (if existing) is then the *i-th* block on a diagonal, and r represents a data type or ontological entity, e.g., proteins, genes, pathways, etc. Constraint matrices are bound in the [-1,1] interval, where -1 indicates the strongest association, and 1 represents a negative association. We therefore can treat the negative values of θs as must-link constraints; reversely, positive values of θs indicate cannot-link constraints.

2.4 Data set.

A first list of 657 human proteases cleaving 3460 human protein targets in 8931 interactions substantiated by experimental evidence has been obtained from MEROPS database. The protease-protein pair interactions represent the positive samples, while the non-interacting pairs represent the negative ones. STRING provided protein interactions data (threshold 0.7) for protease-protease and target-target associations, respectively 949 and 44410 pairs. Sequence similarity was measured with BLAST, and filtered using a 10^{-10} e-value threshold. This procedure produced 12624 and 53508 elements for the protease-protease and protein target-protein target matrices, respectively. An InterPro analysis revealed 4723 domains expressed on 3328 protease-domain relations, and 13368 protein target-domain targets. 2376 domain-domain interactions were retrieved in Domine and 3did. Both proteases and protein targets were mapped with UNIProt on their 3833 coding genes, as 689 protease-gene and 3388 protein target-gene relations emerged. Genes form 6897 interacting pairs were retrieved on the BioGRID database. The genes expressing proteases and their targets map 7749 gene-pathway relations, as they are involved in 290 KEGG pathways. Pathways, in turn, form 1746 pathway-pathway relations. The assembled R and θ matrices are depicted in Figure 1.

2.5 Feature encoding by relational matrices

Both R and θ are characterized by very high sparseness due to the fact that biological interactions are typically a tiny fraction of all potential interactions (Gilchrist *et al.*, 2004; Orlowski *et al.*, 2007). A target matrix R_t is selected among the block matrices. It will be reconstructed through tri-factorization into R'_t. The final goal is to learn the novel interactions, i.e. filling the gaps in R_t to unveil novel relations between the two specific data types of R_t. The newly learned relations are to be found in the *dissimilarities* between R_t and R'_t. In our application, these data types are *proteases* and *protein targets*. This is obtained by factorizing the starting R matrix into the product of three terms. Each R_{ij} block is tri-factorized: $R_{ij} \approx G_i S_{ij} G_j^t$. Here, $G \in R^{N \times K}$ is a non-negative block diagonal matrix, and block $G_i \in R^{n_i x k_i}$ represents the *i-th* data type, while $K = \sum_i \quad k_i$. K_i terms refer to the ranks, defining the dimension of the latent factors for the *i-th* data type. They are typically orders of magnitude smaller than the associated N_i dimension (Žitnik and Zupan, 2015; Vitali *et al.*, 2016). $S \in R^{K \times K}$ is a squared block matrix with null blocks on the main diagonal, and for all the $S_{ij} \in R^{k_i \times k_j}$ blocks where corresponding R_{ij} is null. S models the associations between the latent factors. In particular, each S_{ij} block relates the latent features of the *i-th* type of data type with the latent features of the *j-th* data type. In practice, S_{ij} represents a compressed version of the related Rij the space of the latent factors. The objective function J to be optimized is:

Fig. 2. Data are embedded into three relational, sparse, same-size matrices: R, θ_1 and θ_2. Each matrix is composed by concatenating smaller matrices describing the relation between data types (protease, protein targets, pathways, domains, and genes). R is tri-factorized, and recomposed in $\acute{R}$ by an iterative algorithm. This process both minimizes the dissimilarity between R and $\acute{R}$ (measured through the Frobenius norm) and forces $\acute{R}$ to be compliant with the constraints imposed by θ_1 and θ_2. Novel associations between two data types are found in the differences between R and $\acute{R}$in specific block matrices, here indicated by the red square on $\acute{R}$.

$$J(G;S) = \sum_{R_{ij}} \left\|R_{ij} - G_i S_{ij} G_j^t\right\|_{Fr}^2 + \sum_{f=1}^{F} tr(G^t \theta^{(f)} G)$$

The first part of the objective function, $\sum_{R_{ij}} \left\|R_{ij} - G_i S_{ij} G_j^t\right\|_{Fr}^2$, is the Frobenius norm of the difference between the original matrix and the tri-factorized one, and the second part of the objective function depends on the constraint matrices $\theta_i^{(f)}$. Note that without this second part, the objective function would simply lead to a tri-factorization without the contribution of the θs. The second term, $\sum_{f=1}^{F} tr(G^t \theta^{(f)} G)$, penalizes the objective function according to the must-link and cannot-link values of the constraint matrices. Note that F represents the maximum multiplicity of θ, and, in our case F=2 (Figure 1). The whole tri-factoriazion process is carried out through an iterative process detailed in the Supplementary data. In summary, after initialization of the G_i factors, an alternate optimization of G and S is performed. Keeping G fixed, S is updated, and then, keeping S fixed, G is updated. The update rules for the two types of matrices are obtained by computing the roots and the partial derivative of J, fixing the other matrix. Factorization ranks K_i determine the number of columns in G matrices, one per object type. Fixing the size of G, in other words, is a crucial factor for the data integration approach, as it determines the number of latent features condensing the information of each object type. (As reported in the Section *2.7*, we inferred the best factorization ranks according to a grid search cross-validation.) Stopping criteria is determined by either a maximum number of iterations, or two consecutive values J_i,J_{i+1} such that $J_i - J_{i+1} < T$, where T is a fixed threshold (i.e., the variation between two iterations is small enough to consider the algorithm plateaued). Figure 2 depicts the core idea of feature representation and matrix reconstruction through tri-factorization.

2.6 Inferring novel relations

Once R'_t has been produced by the algorithm, a rule is needed to infer which of the newly non-zero R'_t elements are to be considered newly predicted relations, i.e. predicted protease-protein target pairs. (Note that the matrix-assembling rule binding R_t to the [0, 1] interval does not apply to the tri-factorized R'_t.) R'_t is used to compute a connectivity matrix $Conn_m$, a binary matrix derived from R'_t where elements to be considered (*predicted* or *previously known*) relations are 1s, and unconnected elements are 0s. To set a row *r* and column *c* element E_{rc} of R'_t as a newly predicted relation (a 1 in $Conn_m$), its value should be higher than the average value of non-zero elements in a row *r* or column *c* of original target block matrix R_t, as illustrated in Figure 3. Note that, since R_t here represents the protease-protein target relations, it is a binary matrix (cleavages are either present or absent/unknown, i.e., a protein is either a target or a not for each give protease). As a consequence, the average value of R_t non-zero elements is always 1. Since the initialization of the algorithm is randomized, *nr* runs the algorithm will produce *nr* different $Conn_m$ matrices. By summing all $Conn_m$s and dividing by the number of runs *nr,* a final Consensus Matrix C_m is obtained, as the final output of the approach. Note that since the rule to populate $Conn_m$ can be based on rows or columns, it is possible to build both $C_{m,row}$ ad $C_{m,col}$.

$$C_m = \frac{\sum_{m=1}^{nr} Conn_m}{nr}$$

In other words, given a target $R_t \in R^{n_i \times n_j}$, $C_m \in Z^{n_i \times n_j}$ has elements bound to the [0, 1] interval. Each $C_{m(i,j)}$element reflects the number of times that relation has been predicted as positive in the $Conn_m$s. For example, a relation $C_{m(i,j)}$ predicted as positive in half of the runs will have a value of 0.5; a relation never predicted as positive will have a value of 0; and a relation always predicted as positive will have a value of 1. Note that the output of our approach will therefore be a number in the interval [0, 1], denoting the likeliness or the absence/presence of a relation between the two elements of the target matrix. In our case, this relation represents the likeliness of a protein to be a target for a specific protease. As a consequence, only the cleavage absence/presence is provided, and not the cleavage point. Not having an indication about the possible location of cleavage positions is the major limitation of our approach**.**

2.7 Parameter tuning

We randomly split all the possible protease-protein target pairs into a training (85%) and test (15%) set, stratified on class. The considered parameters for this tri-factorization approach are (a) the stop criteria; (b) the number of runs *nr* to build C_m; (c) the initialization; (d) the threshold(s) on C_m to call for a newly predicted relation; and (e) the factorization ranks k_i. There is no clear consensus in literature about how to select these values (Žitnik and Zupan, 2015; Gligorijević *et al.*, 2016). We therefore opted for an empirical grid search by cross-validating the training set, maximizing Matthews Correlation Coefficient (MCC), a measure of performance of unbalanced class data sets such as ours. In order to ease the computational burden of the whole process, we fixed: the stopping criteria (a) by setting $T = 10^{-5}$, according to literature (Žitnik and Zupan, 2015), and the maximum number of iterations to 10 thousand; the total number of runs for a single fold (b) was set to 5; initialization of G and S from random [0, 1] uniform distribution (c). Other parameters were optimized with a grid search. In particular, we set a double threshold to isolate newly predicted relations (d), searched among five thresholds [0.2, 0.4, 0.6, 0.8, 1] on both $C_{m,row}$ and $C_{m,col}$; factorization ranks were searched independently for each data type as $k_i = \frac{N_i}{l}$, where N_i is the total number of non-zero elements of a data type i, the overall associations

modeled by all the related relation matrices R_{ij} and $R_{ji} \forall j$; while l is a scaling factor to be optimized in the interval [50, 100, 250, 500, 750, 1000]. Note that each data type i was independently considered for its scaling factor l. The grid search was performed with a 5-fold cross validation on the training set, measuring all the possible combination of parameters (d) and (e). The values considered for the parameters are reported in Section 2 of the Supplementary data, along with a description of the grid search.

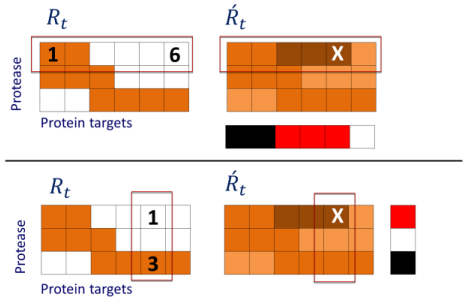


Fig. 3. Discovering novel targets. This picture shows a toy example of the steps used to build a connectivity matrix $Conn_m$. Target reconstructed matrix $\acute{R}_t$ is compared the original matrix R_t by considering rows or columns. In R_t, elements (targets) are either present (1, orange), or absent/unknown (0, white). In $\acute{R}_t$, elements can assume values higher (brown) or lower (light orange) than one. *Row criteria (top panel.)* If we consider the first row of R_t, novel associations could appear in elements 3, 4, 5 and 6, as elements 1 and 2 already indicate an existing association. Elements 3, 4 and 5 all show a value higher than 1, and can be considered as potential new target according to the row criteria. *Column criteria (bottom panel).* Let's now show novel target from the column point of view. Considering the fifth column, we have one known interaction (element 3, R_t). In the fifth column of $\acute{R}_t$, only element 1 shows a value higher than 1, and can be considered as a potential new target according to the column criteria. This element (marked by *x*) fulfills both its row and column rules, i.e. the putative new association is considered true for connectivity matrix $Conn_m$ if it satisfies both criteria.

3 Results

After the grid search optimization on the training set, the best parameter combination was l=500 for domains, genes, pathways, and proteases; and l=250 for protein targets. The thresholds for the Consensus Matrices were 0.2 for columns, and 0.2 for rows. We ran our algorithm 10 times on the test set in order to compute the Consensus Matrices. On the test set, our approach with these parameters provided Specificity 0.999, Precision 0.88, ROC area 0.79, and MCC 0.446.

3.1 Comparison with other algorithms

We measured the performance of the proposed method against five well known other approaches: a BLAST search, Cascleave 2.0, PROSPER, PeptideCutter and PCSS. PoPS was considered as well, but we could not find a suitable implementation. Besides BLAST, all other algorithms have limited scope in terms of protease-target protein when compared to our more general approach. We therefore limited our comparison to the test set parts overlapping protease-protein target pairs predictable by the different algorithms. Table 1 reports details about these algorithm-specific test sets. Note that while our approach outputs a dichotomy (i.e., the protein is or is not a target for a given protease), these algorithms provide multiple cleavage scores (i.e., the likely cleavage points along the primary structure of a target). Therefore, for comparison, we retained the highest cleavage score, rescaled it in the [0, 1] interval, and used the resulting scores to compute the ROC area. We run the models for BLAST, PROSPER and PeptideCutter; for Cascleave 2 and PCSS, we utilized the pre-computed scores provided by their respective websites. Some of the tested protease-protein target pairs, therefore, have very likely been included in the training sets of other algorithms. This condition is unfavorable for our approach, and implies results represent a best case scenario (i.e., an upper bound) for the predictions provided by other models. Our approach outperforms other algorithms, averaging a 0.178 higher ROC area. As a consequence, our model seems to generalize better, to the point of providing better results even when compared to the Caspase-specific sets of Cascleave 2.0. Results are detailed in Table 1 and in Figure 4.

Table 1. Method comparison

Subset	# of overlapping test-set pairs	ROC area	
		Other algorithm	Our method
BLAST	340983 (all)	0.65±0.001	0.79±0.0009
Cascleave	2377	0.63±0.0132	0.74±0.0118
PROSPER	3574	0.51±0.0112	0.77±0.0091
PeptideCutter	1308	0.26±0.0158	0.56±0.0184
PCSS	2279	0.66±0.0131	0.73±0.0121

Our approach outperforms other algorithms in the test subsets. Note that our algorithm is the only approach capable to provide predictions for all the available proteases. The subsets are obtained by considering the protease-protein target pairs of our vast test set that each algorithm is able to classify.

3.2 Inferring novel targets

After the comparison with other algorithms, we retained the best parameter set and applied our method not to the test set only, but to the whole protease-protein target matrix. In other words, we applied our method to all the collected data in order to infer novel protease-protein target relations, using 10 repetitions to build C_m (Section 2.6). We manually analyzed the best 54 scoring pairs, i.e., the ones fulfilling both row and column criteria 10 out 10 times. Twenty-five proteases are involved: one Cathepsin, two Calpains, two Caspases, thirteen Metalloendopeptidases, and seven Serine proteases. We found evidence of at least one pair confirmed as a real cleavage reported in literature, but not yet present in our MEROPS-derived data set, for all families. In particular, we found ten literature-confirmed cleavages: Cathepsin-D, Calpain-1, and Calpain-2 cleaving Angiotensiongen (Andrés, 2014; Jiang *et al.*, 2008); Caspase-7 cleaving SREBF1 (Gibot *et al.*, 2009); Matrix metallopeptidase-9 cleaving Decorin (Yang *et al.*, 2014), CTGF (Hashimoto *et al.*, 2002), and Prolargin (Zhen *et al.*, 2008); t-plasminogen activator cleaving Complement C4-A (Barthel *et al.*, 2012); ADAM28

cleaving FCER2 (Okada, 2017, 8); and Neprilisin-2 cleaving Neurotensin (Skidgel and Erdös, 2004). For all other pairs, we found (a) that the target is cleaved by another protease of the same family of the predicted one, or other targets belonging to same family are cleaved by the protease; and/or (b) that both the protease and the predicted target are involved in the same disease model, or the cleavage was inferred in a quantitative/qualitative mechanism, suggesting the plausibility of the cleavage event. These findings are summarized in Table 2, and provided in detail in the supplementary information.

Table 2. Candidate novel targets for proteases.

Protease (Merops ID)	Confirmed	Same family	Shared mechanism
Cathepsin-D (A01.009)	1	-	1
Calpain-1 (C02.001)	1	-	5
Calpain-2 (C02.002)	1	-	2
Caspase-1 (C14.001)	-	1	-
Caspsase-7 (C14.004)	1	2	-
MMP1 (M10.001)	-	4	-
MMP-8 (M10.002)	-	1	-
MMP-9 (M10.004)	3	4	-
MMP-3 (M10.005)	-	2	-
MMP-7 (M10.008)	-	2	-
MMP-13 (M10.013)	-	1	-
MMP-14 (M10.014)	-	3	-
ADAMTS4 (M12.221)	-	1	-
ADAMTS1 (M12.222)	-	1	-
ADAM28 (M12.224)	1	2	-
ADAMTS2 (M12.301)	-	1	-
ECE-1 (M13.002)	-	-	1
Neprilysin-2 (M13.008)	1	-	-
Chymase (S01.140)	-	-	2
Coag. factor Xa (S01.216)	-	-	1
PLAU (S01.231)	-	-	1
t-plasmin.activator (S01.232)	1	-	-
Plaminogen (S01.233)	-	-	1
Epitheliasin (S01.247)	-	-	1
Furin (S08.071)	-	-	2

Our approach retrieved 54 top scoring protease-protein target pairs, involving 25 proteases. Ten of the predicted pairs are verified in literature (column: Confirmed). We confirmed the remaining ones as plausible, since a similar pair, involving the same protein target, is cleaved by a protease of the same family of the predicted one, or other targets belonging to same family are cleaved by the protease (column: Same family); or that the predicted pair is mentioned in the same qualitative, quantitative or disease machinery described in literature (column: Shared mechanism). The numbers indicate for each Protease how many targets have been retrieved in literature according to the three criteria.

4 Conclusions

In this paper we present a novel application, based on matrix tri-factorization of multiple, heterogeneous data sources, to infer novel protein targets for human proteases. The main limitation of the proposed algorithm is that the output consists on the presence/absence of the cleavage, without indication of the cleavage point. Another limitation is the lack of integration of tertiary and secondary structure data, as well as other physicochemical characteristics, such as the disordered regions, that are typically exploited by other algorithms. Both these limitations are implicit in the features encoding of our model, based on a matricial representation of ontological relations between several data types. We plan to integrate these features in a future work, which could exploit the presented method to infer the presence of the cleavage, and then apply another approach to estimate the cleavage site position along the primary structure. In contrast to previous works, focusing on single-protease models, our approach consists of a broad, general approach, for the first time encapsulating both protease-protein target knowledge and structured biological ontologies in a single framework. Besides providing a larger scope, our model consistently outperforms state-of-the-art models.

Acknowledgements

We are deeply grateful to (in random order) Sarah Boyd, Jiangning.Song, Francesco Pala, Neil Rawlings, Riccardo Bellazzi, and Jerico Revote for the invaluable help and support.

Funding

While developing part of this work, Simone Marini was an International Research Fellow of the Japan Society for the Promotion of Science.

Conflict of Interest: none declared.

References

Andrés,V. (2014) Vitamin D puts the brakes on angiotensin II-induced oxidative stress and vascular smooth muscle cell senescence. *Atherosclerosis*, **236**, 444–447.
Bao,Y. *et al.* (2018) Toward more accurate prediction of caspase cleavage sites: a comprehensive review of current methods, tools and features. *Brief. Bioinform.*
Barkan,D.T. *et al.* (2010) Prediction of protease substrates using sequence and structure features. *Bioinformatics*, **26**, 1714–1722.
Barthel,D. *et al.* (2012) Plasminogen Is a Complement Inhibitor. *J. Biol. Chem.*, **287**, 18831–18842.
Boyd,S.E. *et al.* (2005) PoPS: a computational tool for modeling and predicting protease specificity. *J. Bioinform. Comput. Biol.*, **3**, 551–585.
Chatr-aryamontri,A. *et al.* (2017) The BioGRID interaction database: 2017 update. *Nucleic Acids Res.*, **45**, D369–D379.
Chu,J. *et al.* (2015) Gamma secretase-activating protein is a substrate for caspase-3: implications for Alzheimer's disease. *Biol. Psychiatry*, **77**, 720–728.
Finn,R.D. *et al.* (2017) InterPro in 2017-beyond protein family and domain annotations. *Nucleic Acids Res.*, **45**, D190–D199.
Gene Ontology Consortium (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.*, **43**, D1049-1056.
Gibot,L. *et al.* (2009) Human caspase 7 is positively controlled by SREBP-1 and SREBP-2. *Biochem. J.*, **420**, 473–483.
Gilchrist,M.A. *et al.* (2004) A statistical framework for combining and interpreting proteomic datasets. *Bioinformatics*, **20**, 689–700.
Gligorijević,V. *et al.* (2016) Fuse: multiple network alignment via data fusion. *Bioinformatics*, **32**, 1195–1203.
Hashimoto,G. *et al.* (2002) Matrix metalloproteinases cleave connective tissue growth factor and reactivate angiogenic activity of vascular endothelial growth factor 165. *J. Biol. Chem.*, **277**, 36288–36295.
Hosgood,H.D. *et al.* (2008) Caspase polymorphisms and genetic susceptibility to multiple myeloma. *Hematol. Oncol.*, **26**, 148–151.
James,M.N.G. (1999) Handbook of proteolytic enzymes, edited by A. J. Barrett, N. D. Rawlings, and J. F. Woessner. 1998. London: Academic Press. 1666 pp. $250.00. $90.00 for the CD-ROM. *Protein Sci.*, **8**, 693–694.
Jiang,L. *et al.* (2008) Increased aortic calpain-1 activity mediates age-associated angiotensin II signaling of vascular smooth muscle cells. *PloS One*, **3**, e2231.
Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.
Lam,M.P.Y. *et al.* (2016) Data-Driven Approach To Determine Popular Proteins for Targeted Proteomics Translation of Six Organ Systems. *J. Proteome Res.*, **15**, 4126–4134.
Lee,W.K. *et al.* (2009) Polymorphisms in the Caspase7 gene and the risk of lung cancer. *Lung Cancer Amst. Neth.*, **65**, 19–24.

Lenardo,M. *et al.* (1999) Mature T lymphocyte apoptosis--immune regulation in a dynamic and unpredictable antigenic environment. *Annu. Rev. Immunol.*, **17**, 221–253.
Marini,S. *et al.* (2011) In silico Protein-Protein Interaction prediction with sequence alignment and classifier stacking. *Curr. Protein Pept. Sci.*, **12**, 614–620.
Mosca,R. *et al.* (2014) 3did: a catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Res.*, **42**, D374-379.
Oh,J.E. *et al.* (2010) Mutational analysis of CASP10 gene in colon, breast, lung and hepatocellular carcinomas. *Pathology (Phila.)*, **42**, 73–76.
Okada,Y. (2017) Chapter 8 - Proteinases and Matrix Degradation. In, Firestein,G.S. *et al.* (eds), *Kelley and Firestein's Textbook of Rheumatology (Tenth Edition)*. Elsevier, pp. 106–125.
Orlowski,J. *et al.* (2007) Overrepresentation of interactions between homologous proteins in interactomes. *FEBS Lett.*, **581**, 52–56.
Planas-Iglesias,J. *et al.* (2013) iLoops: a protein-protein interaction prediction server based on structural features. *Bioinformatics*, **29**, 2360–2362.
Raghavachari,B. *et al.* (2008) DOMINE: a database of protein domain interactions. *Nucleic Acids Res.*, **36**, D656–D661.
Rawlings,N.D. *et al.* (2016) Twenty years of the MEROPS database of proteolytic enzymes, their substrates and inhibitors. *Nucleic Acids Res.*, **44**, D343–D350.
Singh,O. and Su,E.C.-Y. (2016) Prediction of HIV-1 protease cleavage site using a combination of sequence, structural, and physicochemical features. *BMC Bioinformatics*, **17**, 478.
Skidgel,R.A. and Erdös,E.G. (2004) Angiotensin converting enzyme (ACE) and neprilysin hydrolyze neuropeptides: a brief history, the beginning and follow-ups to early studies. *Peptides*, **25**, 521–525.
Song,J. *et al.* (2010) Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics*, **26**, 752–760.
Song,J. *et al.* (2012) PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites. *PloS One*, **7**, e50300.
Szklarczyk,D. *et al.* (2017) The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.*, **45**, D362–D368.
Vitali,F. *et al.* (2016) A Network-Based Data Integration Approach to Support Drug Repurposing and Multi-Target Therapies in Triple Negative Breast Cancer. *PLOS ONE*, **11**, e0162407.
Vitali,F. *et al.* (2018) Patient similarity by joint matrix trifactorization to identify subgroups in acute myeloid leukemia. *JAMIA Open*, **1**, 75–86.
Wang,M. *et al.* (2014) Cascleave 2.0, a new approach for predicting caspase and granzyme cleavage targets. *Bioinformatics*, **30**, 71–80.
Wilkins,M.R. *et al.* (1999) Protein identification and analysis tools in the ExPASy server. *Methods Mol. Biol. Clifton NJ*, **112**, 531–552.
Yang,B. *et al.* (2014) Matrix metalloproteinase-9 overexpression is closely related to poor prognosis in patients with colon cancer. *World J. Surg. Oncol.*, **12**, 24.
Zhao,X. *et al.* (2016) Caspase-2 cleavage of tau reversibly impairs memory. *Nat. Med.*, **22**, 1268.
Zhen,E.Y. *et al.* (2008) Characterization of metalloprotease cleavage products of human articular cartilage. *Arthritis Rheum.*, **58**, 2420–2431.
Žitnik,M. and Zupan,B. (2015) Data Fusion by Matrix Factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, **37**, 41–53.