Supervised biomedical semantic similarity

Rita T. Sousa[1*], Sara Silva[1], Catia Pesquita[1]

1 LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal

* risousa@ciencias.ulisboa.pt

Abstract

Semantic similarity between concepts in knowledge graphs is essential for several bioinformatics applications, including the prediction of protein-protein interactions and the discovery of associations between diseases and genes. Although knowledge graphs describe entities in terms of several perspectives (or semantic aspects), state-of-the-art semantic similarity measures are general-purpose. This can represent a challenge since different use cases for the application of semantic similarity may need different similarity perspectives and ultimately depend on expert knowledge for manual fine-tuning.

We present a new approach that uses supervised machine learning to tailor aspect-oriented semantic similarity measures to fit a particular view on biological similarity or relatedness. We implement and evaluate it using different combinations of representative semantic similarity measures and machine learning methods with four biological similarity views: protein-protein interaction, protein function similarity, protein sequence similarity and phenotype-based gene similarity. The results demonstrate that our approach outperforms non-supervised methods, producing semantic similarity models that fit different biological perspectives significantly better than the commonly used manual combinations of semantic aspects. Moreover, although black-box machine learning models produce the best results, approaches such as genetic programming and linear regression still produce improved results while generating models that are interpretable.

Background

Recent developments in knowledge representation have led to a rise in the amount of knowledge that is being stored in biomedical knowledge graphs (KGs) [1] and, consequently, to an increase of the relevance of the KGs in biomedical research. Such biomedical KGs contain factual knowledge about biological entities and their relations (linked data) in a fully machine-readable format. In particular, biomedical ontologies have been extensively used to describe biological entities in those KGs. Biomedical ontologies are used in areas ranging from gene function [2], to biomolecules [3] or phenotypes [4, 5]. There are currently over 800 biomedical ontologies in BioPortal [6].

Since measuring the similarity or distance between entities is fundamental to many tasks, semantic similarity has been used to explore KGs as a way to compare KG entities. There are a wide variety of bioinformatics applications that benefit from using semantic similarity over biomedical KGs, namely protein-protein interaction (PPI) prediction [7, 8], disease-associated genes identification [9, 10], and drug-drug interaction prediction [11, 12]. A semantic similarity measure (SSM) can be defined as a function that estimates the closeness in meaning between two entities. Over the years, several SSMs have been developed with most measures falling in the category of taxonomic

semantic similarity (also referred to as ontology-based semantic similarity, or only semantic similarity) [13]. However, graph embeddings, a most recent research direction, can also be used to compute semantic similarity [14–16].

Taxonomic semantic similarity compares ontology entities based on the taxonomic relations within the ontology graph. In KGs, this translates to measuring the similarity between either ontology classes or KG entities (annotated with a set of classes) [17]. Taxonomic SSMs are generally designed by an expert based on a set of assumptions about how an ontology is used and what should constitute a similarity. For class-based semantic similarity, edge-based measures rely on algorithms designed for graph analysis [18, 19]. However, the majority of methods explore the properties of each class involved, typically relying on the information content (IC) of a class, a measure of how informative (or in other words, specific) a class is, and then using it to measure the shared meaning between two classes [20, 21]. In entity-based semantic similarity, each instance is described with a set of classes which are then processed using one of two approaches: pairwise, where pairwise comparisons between all concepts annotating each entity are considered; groupwise, where set, vector, or graph-based measures are employed, circumventing the need for pairwise comparisons [22, 23]. More recently, GARUM [24], a semantic similarity approach based on machine learning (ML), was proposed. In this approach, a supervised regression algorithm receives several similarity measures of hierarchy, neighborhood, shared information, and attributes, and then predicts a final similarity score.

Regarding graph embedding similarity, an embedding is a vector representation resulting from the use of semantic information mapping techniques. There are a variety of methods for building KG embeddings [25]. While some focus on exploring solely the KG facts (like translational distance models [26, 27] or semantic matching [28, 29]), others also include additional information, such as entity types, relation paths, axioms and rules, or textual information. More recently, path-based approaches, such as node2vec [30], RDF2Vec [31], and Onto2Vec [14], have been proposed by transforming the KG into node sequences. A graph is represented as a set of random walk paths sampled from it, and then natural language methods are applied to the sampled paths for graph embedding. After employing graph embedding methods, each entity is represented by a vector. It is then possible to compute the graph embedding similarity between two entities by computing the distance of their corresponding vectors in the Euclidean space.

A drawback of semantic similarity is that these measures are general-purpose with no application-specific knowledge. However, KGs provide multiple perspectives (or semantic aspects) over an entity and, depending on our viewpoint of the domain or the task for which we want to use semantic similarity, some portions of the KG may be irrelevant for similarity computation. This brings us to the challenge of tailoring SSMs to the problem at hand.

Consider the following example on comparing proteins according to their function. One of the most prominent tools for representing this knowledge in a formal way is the Gene Ontology (GO) [2] which describes protein function concerning three semantic aspects: the *molecular functions* they perform, the *biological processes* they intervene in and the *cellular components* where they are active. These descriptions are materialized in the form of GO annotations [32] - GOA files, which link a protein identifier to a GO class with associated metadata. We can consider the GO and associated GOA to compose a KG. From a biochemist point of view, two proteins playing the same molecular functions are very similar. However, these proteins can be very different from a physiological perspective if they participate in different biological processes at the whole-organism level. Therefore, depending on our goal, different semantic aspects should be taken into consideration in similarity computation. The selection of which

semantic aspects to use for computing semantic similarity usually falls to the domain expert, rendering semantic similarity applications dependent on fine-tuning.

In previous work, we developed a methodology to predict protein-protein interactions that uses genetic programming to evolve combinations of aspect-oriented semantic similarities [33]. The positive results inspired us to hypothesize that if data regarding a specific biomedical entity similarity is available (e.g, sequence similarity, domain similarity, etc), we can use it to effectively learn a supervised semantic similarity, that is able to combine different semantic aspects without user input, but is tailored to capture a specific biological similarity view.

In this work, we present a novel approach that uses ML methods over a set of semantic similarities, each computed over a different semantic aspect, to learn the best combination of semantic aspects tailored to a specific biological similarity, in effect producing a supervised similarity. The result is a semantic similarity model that is tailored to better capture particular biological similarity views, such as domain similarity or protein-protein interaction. The proposed approach was implemented using different SSMs (three based on embeddings and two based on taxonomic similarity) coupled with eight ML methods (classical ML approaches and neural network-based approaches). This implementation aims to elucidate which SSMs are more suitable for different combinations of supervised learning approaches. Since there is no gold standard for the similarity between complex biomedical entities, we take advantage of similarity proxies to train the models and evaluate them. These proxies of similarity rely on objective representations of entities (e.g., gene sequence, domains) and calculate similarity using mathematical expressions or other algorithms (e.g., BLAST-based similarity for sequences).

We evaluate the proposed approach in a set of 21 benchmark datasets [34] that have varying sizes with different semantic annotation characteristics and include data from two biomedical ontologies, GO and Human Phenotype Ontology (HPO). These datasets contain four proxies for biomedical entity similarity calculated based on protein sequence similarity, protein function family similarity, protein-protein interactions and phenotype-based gene similarity. Although these proxy similarities do not provide the broad spectrum comparison that semantic similarity supports, they are known to relate to relevant characteristics of the underlying entities. Our approach is compared with combinations of semantic aspects that emulate expert choices to understand how well the approach captures entity similarity. The results on benchmark datasets demonstrate the ability of our approach to significantly improve the estimation of similarity between biomedical entities.

Methods and materials

We have developed a novel approach[1] to learn the similarity between entities represented in KGs optimized towards a specific similarity proxy. This tailoring is achieved by considering the similarities for different semantic aspects (Definition 1), as opposed to the static SSMs (Definition 2).

Definition 1. A **semantic aspect** represents a perspective of the representation of KG entities. It can correspond to a given set of property types (e.g., describing a person only through properties having geographical locations as a range) or portions of the graph (e.g., describing a protein only through the biological process subgraph of the GO).

Definition 2. A **static SSM** calculates values of similarity by processing the KG without additional external input or tailoring to a specific similarity proxy.

[1] will be made available on github

An overview of the approach is shown in Fig 1. The first step consists of identifying the semantic aspects that describe the KG entities. Our approach takes as pre-defined semantic aspects the subgraphs when the KGs have multiple roots (such as GO) or the subgraphs rooted in the classes at a distance of one from the KG root class. As an alternative, semantic aspects can be manually defined. The next step is representing each instance (i.e., a pair of KG entities) according to static KG-based similarities computed for each semantic aspect. The third step in our approach is to select the similarity proxy for which we want to tailor the similarity. The last step is to employ a ML method to learn a supervised semantic similarity. The ML algorithms are used for regression where the expected outputs are the proxy similarity values.

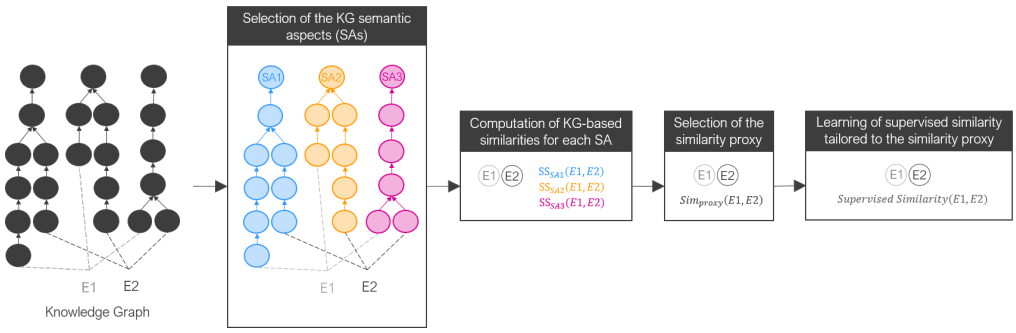

Fig 1. Overview of the proposed approach.

This approach is independent of the semantic aspects, the specific implementation of KG-based similarity and the ML algorithm employed in regression. The following sections present the specific details of the implementation that currently supports five different SSMs (three based on embedding similarity and two based on taxonomic similarity) and eight targeted supervised learning approaches (classical ML approaches and a neural network-based approach).

The approach is evaluated in protein and gene benchmark datasets. For the protein datasets, we consider the GO aspects as semantic aspects. After semantic similarity computations, each instance of the dataset, that represents a protein pair, is characterized by three values corresponding to the semantic similarity between them for the three GO aspects, and a proxy similarity value. The models returned in the second step are then the combinations of the similarity scores of the three GO aspects. For the gene dataset, in addition to the three GO aspects, the similarity is also calculated for the HPO phenotypic abnormality subgraph. Therefore, instead of three semantic aspects, we consider four semantic aspects.

Data

Our approach takes as input an ontology file, an instance annotation file and a list of instance pairs with proxy similarity values. We evaluate our approach using benchmark datasets and two different KGs.

Benchmark datasets

The 21 benchmark datasets are presented in [34] and are available online[2] (dated June 2020). These datasets explore four proxy similarities based on protein and gene properties. This resulted in one gene dataset and 16 protein datasets, divided by species, level of annotation completion and similarity proxy, and four additional

[2] `https://github.com/liseda-lab/kgsim-benchmark`

datasets, combining all species' protein pairs in the same proxy group. Datasets range in size from 264 individual proteins and 428 pairs to 27 thousand proteins and 158 thousand pairs. The protein datasets are described in Table 1. Regarding the gene dataset, it has 2026 distinct human genes and 12000 pairs.

Table 1. Species, number of proteins and pairs, annotation completion and proxy measure they are based on for all protein datasets.

	PFAM datasets				PPI datasets			
	1 Aspect		3 Aspects		1 Aspect		3 Aspects	
Species	**Prots**	**Pairs**	**Prots**	**Pairs**	**Prots**	**Pairs**	**Prots**	**Pairs**
DM	7494	53797	5810	52460	481	397	335	270
EC	1250	4623	748	1813	371	738	264	428
HS	13604	60176	12487	60163	7644	44677	7149	42204
SC	4783	42192	3660	30747	3874	34772	2959	21577
ALL	27131	158512	22705	142736	12370	80584	10707	64479

Table 2 shows the similarity approaches employed. In the PFAM datasets, two proxies of protein similarity based on their biological properties were employed: sequence similarity and PFAM similarity. In PPI protein datasets, two similarity proxies were also employed: sequence similarity and protein-protein interactions. Concerning the gene benchmark dataset, the proxy similarity is based on phenotypic series.

- ***Sequence similarity*** ($\mathrm{Sim}_{\mathrm{seq}}$) measures the relationship between two sequences and it establishes the likelihood for sequence homology. We infer homology (i.e., common evolutionary ancestry) when two sequences share more similarity than would be expected by chance. A sequence similarity value is aimed to approximate the evolutionary distance between proteins.

- ***PFAM similarity*** ($\mathrm{Sim}_{\mathrm{PFAM}}$) is computed by comparing the functional regions (commonly termed domains) that exist in each protein sequence. Protein functional domains were extracted from the PFAM [35]. Since protein domains typically correspond to functional sites of a protein, determining similarity between domains can help to define protein function.

- ***Protein-protein interaction*** (PPI) has a binary representation: 1 if the proteins interact, 0 otherwise. Two proteins are considered to be similar if they interact. PPIs are responsible for many critical functions in biology and are highly relevant to disease states.

- ***Phenotypic series similarity*** ($\mathrm{Sim}_{\mathrm{PS}}$) is based on OMIM's Phenotypic Series [36], which are groups of identical or similar phenotypes and their associated genes. Phenotypic similarity reflects the similarity between genes and can help to find biological modules of functionally related genes.

Table 2. Available similarity proxies for each dataset type.

Dataset Type	Similarity Proxies
PFAM	$\mathrm{Sim}_{\mathrm{seq}}$ and $\mathrm{Sim}_{\mathrm{PFAM}}$
PPI	$\mathrm{Sim}_{\mathrm{seq}}$ and PPI
Gene	$\mathrm{Sim}_{\mathrm{PS}}$

Gene ontology knowledge graph

GO [2] is the most widely used biological ontology. It defines the universe of classes, also called "GO terms", associated with gene product (proteins or RNA) functions and how these functions are related with each other with respect to these three aspects: (i) biological process (BP), the larger process accomplished by multiple molecular activities in which the gene product is active; (ii) molecular function (MF), the biochemical or molecular-level activity of the gene product; (iii) cellular component (CC), the location relative to cellular structures in which the gene product performs a function. The GO KG is composed by the GO and GO annotations where instances are proteins, which are linked to GO via their annotations.

In this work, the GO KG, with its three semantic aspects (BP, CC and MF), is used to compute the semantic similarity between two proteins for the protein datasets and between two genes for the gene dataset.

Human phenotype knowledge graph

The HPO [4] contains about terms describing phenotypic abnormalities found in human hereditary diseases. The HPO is organized as independent subontologies that cover different categories: "Phenotypic abnormality", "Mode of inheritance", "Clinical course", "Clinical modifier" and "Frequency". Since the subontology "Phenotypic Abnormality" is the ontology branch that describes the phenotypes the gene is associated with, the HPO KG is composed by this subontology and HPO annotations. An HPO annotation associates a specific gene with a specific HPO class.

In this work, the HPO KG is used to compute the semantic similarity between two genes based on the phenotypes that describe them.

Static similarity computation

The following subsections present the specific details of the five different KG-based SSMs: two based on taxonomic similarity and three based on embeddings.

Taxonomic semantic similarity

The taxonomic semantic similarity is calculated using two state-of-the-art measures, derived by combining one information content approach ($\mathrm{IC}_{\mathrm{Seco}}$) with one of two set similarity measures (ResnikBMA, SimGIC). These were selected by their high performance in the biomedical domain [37].

$\mathrm{IC}_{\mathrm{Seco}}$ is a structure-based approach proposed by [21] based on the number of direct and indirect descendants and given by

$$\mathrm{IC}_{\mathrm{Seco}}(t) = 1 - \frac{\log\left[\mathrm{hypo}(t) + 1\right]}{\log\left[\mathrm{maxnodes}\right]} \tag{1}$$

where $\mathrm{hypo}(t)$ is the number of direct and indirect descendants from term t (including term t) and maxnodes is the total number of concepts in the ontology.

ResnikBMA is based on the class-based measure proposed by Resnik [20] in which the similarity between two classes corresponds to the IC of their most informative common ancestor and given by:

$$\mathrm{ResnikBMA}(e_1, e_2) = \frac{\sum_{t_1 \in \mathrm{S}(e_1)} \mathrm{sim}(t_1, t_2)}{2|\mathrm{S}(e_1)|} + \frac{\sum_{t_2 \in \mathrm{S}(e_2)} \mathrm{sim}(t_1, t_2)}{2|\mathrm{S}(e_2)|} \tag{2}$$

where $\mathrm{S}(e_i)$ is the set of annotations for entity e_i and $\mathrm{sim}(t_1, t_2)$ is the semantic similarity between class t_1 and class t_2 and is defined as:

$$\mathrm{sim}(c_1, c_2) = \max\{\mathrm{IC}(c) : c \in \{\mathrm{A}(c_1) \cap \mathrm{A}(c_2)\}\} \tag{3}$$

where $\mathrm{A}(c_i)$ is the set of ancestors of c_i.

SimGIC is a groupwise approach proposed by [22], based on a Jaccard index in which each term is weighted by its IC and given by

$$\mathrm{SimGIC}(e_1, e_2) = \frac{\sum_{t \in \{\mathrm{S}(e_1) \cap \mathrm{S}(e_2)\}} \mathrm{IC}(t)}{\sum_{t \in \{\mathrm{S}(e_1) \cup \mathrm{S}(e_2)\}} \mathrm{IC}(t)} \tag{4}$$

where $\mathrm{S}(e_i)$ is the set of annotations (direct and inherited) for entity e_i.

The Semantic Measures Library 0.9.1 [38] was used to compute the taxonomic semantic similarity.

Graph embedding similarity

We employ three graph embedding approaches, namely RDF2Vec, TransE, and distMult, using an RDF2Vec python implementation[3] and the OpenKE library[4]. These approaches were selected because they are representative of the main types of graph embedding techniques.

- ***RDF2Vec*** [31] is a path-based approach that adapts the node2vec [30] to RDF graphs. In RDF2Vec, edge direction is taken into account, enriching the semantics of the learning approach, and neural language models are employed over random walks on the RDF graph to produce the embeddings.

- ***TransE*** [26] is the most representative translational distance embedding approach that exploits distance-based scoring functions. The basic idea of the translational distance models is that each fact represents the distance between the two entities, usually after a translation carried out by the relations.

- ***distMult*** [28] is a semantic matching approach that exploits similarity-based scoring functions by matching latent semantics of entities and relations embodied in their vector space representations. DistMult takes the inherent structure of relations into account by employing the tensor factorization.

We generate protein or gene graph embeddings for each semantic aspect using these approaches[5] and then, to compute the graph embeddings similarities, we employ cosine similarity between the vectors representing each entity in the pair.

Supervised similarity computation

Our approach combines the semantic similarities computed for each semantic aspect and returns a supervised similarity. The aggregation function is computed by a supervised regression algorithm. Therefore, each regressor receives the similarity values for each semantic aspect as input features (independent variables) and a similarity proxy value as the expected output (dependent variable), and returns an aggregated similarity score as the predicted output. We employ eight well-known classes of ML models to train regressors using scikit-learn 21.3 [39] library: linear regression (LR), bayesian ridge

[3] https://github.com/IBCNServices/pyRDF2Vec
[4] https://github.com/thunlp/OpenKE/tree/OpenKE-Tensorflow1.0
[5] parameters for each embedding method are supplied in S1 File

(BR), K-nearest neighbor (KNN), genetic programming (GP), decision tree (DT), random forest (RF), XGBoost (XGB), and multi-layer perception (MLP).

These algorithms are selected as representative of different types of ML methods. LR [40] assumes there is a linear relationship between the independent and dependent variables. BR [41] is also a linear model but uses the Bayes theorem to find the posterior distribution over all parameters. KNN [42] explores the feature space and reaches a prediction for each sample based on the expected outputs of its neighbors. In DT [43], trees are constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. The basic idea of tree growth is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". GP [44] is an evolutionary computation technique inspired by Darwinian natural selection and Mendelian genetics. GP tries to optimize a combination of variable and operators/functions. MLP [45] is a class of feedforward artificial neural networks that learn non-linear functions for regression through backpropagation of errors. At last, RF [46] and XGB [47] are ensemble methods. The goal of ensemble methods is to combine the decisions from multiple models to improve the overall performance. RF builds several estimators independently and then combines their predictions through a voting scheme, while XGB builds base estimators sequentially and tries to reduce the bias of the combined estimator through gradient boosting.

Except for GP, XGB and RF, the parameters are used with the default scikit-learn values. For running GP, we use gplearn 3.0[6], a freely available package that runs with the scikit-learn library. Although there is a vast array of tunable parameters in every GP implementation, they usually do not substantially influence the quality of the evolved solutions [48]. For this reason, the parameters have the same values as proposed in [33]. For XGB, we use the XGBoost 1.1.1 package[7], with the values of some parameters altered to maximize the performance of the model, through grid search. For RF, using scikit-learn, we also optimize some parameters[8].

Supervised similarity evaluation

The focus of our evaluation approach is to assess the ability of our approach to improve semantic similarity computations, avoiding the need for expert knowledge. For each combination of an SSM with an ML algorithm, we compute the Pearson's correlation coefficient between the obtained supervised similarity (predicted values) and the respective similarity proxies (expected values).

For cross-validation, each dataset is split into ten folds. The same ten folds are used throughout all the experiments. For each fold, we take that fold as the test set and the remaining nine folds as the training set. Each ML algorithm learns on the training set and outputs its predictions for the test set, where the Pearson correlation coefficient is calculated. The results we report are the median of the ten Pearson correlation coefficients calculated on the ten folds.

We compute the static similarity for each semantic aspect and use, as baselines, the single aspect similarities and two well-known strategies for combining the single aspect scores, the average and maximum. By comparing these baselines to the supervised approaches, we aim at investigating the ability of ML methods to learn combinations of semantic aspects that improve the calculation of similarity.

[6] https://gplearn.readthedocs.io/en/stable/
[7] https://xgboost.readthedocs.io
[8] optimized parameters are supplied in S1 File

Results and discussion

Static similarity

Before performing the comparative evaluation, we investigate the behavior of the five similarity-based semantic measures employed. Tables 3 and 4 show the Pearson correlation coefficient between the static similarity and the proxy similarity using different SSMs for the gene and protein datasets, respectively. For the sake of simplicity, Table 3 presents only the correlation for the protein datasets combining all species' protein pairs in the same group proxy, including proteins with at least one annotation in each GO aspect (ALL1). S1 File provides the results for the remaining protein datasets.

Table 3. Pearson correlation coefficient between static semantic similarity and similarity proxies for the protein datasets with one level of annotation.

Proxy	Dataset	SSM	Static Similarity				
			BP	**CC**	**MF**	**AVG**	**MAX**
Sim_{seq}	PFAM	ResnikBMA	0.528	0.373	0.291	0.481	0.399
		SimGIC	0.552	0.406	0.415	0.547	0.406
		RDF2Vec	0.540	0.437	0.419	0.544	0.457
		TransE	-0.020	-0.017	-0.001	-0.021	-0.011
		distMult	0.398	0.236	0.322	0.467	0.429
	PPI	ResnikBMA	0.258	0.222	0.326	0.323	0.250
		SimGIC	0.317	0.257	0.370	0.380	0.280
		RDF2Vec	0.274	0.237	0.297	0.316	0.268
		TransE	0.355	0.352	0.364	0.498	0.401
		distMult	0.277	0.239	0.202	0.369	0.310
Sim_{PFAM}	PFAM	ResnikBMA	0.448	0.370	0.456	0.525	0.500
		SimGIC	0.494	0.451	0.591	0.621	0.604
		RDF2Vec	0.524	0.466	0.619	0.627	0.623
		TransE	-0.032	-0.031	-0.011	-0.039	-0.027
		distMult	0.414	0.254	0.388	0.516	0.457
PPI	PPI	ResnikBMA	0.545	0.486	0.353	0.569	0.545
		SimGIC	0.486	0.428	0.318	0.496	0.458
		RDF2Vec	0.457	0.404	0.353	0.477	0.440
		TransE	0.031	0.087	0.035	0.071	0.076
		distMult	0.452	0.244	0.015	0.388	0.396

Table 4. Pearson correlation coefficient between static semantic similarity and similarity proxies for the human gene dataset.

Proxy	SSM	Static Similarity					
		HPO	**BP**	**CC**	**MF**	**AVG**	**MAX**
Sim_{PS}	ResnikBMA	0.601	0.210	0.142	0.055	0.413	0.552
	SimGIC	0.489	0.205	0.158	0.095	0.399	0.429
	RDF2Vec	0.526	0.230	0.182	0.123	0.396	0.351
	TransE	0.042	-0.019	-0.012	-0.019	-0.004	0.016
	distMult	0.015	0.184	0.105	0.041	0.179	0.182

For most datasets, each SSM shows a consistent behavior. Comparing the two taxonomic semantic similarity approaches, we verify that, in most cases, the maximum correlation is achieved when the ResnikBMA approach is used. Regarding the graph embedding approaches, TransE has performed worse than the other embedding methods. These differences are not unexpected, since the methods that put more emphasis on

local neighborhoods, such as translational distance approaches, are less suitable since they fail to capture longer-distance relations. This is relevant when most of the information to be processed is represented in the ontology portion of the KG, where taxonomic relations play an essential role. For that reason, in the following sections, the results obtained with TransE were excluded. distMult, a semantic matching method, is the second-best class of embeddings. Finally, RDF2Vec, a path-based approach, can capture taxonomic (longer-distance) relations which translates into a broader representation of the entities, achieving better results than the other embedding methods in most experiments.

When comparing the two types of semantic similarity, taxonomic similarity performs well across many evaluations and, in the majority of the datasets, has better performance than embedding similarity. The initial assumption was that embedding similarity could potentially outperform taxonomic similarity since semantic similarity is limited to the taxonomic relations within the ontology. In contrast, embeddings take into account all types of relations and, therefore, the embedding representations could, in principle, be more informative. However, the ability of taxonomic similarity to take into account class specificity may give it the advantage over embedding similarity to estimate similarity more accurately. Besides, taxonomic similarity measures are usually hand-crafted, providing human interpretable results for further analysis. On the contrary, embeddings methods describe an entity as a numerical vector and, most of the times, are not interpretable since it is not possible to obtain an explanation for the results.

It is also important to point out the differences between semantic aspects. These differences depend on the similarity proxy we are considering. For the sequence proxy, the differences between semantic aspects are not relevant, and there is no semantic aspect clearly superior to others. Previous works [49] already suggested the absence of a strong correlation between sequence and semantic similarities since there is a large number of proteins with low sequence similarity and high semantic similarity. Concerning the PPI proxy, proteins that interact in a cell are expected to participate in similar cellular locations and processes. As expected, the results indicate that using only the semantic similarity for MF provides worse results compared to the other single aspects. In opposition, for the PFAM proxy, we verify that the MF is a relevant semantic aspect. The more functional (or PFAM) domains two proteins share, the more likely it will be to share molecular functions since these domains are usually responsible by assigning functions to proteins. For the gene dataset, the HPO semantic aspect achieves better results compared to the GO semantic aspects. These results were also expected since the more phenotypic series two genes are associated with, the more likely it is that they share HPO classes. Regarding static combination approaches, in most cases, they achieve better results than the single aspects, with the average combination outperforming the maximum.

Supervised similarity

The similarity proxies reflecting different biological features allow us to use ML algorithms to learn a supervised similarity towards a viewpoint of the domain. We employ eight representative ML methods, including classical, ensemble and neural network based methods: linear regression (LR), bayesian ridge (BR), K-nearest neighbor (KNN), genetic programming (GP), decision tree (DT), random forest (RF), XGBoost (XGB), and multi-layer perception (MLP).

S1 File shows the median Pearson correlation coefficient between the similarity proxies and supervised similarity obtained with different ML methods.

To better compare the eight ML algorithms, we generated radar charts (Figs 2 to 6) showing the median Pearson correlation coefficient between supervised similarity and

each similarity proxy. Radar charts reveal which ML algorithms combined with different SSMs are scoring high or low within a dataset. In each radar plot, the ML algorithms are represented by different colors[9], and the SSMs are represented on different axes.

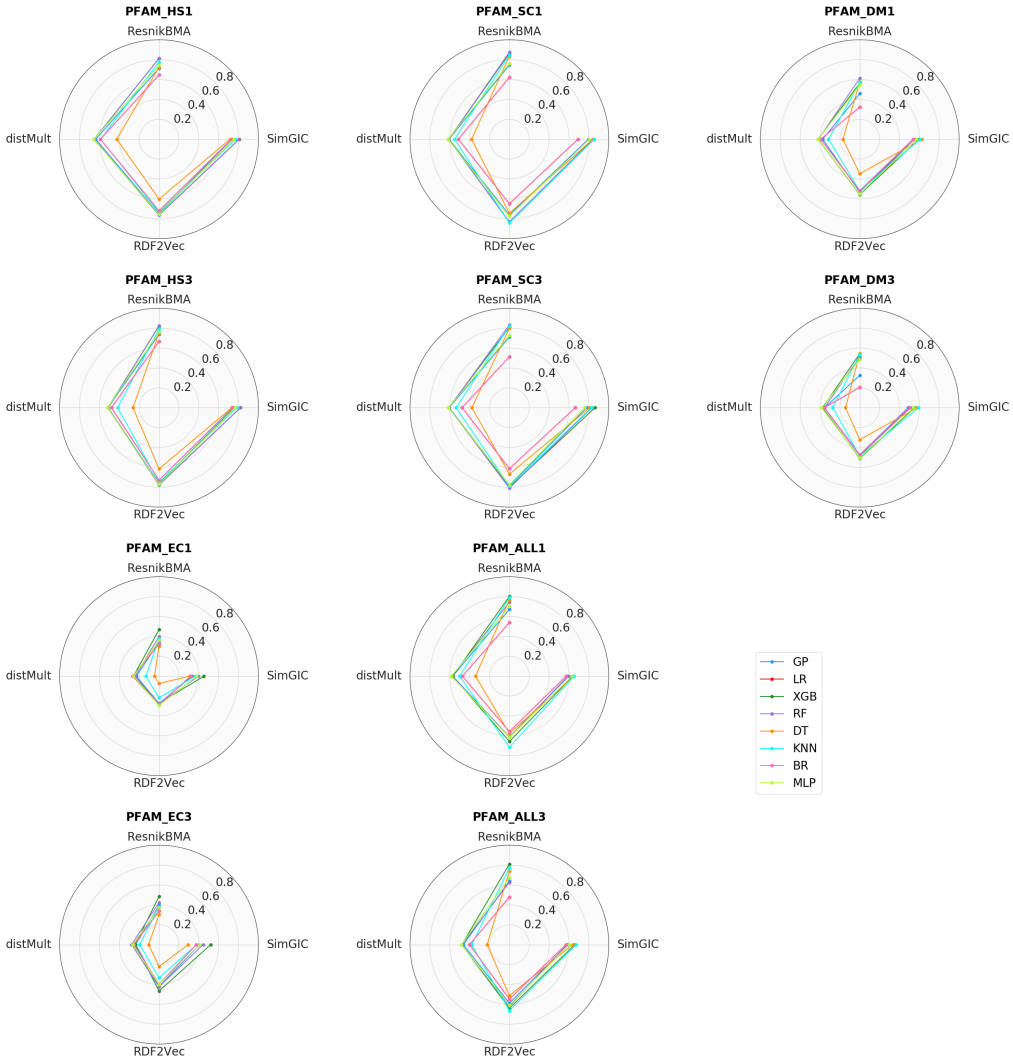

Fig 2. Radar charts using sequence proxy for PFAM datasets.

The performance of regression models obtained by DT is globally lower compared to the other ML algorithms. DT is one of the most commonly used approaches for supervised learning. However, since it is based on recursive binary splitting, DT may not be suitable for the current regression problem of finding the best combination of semantic aspects. LR and BR also show lower correlations in many cases. The Pearson correlation coefficients obtained by LR and BR are identical in most of the datasets. LR and BR assume a linear relationship between the independent and dependent variables, which is not valid for many cases. This characteristic may explain why these ML methods were not capable of learning suitable combinations of semantic aspects.

The very tight lines in the radar plots show that KNN, GP, and MLP achieve comparable results. Ensemble methods, like XGB and RF, achieve higher results in

[9]In most of the radar charts, the red line representing LR is not visible due to the overlapping of the BR pink line.

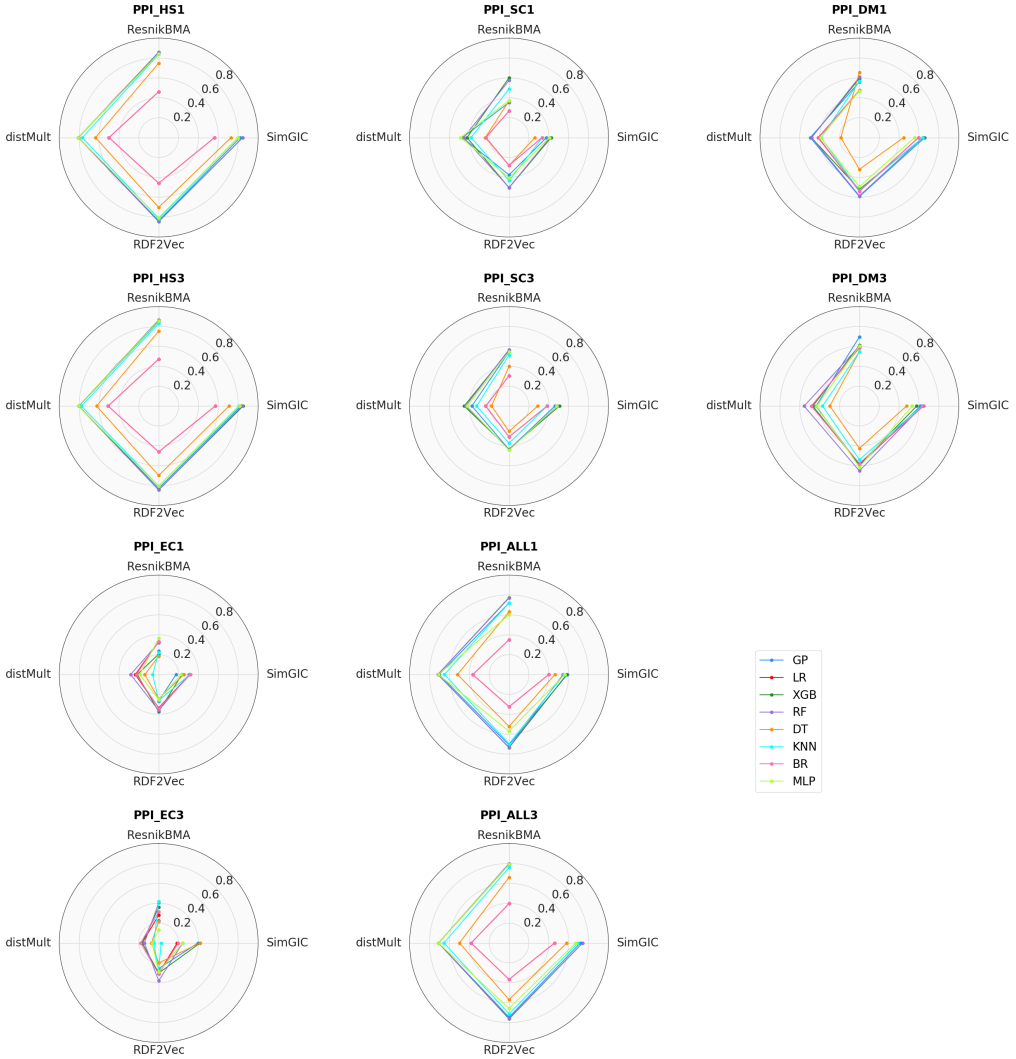

Fig 3. Radar charts using sequence proxy for PPI datasets.

most experiments. These results are expected since the ensemble methods combine the decisions from multiple models to improve the overall performance, and these methods have been successfully applied to different domains [50].

Comparing the SSMs, the results seem to indicate that taxonomic semantic similarity is a more suitable similarity-based semantic representation for learning. Although the static similarity results have already demonstrated that taxonomic semantic similarity achieves higher correlations than graph embedding similarity, these differences are more evident when we apply ML methods.

A close investigation of the full results presented in S1 File also reveals interesting results regarding the significance tests. Statistically significant differences are determined using pairwise non-parametric Kruskal-Wallis tests at $p < 0.01$. Although MLP seems to achieve lower correlation values, there are no significant differences between XGB, RF or MLP and the best ML algorithm in most cases (which is either XGB or RF). Significant differences are more common between SSMs.

In order to assess whether a particular combination of an ML method and a specific SSM increases performance, for each proxy similarity we ranked the possible combinations of SSMs with ML algorithms within each dataset. Then, we calculated the

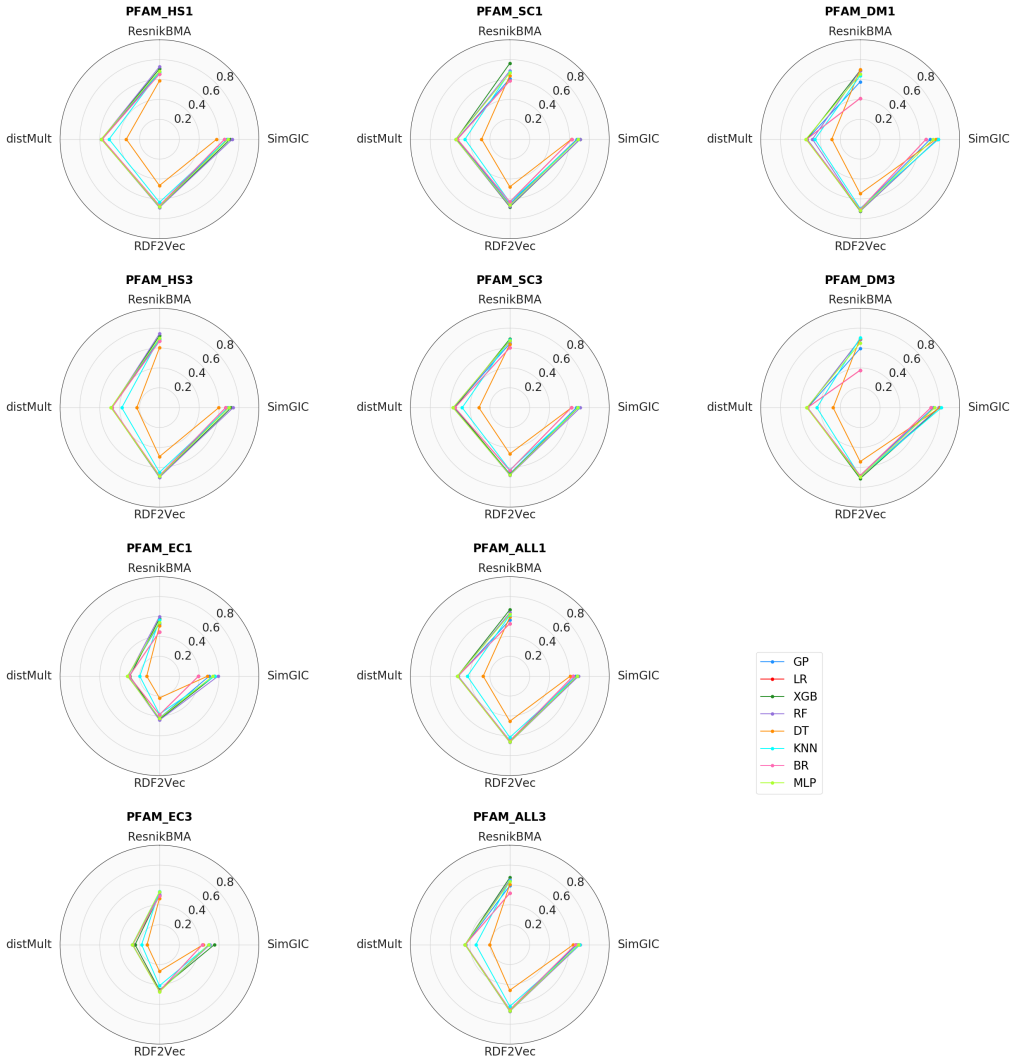

Fig 4. Radar charts using PFAM proxy for PFAM datasets.

average ranking of each SSM-ML combination. Table 5 shows the best combination for each proxy similarity, and they are always composed by a taxonomic SSM and an ensemble method.

Table 5. Best SSM-ML combination for each proxy similarity.

Proxy	**SSM**	**ML Algorithm**
Sim_{seq}	ResnikBMA	RF
Sim_{PFAM}	SimGIC	RF
PPI	ResnikBMA	XGB
Sim_{PS}	ResnikBMA	XGB

Choosing the right SSM and the right ML method for a particular application can be very hard. Several studies have been conducted to assess the performance of different SSMs, and different performances have been reported in different contexts [51]. Likewise, ML algorithms behave differently depending on many factors, from the type of problem at hand to the size of the data (No Free Lunch theorem [52]). Therefore, it is

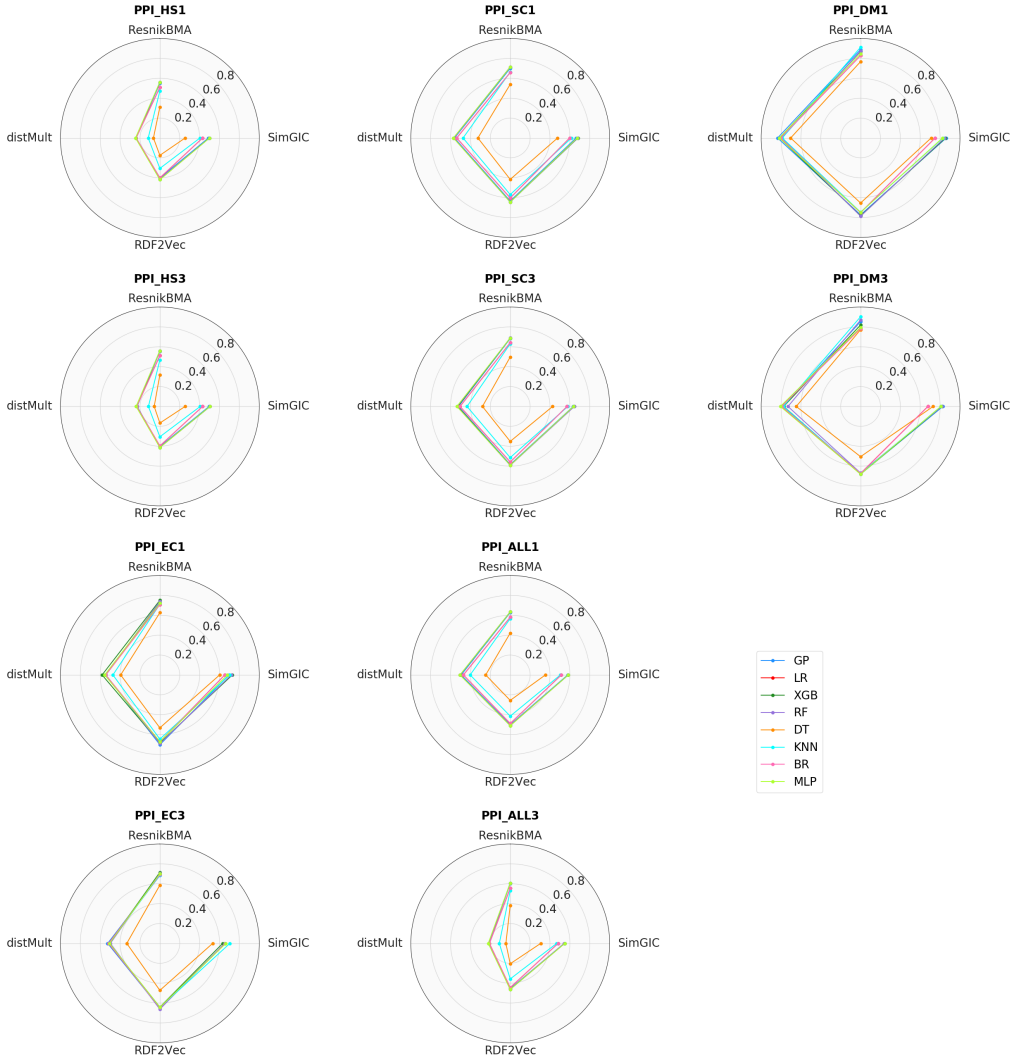

Fig 5. Radar charts using PPI proxy for PPI datasets.

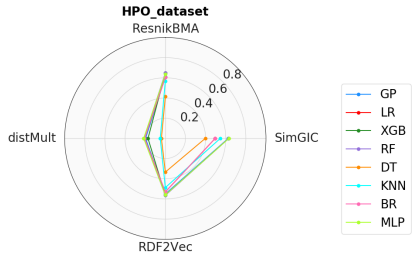

Fig 6. Radar charts using phenotype series proxy for gene dataset.

not straightforward to identify the best combination of SSM with ML algorithm that will work for all datasets and use cases. Nevertheless, the results in Table 5 seem to indicate that combining a taxonomic SSM with an ensemble method is a safe choice.

Supervised similarity interpretability

Although static SSMs, such as taxonomic SSMs, are hand-crafted and interpretable, supervised learning can lead to a loss of this valuable characteristic. Therefore, it is interesting to compare ML algorithms not only in terms of performance but also in terms of interpretability. The models obtained by KNN, BR, MLP and ensemble methods are more challenging to interpret, although some methods for explaining black-box models have been proposed [53]. In opposition, the LR models predict the target as a weighted sum of the feature inputs. These linear equations have an easy to understand interpretation. Table 6 shows, for each similarity proxy, a LR model obtained in one of the folds.

Proxy	**Model**
$\mathrm{Sim}_{\mathrm{seq}}$	$0.1450\ \mathrm{SS}_{\mathrm{BP}}+0.0693\ \mathrm{SS}_{\mathrm{CC}}+0.0916\ \mathrm{SS}_{\mathrm{MF}}-0.0058$
$\mathrm{Sim}_{\mathrm{PFAM}}$	$0.1943\ \mathrm{SS}_{\mathrm{BP}}+0.1861\ \mathrm{SS}_{\mathrm{CC}}+0.4344\ \mathrm{SS}_{\mathrm{MF}}+0.1211$
PPI	$0.6001\ \mathrm{SS}_{\mathrm{BP}}+0.6864\ \mathrm{SS}_{\mathrm{CC}}+0.0132\ \mathrm{SS}_{\mathrm{MF}}+0.1701$
$\mathrm{Sim}_{\mathrm{PS}}$	$0.8406\ \mathrm{SS}_{\mathrm{HPO}}+0.2063\ \mathrm{SS}_{\mathrm{BP}}+0.1282\ \mathrm{SS}_{\mathrm{CC}}+0.0004\ \mathrm{SS}_{\mathrm{MF}}+0.2261$

Table 6. Linear Regression models.

The solutions obtained by DT and GP are also, in principle, interpretable. However, in both cases, trees may grow to be very complex while learning complicated datasets, which can raise some difficulty in interpreting the solutions. Fig 7 shows, for each similarity proxy, a GP model obtained in one of the folds. To allow a better understanding, these models were simplified to remove redundant and inviable code. Although the frequency in which a given variable appears in a GP model does not necessarily measure its importance for the predictions, the GP model analysis can indicate which semantic aspects are most relevant for each similarity proxy. The obtained DT models are very large with multiple levels deep, which decreases their interpretability and visualization and are thus not shown.

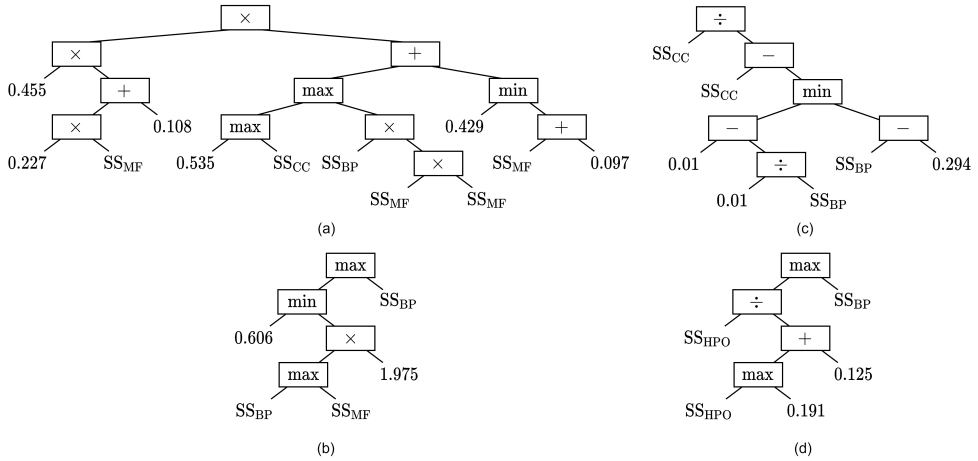

Fig 7. Parse trees representing GP models are shown for: (a) $\mathrm{Sim}_{\mathrm{seq}}$; (b) $\mathrm{Sim}_{\mathrm{PFAM}}$; (c) PPI; (d) $\mathrm{Sim}_{\mathrm{PS}}$. In Table 7, the GP models are translated to readable strings.

It is important to note that, although interpretable models achieve lower performance values than black-box models in most cases, as shown above, the supervised similarity obtained using LR and GP is still able to improve over the baselines.

Proxy	Model
$\mathrm{Sim}_{\mathrm{seq}}$	$0.455\times[0.227\mathrm{SS}_{\mathrm{MF}}+0.108]\times\left[\max\left(\max\left(0.535,\mathrm{SS}_{\mathrm{CC}}\right),\mathrm{SS}_{\mathrm{BP}}\times\mathrm{SS}_{\mathrm{MF}}\times\mathrm{SS}_{\mathrm{MF}}\right)+\min\left(\mathrm{SS}_{\mathrm{BP}},2\mathrm{SS}_{\mathrm{MF}}\right)\right]$
$\mathrm{Sim}_{\mathrm{PFAM}}$	$\max\left(\mathrm{SS}_{\mathrm{BP}},\min\left(0.606,\max\left(\mathrm{SS}_{\mathrm{BP}},\mathrm{SS}_{\mathrm{MF}}\right)\times 1.975\right)\right)$
PPI	$\frac{\mathrm{SS}_{\mathrm{CC}}}{\mathrm{SS}_{\mathrm{CC}}-\min\left(0.01-\frac{0.01}{\mathrm{SS}_{\mathrm{BP}}},\mathrm{SS}_{\mathrm{BP}}-0.294\right)}$
$\mathrm{Sim}_{\mathrm{PS}}$	$\max\left(\frac{\mathrm{SS}_{\mathrm{HPO}}}{\max\left(\mathrm{SS}_{\mathrm{HPO}},0.191\right)+0.125},\mathrm{SS}_{\mathrm{BP}}\right)$

Table 7. Genetic Programming models.

Static versus supervised similarity

Tables 8 to 11 compare the results obtained using static similarity and supervised similarity for sequence, PFAM, PPI and phenotypic series proxies, respectively. The static similarity was obtained using taxonomic SSMs (SimGIC or ResnikBMA), and then the Pearson correlation coefficient was computed for each proxy. Regarding supervised similarity, the median of Pearson correlation values was calculated for the proposed approach using a taxonomic SSM (SimGIC or ResnikBMA) with an ensemble method (XGB or RF) for each proxy, the combinations previously shown to produce the best results.

Table 8. Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{seq}}$ and ResnikBMA or SimGIC for the baselines and the median Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{seq}}$ and supervised similarity obtained using XGB or RF. In bold, the best result for each dataset-SSM.

Dataset	SSM	Static					Supervised	
		BP	CC	MF	AVG	MAX	XGB	RF
PFAM								
ALL1	ResnikBMA	0.528	0.373	0.291	0.481	0.399	**0.803**	0.746
	SimGIC	0.552	0.406	0.415	0.547	0.406	**0.640**	0.589
ALL3	ResnikBMA	0.466	0.334	0.325	0.445	0.349	**0.810**	0.626
	SimGIC	0.544	0.374	0.451	0.539	0.411	**0.658**	0.580
PPI								
ALL1	ResnikBMA	0.258	0.222	0.326	0.323	0.250	**0.774**	0.770
	SimGIC	0.317	0.257	0.370	0.380	0.280	**0.580**	0.542
ALL3	ResnikBMA	0.293	0.260	0.370	0.376	0.289	**0.798**	0.795
	SimGIC	0.371	0.295	0.411	0.439	0.316	0.739	**0.740**

Table 9. Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{PFAM}}$ and ResnikBMA or SimGIC for the baselines and the median Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{PFAM}}$ and supervised similarity obtained using XGB or RF. In bold, the best result for each dataset-SSM.

Dataset	SSM	Static					Supervised	
		BP	CC	MF	AVG	MAX	XGB	RF
PFAM								
ALL1	ResnikBMA	0.448	0.370	0.456	0.525	0.500	**0.669**	0.638
	SimGIC	0.494	0.451	0.591	0.621	0.604	0.680	**0.691**
ALL3	ResnikBMA	0.431	0.387	0.463	0.514	0.480	**0.674**	0.651
	SimGIC	0.506	0.498	0.608	0.644	0.622	0.692	**0.706**

Table 10. Pearson correlation coefficient between PPI and ResnikBMA or SimGIC for the baselines and the median Pearson correlation coefficient between PPI and supervised similarity obtained using XGB or RF. In bold, the best result for each dataset-SSM.

Dataset	SSM	Static					Supervised	
		BP	CC	MF	AVG	MAX	XGB	RF
PPI								
ALL1	ResnikBMA	0.545	0.486	0.353	0.569	0.545	**0.634**	0.634
	SimGIC	0.486	0.428	0.318	0.496	0.458	0.584	**0.585**
ALL3	ResnikBMA	0.523	0.446	0.330	0.545	0.507	**0.607**	0.607
	SimGIC	0.457	0.390	0.290	0.462	0.416	**0.548**	0.541

Table 11. Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{PS}}$ and ResnikBMA or SimGIC for the baselines and the median Pearson correlation coefficient between $\mathrm{Sim}_{\mathrm{PS}}$ and supervised similarity obtained using XGB or RF. In bold, the best result for each dataset-SSM.

SSM	Static						Supervised	
	HPO	BP	CC	MF	AVG	MAX	XGB	RF
ResnikBMA	0.601	0.210	0.142	0.055	0.413	0.552	**0.648**	0.648
SimGIC	0.489	0.205	0.158	0.095	0.399	0.429	**0.630**	0.629

These results show that whatever the ensemble method and taxonomic SSM, supervised similarity always achieves higher values of correlation than static similarity. Improvements over the single aspect similarities are consistent for all datasets and also clear when considering the combination baselines. However, there are some differences between the similarity proxies. For sequence proxy, it is known that the relationship between sequence similarity and semantic similarity is non-linear [54], so improvements over the best static similarity are very pronounced (up to 58% for PPI_ALL1). Regarding PFAM proxy, supervised similarity outperforms both single aspects and static combinations (average and maximum), although the improvements are more relevant for single aspects. Concerning PPI proxy, improvements over the single aspect baselines are, as expected, more pronounced for the MF baseline (between 44 and 47%). In the gene dataset, the differences between static and supervised similarity are much more accentuated for the GO single aspects.

For the sake of brevity, Tables 8, 9, and 10 only show the results for the protein datasets combining all species' protein pairs in the same group proxy. However, Tables S4, S5 and S6 of S1 File provide the results for the remaining protein datasets and shows that supervised similarity, obtained with an ensemble method coupled with a taxonomic SSM, achieves better results than static similarity. Furthermore, we verify that, also for embedding similarity, our approach can learn a combination of semantic aspects that outperforms the best static similarity.

Finally, the comparison of results using protein datasets with different levels of annotation completion can be interesting. It is known that the biological entities should be well characterized in the context of the ontology to have enough information to compute semantic similarity between them. We have used benchmark datasets with two levels of annotation completion: the datasets ending in "1" include proteins with leaf-class annotations in, at least, one aspect; the datasets ending in "3" include proteins with at least one leaf-class annotation in each aspect. Analyzing our results, we conclude that in the PFAM datasets, lower correlations were generally found for the incomplete annotation datasets, but the opposite happens in the PPI datasets. These

results are in agreement with conclusions in [55].

Conclusion

Measuring the similarity between two gene products is a fundamental aspect of today's biomedical informatics research. Biomedical ontologies and KGs provide meaningful context to data and support the comparison of biomedical entities through semantic similarity. Many KGs afford different perspectives over the data, however, existing SSMs are general-purpose and typically depend on expert knowledge to select and combine the relevant KG semantic aspects for each use case. Tailoring semantic similarity to a viewpoint of the domain or a particular use case in an automated fashion had not yet been tackled.

We have developed an approach that considers the different KG semantic aspects used to describe entities and relies on ML to learn a supervised semantic similarity. Currently, our approach includes five KG-based similarity measures based on embeddings or taxonomic semantic similarity, and eight ML methods. A comparative evaluation of the five SSMs combined with the eight ML algorithms was conducted using 21 benchmark datasets divided by species, level of annotation completion, KGs describing them, and similarity proxies employed in them. The similarity proxies rely on biological similarity measures and features - protein-protein interaction, protein function similarity, protein sequence similarity and phenotype-based gene similarity - and were used to train and test the supervised models. This evaluation elucidated which combinations of SSMs with ML algorithms are more suitable for each biological similarity.

The results showed that our approach is able to learn a supervised semantic similarity that outperforms static semantic similarity both using KG embeddings and standard taxonomic SSMs, obtaining more accurate similarity values. We specifically highlight the ability of the proposed approach to find better semantic aspect combination functions than static combinations emulating expert knowledge.

Although in this work we applied supervised ML algorithms to tailor semantic similarity to different similarity proxies, the proposed approach is versatile and can also be applied to tailor semantic similarity to a specific learning task. Consequently, there are multiple real-world tasks, where KG-based similarity is a suitable instance representation, that can benefit from this work. In future work, the impact of supervised similarity in tasks such as predicting protein-protein interactions, drug-target interactions or gene-disease associations should be evaluated.

Since interpretability has become a significant concern in ML, this issue should also be addressed in the future. Although we have employed interpretable ML algorithms, black-box models usually produced predictions with higher accuracy in our experiments, highlighting the need to explore the trade-off between performance and interpretability.

Supporting information

S1 File. Supplementary File. Supplementary tables.

Acknowledgments

CP, SS, RTS are funded by the FCT through LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020. CP and RTS are funded by project SMILAX (ref. PTDC/EEI-ESS/4633/2014), and SS by projects BINDER (ref. PTDC/CCI-INF/29168/2017) and PREDICT (ref. PTDC/CCI-CIF/29877/2017). SS

and CP are also funded by project GADgET (ref. DSAIPA/DS/0022/2018), and RTS by FCT PhD grant (ref. SFRH/BD/145377/2019).

References

1. Alshahrani M, Khan MA, Maddouri O, Kinjo AR, Queralt-Rosinach N, Hoehndorf R. Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics. 2017;33(17):2723–2730. doi:10.1093/bioinformatics/btx275.

2. The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Research. 2018;47(D1):D330–D338. doi:10.1093/nar/gky1055.

3. Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, et al. ChEBI in 2016: Improved services and an expanding collection of metabolites. Nucleic Acids Research. 2015;44(D1):D1214–D1219. doi:10.1093/nar/gkv1031.

4. Köhler S, Carmody L, Vasilevsky N, Jacobsen JO, Danis D, Gourdine JP, et al. Expansion of the Human Phenotype Ontology (HPO) knowledge base and resources. Nucleic Acids Research. 2018;47(D1):D1018–D1027. doi:10.1093/nar/gky1105.

5. Schofield PN, Sundberg JP, Sundberg BA, McKerlie C, Gkoutos GV. The mouse pathology ontology, MPATH; structure and applications. Journal of Biomedical Semantics. 2013;4(1):1–8. doi:10.1186/2041-1480-4-18.

6. Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, et al. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic acids research. 2011;39(suppl_2):W541–W545.

7. Zhang SB, Tang QR. Protein–protein interaction inference based on semantic similarity of Gene Ontology terms. Journal of Theoretical Biology. 2016;401:30–37. doi:10.1016/j.jtbi.2016.04.020.

8. Chen KH, Wang TF, Hu YJ. Protein-protein interaction prediction using a hybrid feature representation and a stacked generalization scheme. BMC Bioinformatics. 2019;20(1):308. doi:10.1186/s12859-019-2907-1.

9. Hoehndorf R, Schofield PN, Gkoutos GV. PhenomeNET: a whole-phenome approach to disease gene discovery. Nucleic Acids Research. 2011;39(18):e119. doi:10.1093/nar/gkr538.

10. Asif M, Martiniano HFMCM, Vicente AM, Couto FM. Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology. PLOS ONE. 2018;13(12):1–15. doi:10.1371/journal.pone.0208626.

11. Abdelaziz I, Fokoue A, Hassanzadeh O, Zhang P, Sadoghi M. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. Journal of Web Semantics. 2017;44:104 – 117. doi:10.1016/j.websem.2017.06.002.

12. Kastrin A, Ferk P, Leskošek B. Predicting potential drug-drug interactions on topological and semantic similarity features using statistical learning. PLOS ONE. 2018;13(5):1–23. doi:10.1371/journal.pone.0196865.

13. Harispe S, Ranwez S, Janaqi S, Montmain J. Semantic Similarity from Natural Language and Ontology Analysis. Morgan Claypool Publishers; 2015.

14. Smaili FZ, Gao X, Hoehndorf R. Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations. Bioinformatics (Oxford, England). 2018;34(13):i52—i60. doi:10.1093/bioinformatics/bty259.

15. Smaili FZ, Gao X, Hoehndorf R. OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics. 2018;35(12):2133–2140. doi:10.1093/bioinformatics/bty933.

16. Kulmanov M, Smaili FZ, Gao X, Hoehndorf R. Semantic similarity and machine learning with ontologies. Briefings in Bioinformatics. 2020;doi:10.1093/bib/bbaa199.

17. Pesquita C, Faria D, Falcão AO, Lord P, Couto FM. Semantic Similarity in Biomedical Ontologies. PLOS Computational Biology. 2009;5(7):1–12. doi:10.1371/journal.pcbi.1000443.

18. Rada R, Mili H, Bicknell E, Blettner M. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics. 1989;19(1):17–30. doi:10.1109/21.24528.

19. Pekar V, Staab S. Taxonomy Learning: Factoring the Structure of a Taxonomy into a Semantic Classification Decision. In: Proceedings of the 19th International Conference on Computational Linguistics - Volume 1. COLING '02. USA: Association for Computational Linguistics; 2002. p. 1–7. Available from: `https://doi.org/10.3115/1072228.1072318`.

20. Resnik P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1995. p. 448–453.

21. Seco N, Veale T, Hayes J. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In: Proceedings of the 16th European Conference on Artificial Intelligence. ECAI'04. NLD: IOS Press; 2004. p. 1089–1090.

22. Pesquita C, Faria D, Bastos H, Falcao A, Couto F. Evaluating GO-based semantic similarity measures. In: Proceedings of the 10th Annual Bio-Ontologies Meeting. Vienna, Austria; 2007. p. 37–40.

23. Traverso I, Vidal ME, Kämpgen B, Sure-Vetter Y. GADES: A graph-based semantic similarity measure. In: Proceedings of the 12th International Conference on Semantic Systems. SEMANTiCS 2016. New York, NY, USA: Association for Computing Machinery; 2016. p. 101–104. Available from: `https://doi.org/10.1145/2993318.2993343`.

24. Traverso-Ribón I, Vidal ME. GARUM: A Semantic Similarity Measure Based on Machine Learning and Entity Characteristics. In: Hartmann S, Ma H, Hameurlain A, Pernul G, Wagner RR, editors. Database and Expert Systems Applications - Volume 11029. Cham: Springer International Publishing; 2018. p. 169–183. Available from: `https://doi.org/10.1007/978-3-319-98809-2_11`.

25. Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering. 2017;29(12):2724–2743. doi:10.1109/TKDE.2017.2754499.

26. Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O. Translating Embeddings for Modeling Multi-Relational Data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13. Red Hook, NY, USA: Curran Associates Inc.; 2013. p. 2787–2795.

27. Wang Z, Zhang J, Feng J, Chen Z. Knowledge Graph Embedding by Translating on Hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. AAAI Press; 2014. p. 1112–1119.

28. Yang B, tau Yih W, He X, Gao J, Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases; 2015.

29. Trouillon T, Welbl J, Riedel S, Gaussier E, Bouchard G. Complex Embeddings for Simple Link Prediction. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML'16. JMLR.org; 2016. p. 2071–2080.

30. Grover A, Leskovec J. Node2vec: Scalable Feature Learning for Networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York, NY, USA: Association for Computing Machinery; 2016. p. 855–864. Available from: `https://doi.org/10.1145/2939672.2939754`.

31. Ristoski P, Paulheim H. RDF2Vec: RDF Graph Embeddings for Data Mining. In: Groth P, Simperl E, Gray A, Sabou M, Krötzsch M, Lecue F, et al., editors. The Semantic Web – International Semantic Web Conference 2016. Cham: Springer International Publishing; 2016. p. 498–514.

32. Consortium TGO. Gene Ontology Annotations and Resources. Nucleic Acids Research. 2012;41(D1):D530–D535. doi:10.1093/nar/gks1050.

33. Sousa RT, Silva S, Pesquita C. Evolving knowledge graph similarity for supervised learning in complex biomedical domains. BMC Bioinformatics. 2020;21(1):6. doi:10.1186/s12859-019-3296-1.

34. Cardoso C, Sousa RT, Köhler S, Pesquita C. A Collection of Benchmark Data Sets for Knowledge Graph-Based Similarity in the Biomedical Domain. In: Harth A, Presutti V, Troncy R, Acosta M, Polleres A, Fernández JD, et al., editors. The Semantic Web: ESWC 2020 Satellite Events. Cham: Springer International Publishing; 2020. p. 50–55.

35. El-Gebali S, Mistry J, Bateman A, Eddy SR, Luciani A, Potter SC, et al. The Pfam protein families database in 2019. Nucleic Acids Research. 2018;47(D1):D427–D432. doi:10.1093/nar/gky995.

36. Amberger JS, Bocchini CA, Schiettecatte F, Scott AF, Hamosh A. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. Nucleic Acids Research. 2015;43(Database issue):D789—98. doi:10.1093/nar/gku1205.

37. Pesquita C. Semantic Similarity in the Gene Ontology. In: Dessimoz C, Škunca N, editors. The Gene Ontology Handbook. New York, NY: Springer New York; 2017. p. 161–173. Available from: `https://doi.org/10.1007/978-1-4939-3743-1_12`.

38. Harispe S, Ranwez S, Janaqi S, Montmain J. The semantic measures library and toolkit: fast computation of semantic similarity and relatedness using biomedical ontologies. Bioinformatics. 2013;30(5):740–742. doi:10.1093/bioinformatics/btt581.

39. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825–2830.

40. Poole MA, O'Farrell PN. The Assumptions of the Linear Regression Model. Transactions of the Institute of British Geographers. 1971;(52):145–158.

41. Brown PJ, Zidek JV. Adaptive Multivariate Ridge Regression. Annals of Statistics. 1980;8(1):64–74. doi:10.1214/aos/1176344891.

42. Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory. 1967;13(1):21–27. doi:10.1109/TIT.1967.1053964.

43. Denison DGT, Mallick BK, Smith AFM. A Bayesian CART algorithm. Biometrika. 1998;85(2):363–377. doi:10.1093/biomet/85.2.363.

44. Koza JR, Koza JR. Genetic programming: on the programming of computers by means of natural selection. vol. 1. MIT press; 1992.

45. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533–536. doi:10.1038/323533a0.

46. Breiman L. Random forests. Machine learning. 2001;45(1):5–32.

47. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery; 2016. p. 785–794. Available from: `https://doi.org/10.1145/2939672.2939785`.

48. Sipper M, Fu W, Ahuja K, Moore JH. Investigating the parameter space of evolutionary algorithms. BioData Mining. 2018;11(1). doi:10.1186/s13040-018-0164-x.

49. Ikram N, Qadir MA, Afzal MT. Investigating Correlation between Protein Sequence Similarity and Semantic Similarity Using Gene Ontology Annotations. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2018;15(3):905–912. doi:10.1109/TCBB.2017.2695542.

50. Sagi O, Rokach L. Ensemble learning: A survey. WIREs Data Mining and Knowledge Discovery. 2018;8(4):e1249. doi:10.1002/widm.1249.

51. Guzzi PH, Mina M, Guerra C, Cannataro M. Semantic similarity analysis of protein data: assessment with biological features and issues. Briefings in Bioinformatics. 2011;13(5):569–585. doi:10.1093/bib/bbr066.

52. Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation. 1997;1(1):67–82. doi:10.1109/4235.585893.

53. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A Survey of Methods for Explaining Black Box Models. ACM Comput Surv. 2018;51(5). doi:10.1145/3236009.

54. Pesquita C, Faria D, Bastos H, Ferreira AEN, Falcão AO, Couto FM. Metrics for GO based protein semantic similarity: a systematic evaluation. BMC bioinformatics. 2008;9 Suppl 5:S4. doi:10.1186/1471-2105-9-s5-s4.

55. Cardoso C, Sousa RT, Köhler S, Pesquita C. A Collection of Benchmark Data Sets for Knowledge Graph-based Similarity in the Biomedical Domain. Database. 2020;2020. doi:10.1093/database/baaa078.

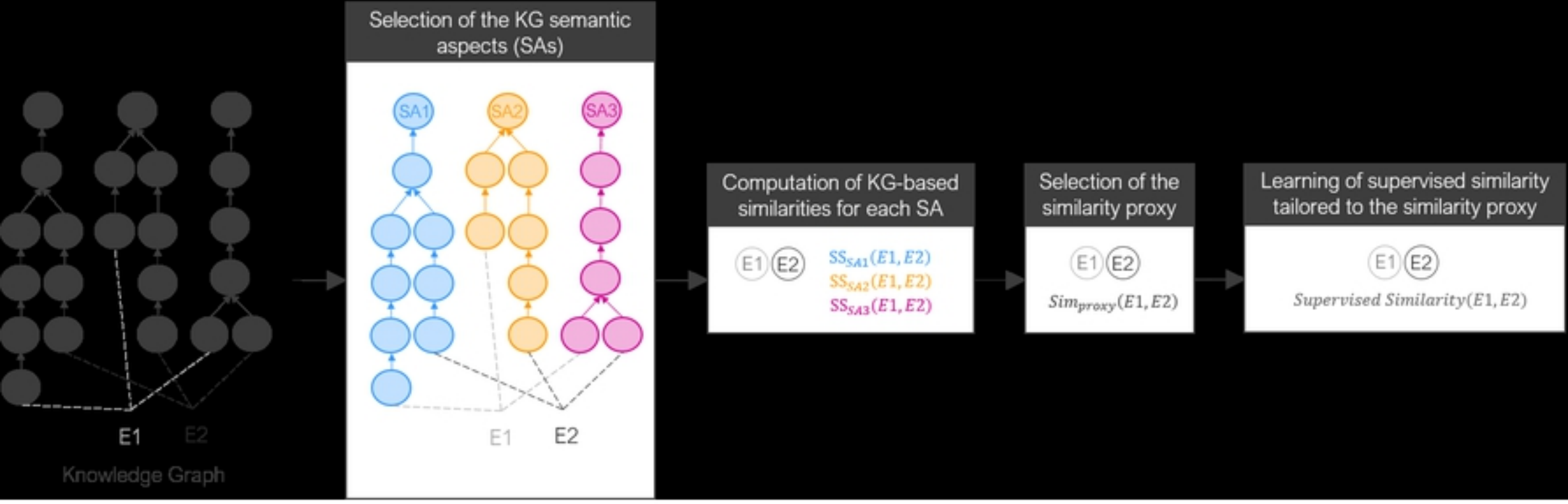
E1
E2
Knowledge Graph
Selection of the KG semantic aspects (SAs)
SA1
SA2
SA3
E1
E2
Computation of KG-based similarities for each SA
E1
E2
$SS_{SA1}(E1, E2)$
$SS_{SA2}(E1, E2)$
$SS_{SA3}(E1, E2)$
Selection of the similarity proxy
E1
E2
$Sim_{proxy}(E1, E2)$
Learning of supervised similarity tailored to the similarity proxy
E1
E2
$Supervised\ Similarity(E1, E2)$

Figure 1

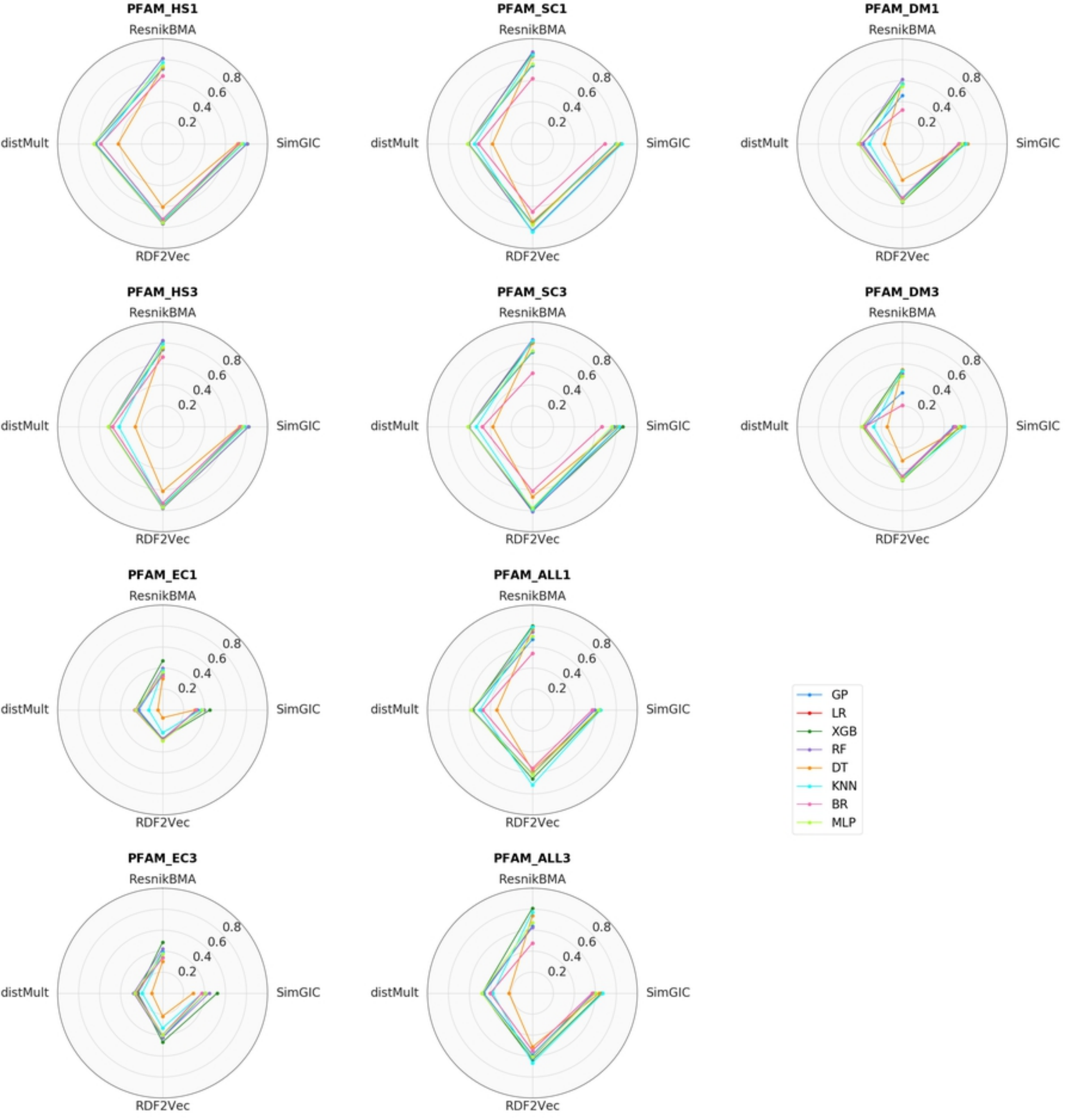
PFAM_HS1
PFAM_SC1
PFAM_DM1
PFAM_HS3
PFAM_SC3
PFAM_DM3
PFAM_EC1
PFAM_ALL1
PFAM_EC3
PFAM_ALL3
ResnikBMA
SimGIC
RDF2Vec
distMult
0.2
0.4
0.6
0.8
GP
LR
XGB
RF
DT
KNN
BR
MLP

Figure 2

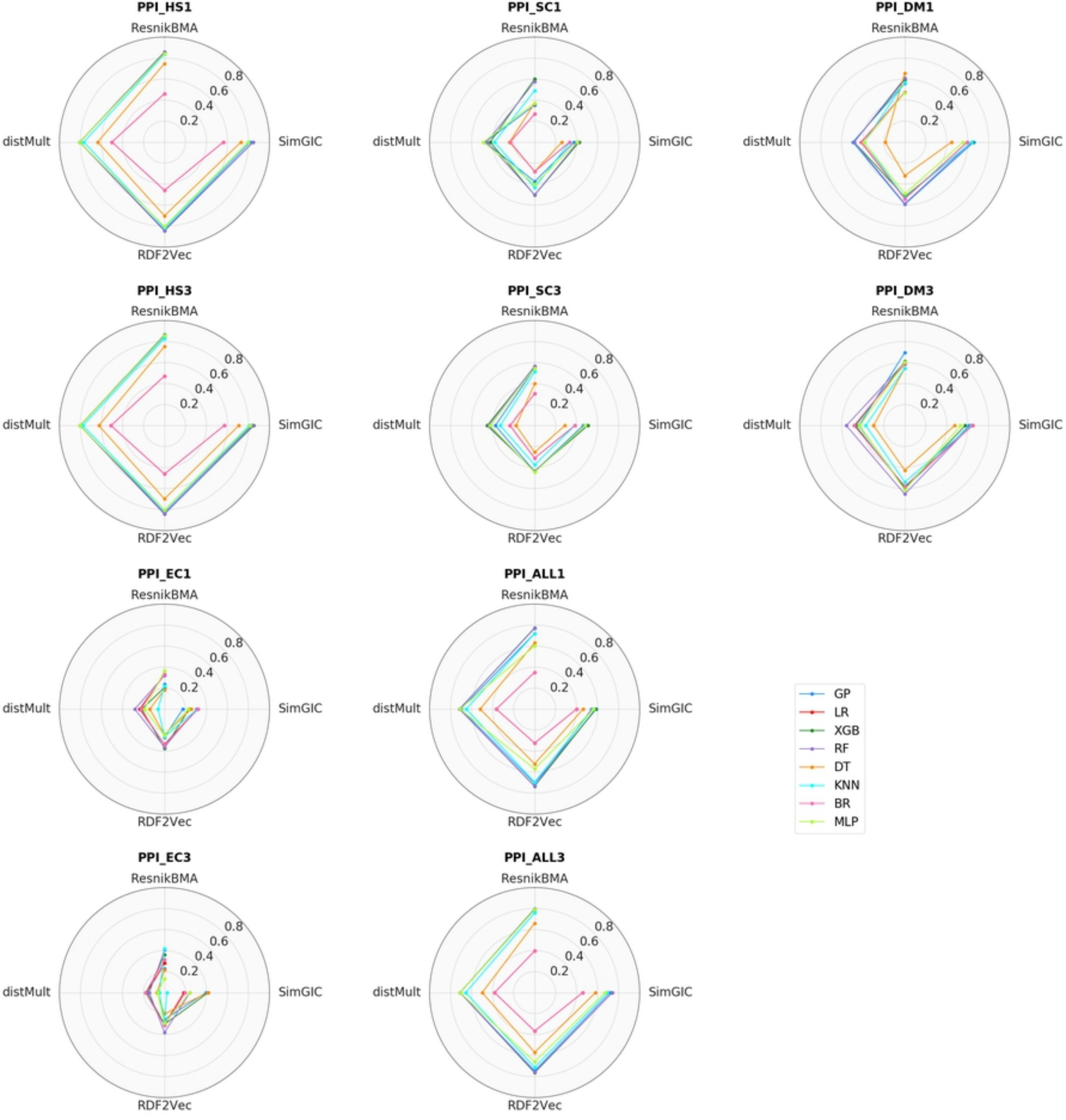
PPI_HS1
PPI_SC1
PPI_DM1
PPI_HS3
PPI_SC3
PPI_DM3
PPI_EC1
PPI_ALL1
PPI_EC3
PPI_ALL3
ResnikBMA
SimGIC
RDF2Vec
distMult
0.2
0.4
0.6
0.8
GP
LR
XGB
RF
DT
KNN
BR
MLP

Figure 3

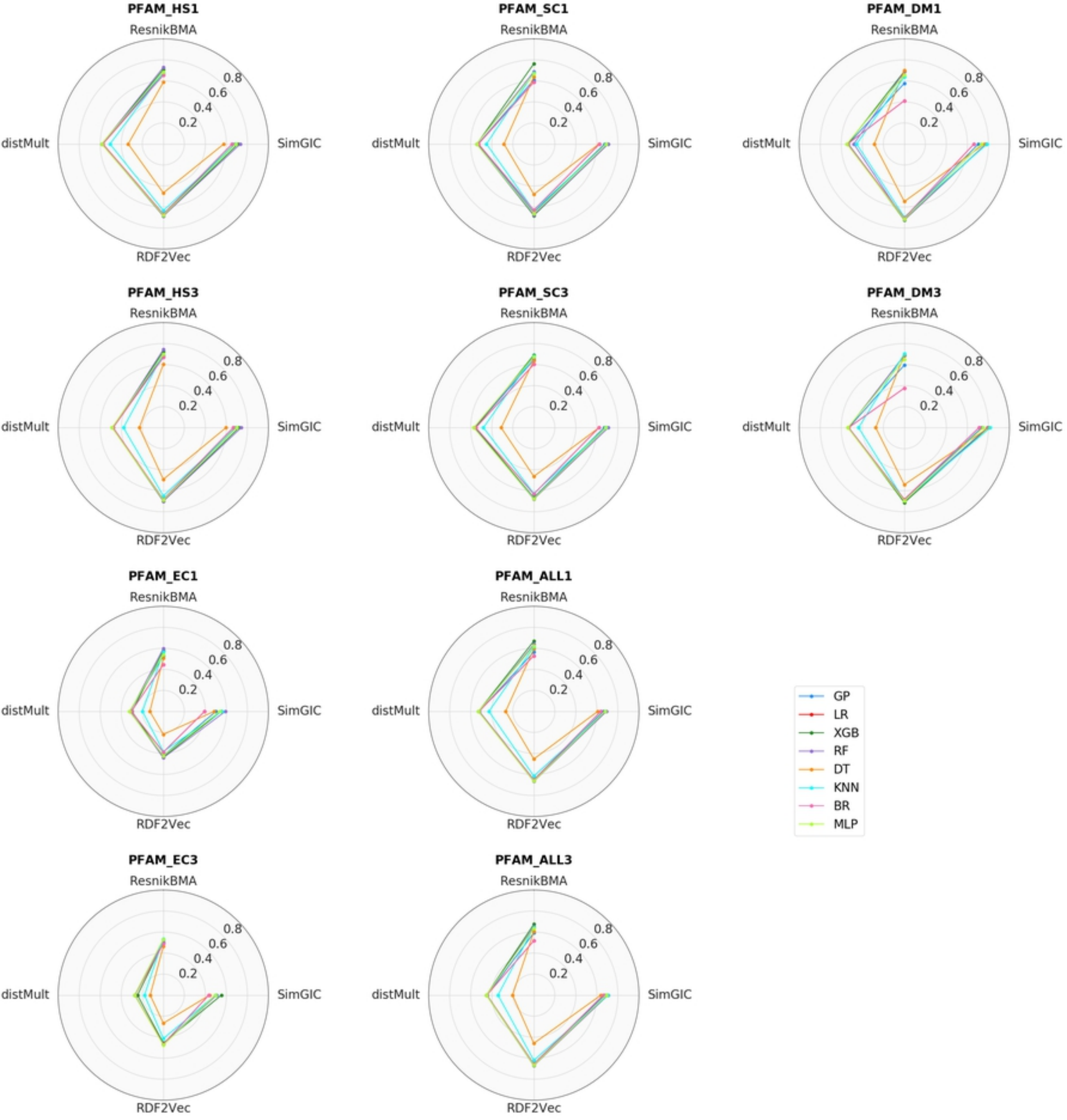
PFAM_HS1
PFAM_SC1
PFAM_DM1
PFAM_HS3
PFAM_SC3
PFAM_DM3
PFAM_EC1
PFAM_ALL1
PFAM_EC3
PFAM_ALL3
ResnikBMA
SimGIC
RDF2Vec
distMult
0.2
0.4
0.6
0.8
GP
LR
XGB
RF
DT
KNN
BR
MLP

Figure 4

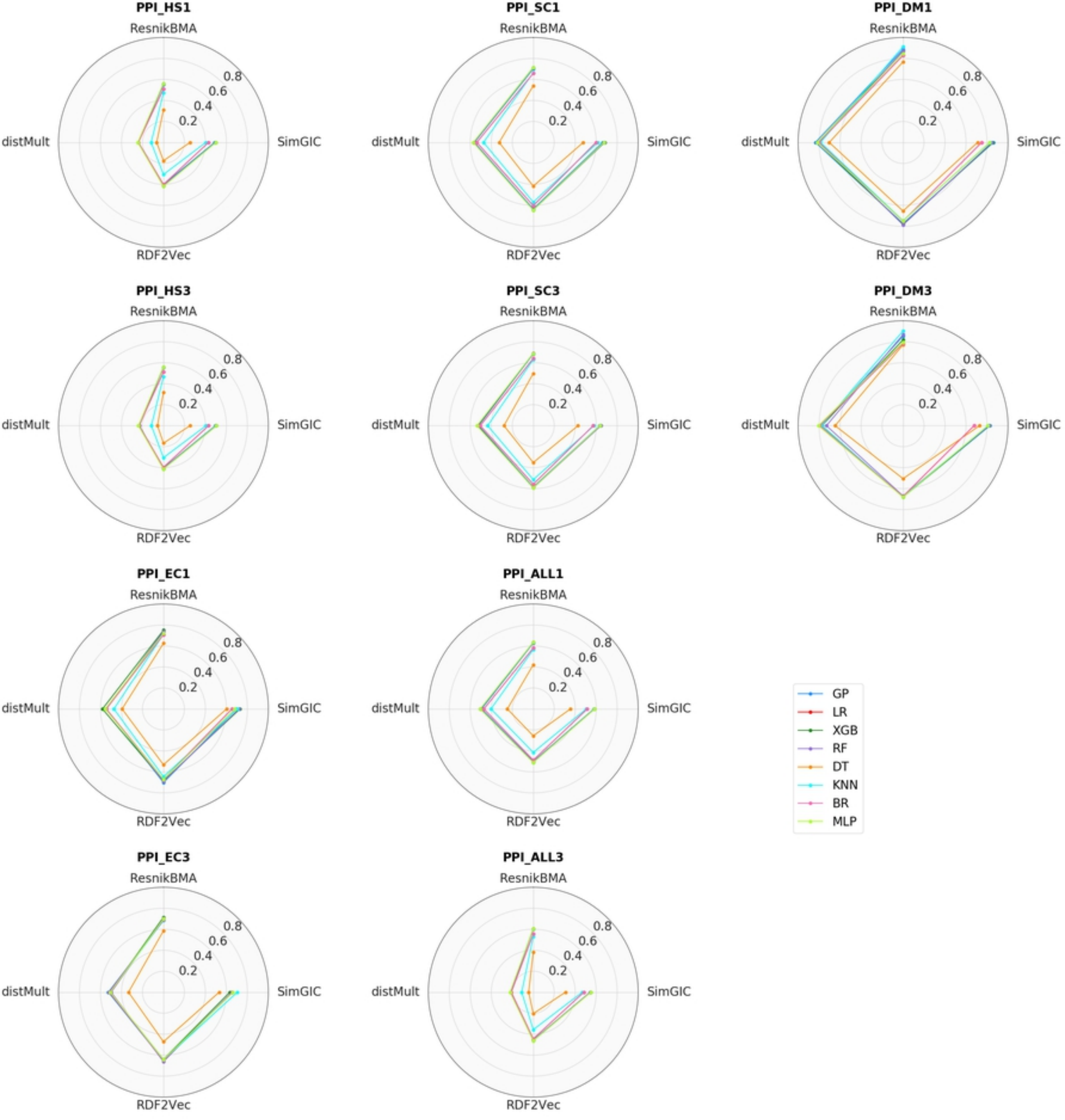
PPI_HS1
PPI_SC1
PPI_DM1
PPI_HS3
PPI_SC3
PPI_DM3
PPI_EC1
PPI_ALL1
PPI_EC3
PPI_ALL3
ResnikBMA
SimGIC
RDF2Vec
distMult
0.2
0.4
0.6
0.8
GP
LR
XGB
RF
DT
KNN
BR
MLP

Figure 5

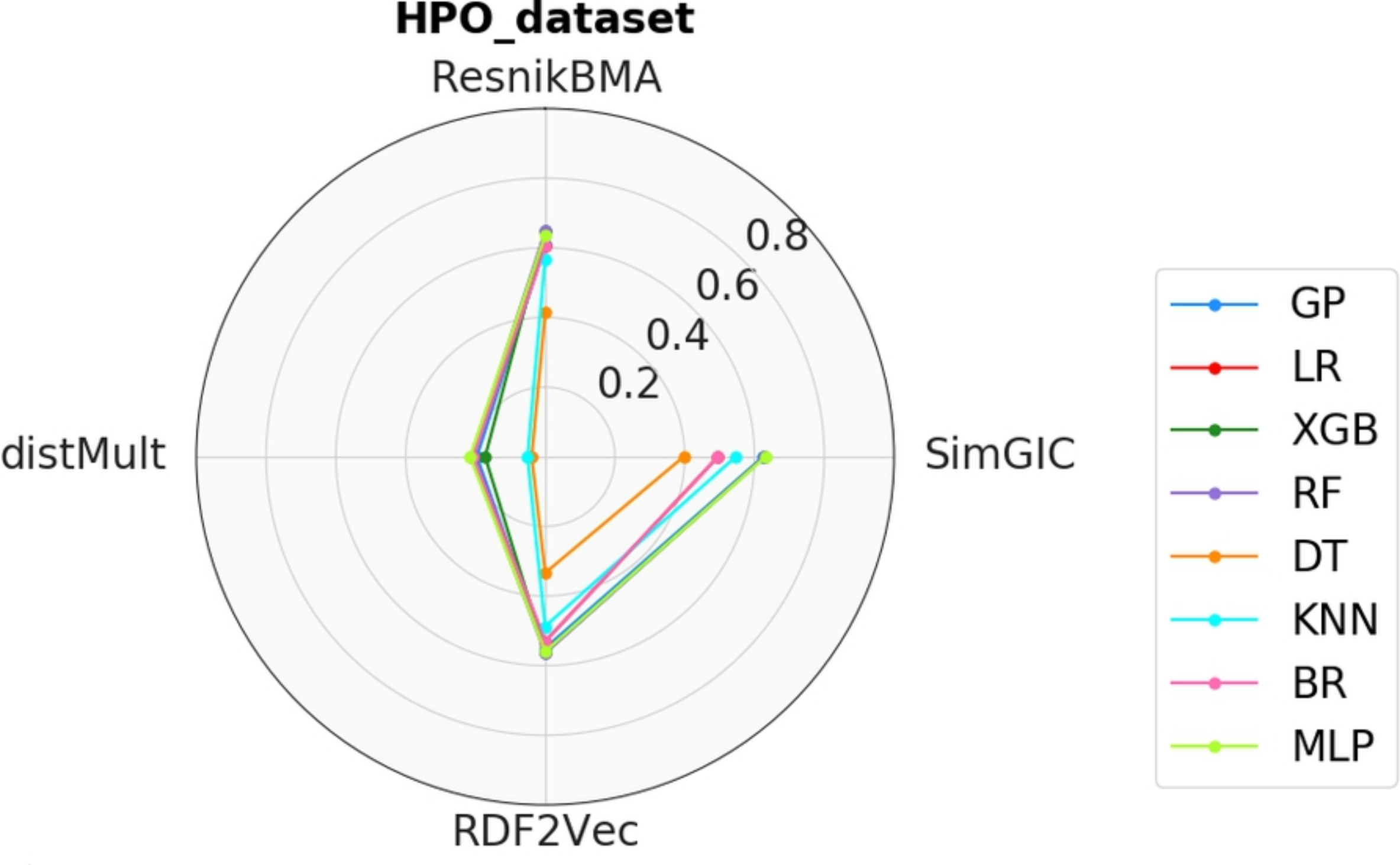
HPO_dataset
ResnikBMA
SimGIC
RDF2Vec
distMult
0.2
0.4
0.6
0.8
GP
LR
XGB
RF
DT
KNN
BR
MLP

Figure 6

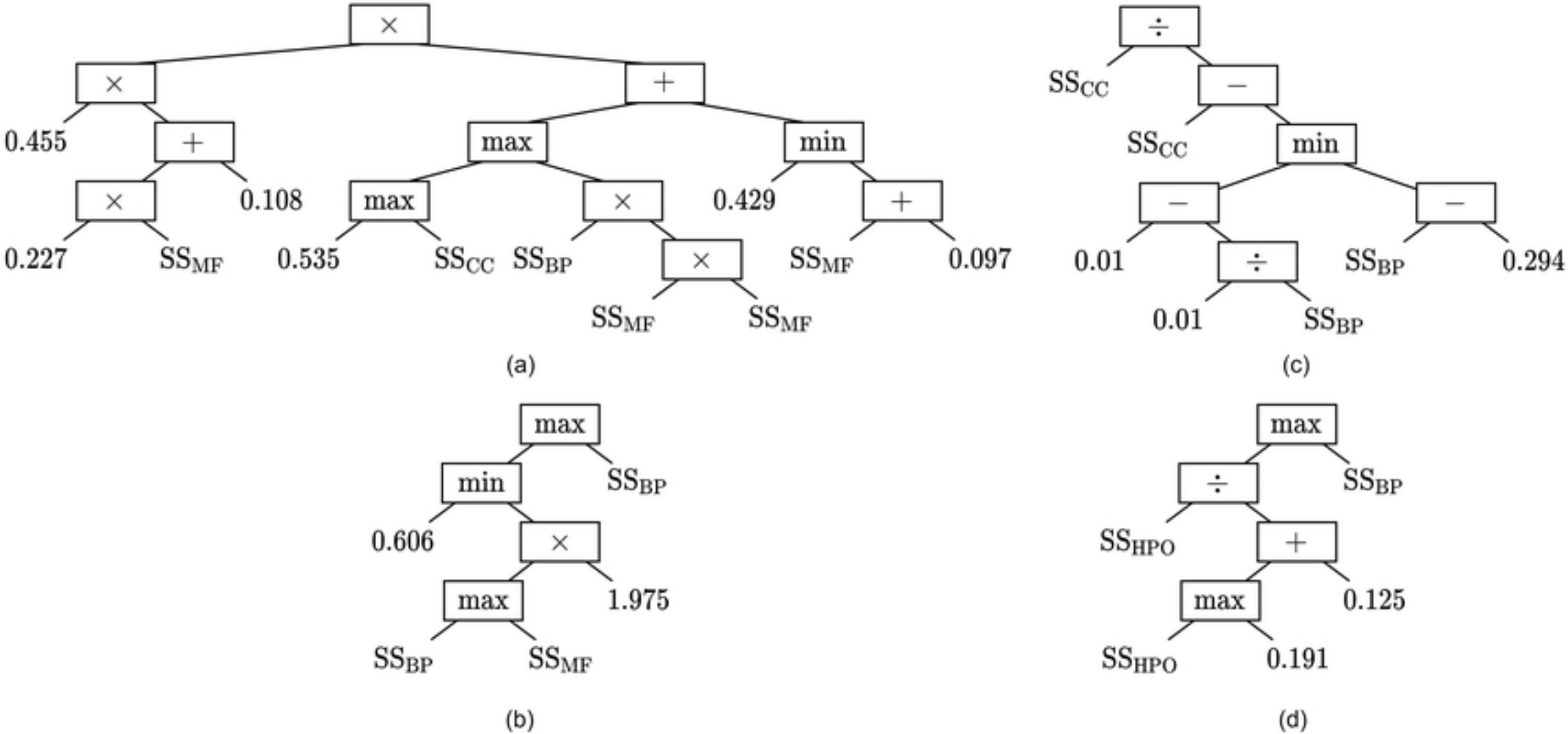
×
×
+
0.455
+
max
min
×
0.108
max
×
0.429
+
0.227
SS_MF
0.535
SS_CC
SS_BP
×
SS_MF
0.097
SS_MF
SS_MF
(a)
÷
SS_CC
−
SS_CC
min
−
−
0.01
÷
SS_BP
0.294
0.01
SS_BP
(c)
max
min
SS_BP
0.606
×
max
1.975
SS_BP
SS_MF
(b)
max
÷
SS_BP
SS_HPO
+
max
0.125
SS_HPO
0.191
(d)

Figure 7