Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

HPO paper - Supplementary Information

Marco Notaro [1], Peter N. Robinson [2,3,4,5] and Giorgio Valentini [1,*]

[1]Anacleto Lab - Dipartimento di Informatica, Universitá degli Studi di Milano, Via Comelico 39, 20135 Milano, Italy
[2]Institute for Medical and Human Genetics, Charite-Universitatsmedizin Berlin, Augustenburger Platz 1, 13353 Berlin, Germany
[3]Max Planck Institute for Molecular Genetics, Ihnestrasse 63-73, 14195 Berlin, Germany
[4]The Jackson Laboratory for Genomic Medicine, 10 Discovery Drive. Farmington, CT 06032, USA
[5]Institute for Systems Genomics, University of Connecticut, Farmington, CT 06032, USA

Abstract

1 Variants of the *TPR* algorithm for DAGs

Some variants of the basic *TPR* algorithms can be obtained by modifying the top-down step, using other strategies to achieve predictions obeying the true path rule constraints, i.e. such that $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$. For instance we could use isotonic regression strategies (Barlow and Brunk, 1972) or the Kullback-Leibler divergence.

The *Isotonic regression* method finds a set of marginal probabilities p_i that are close to the set of calibrated values $\widehat{p}_i$ obtained from the logistic regression. The euclidean distance is used as a measure of closeness. Hence, considering that the true path rule requires that $p_i \geq p_j$ when $(i,j) \in E$, this approach yields the following quadratic program:

$$\begin{aligned} &\min_{p_i, i \in I} \ \sum_{i \in I} (p_i - \widehat{p}_i)^2 \\ &s.t. \qquad p_j \leq p_i, \quad (i,j) \in E \end{aligned} \tag{1}$$

This problem is the classical isotonic regression problems that can be solved using an interior point solver or also approximated algorithm when the number of edges of the graph is too large (Burdakov *et al.*, 2006).

If we use flat base learners able to output estimates of probabilities (e.g. logistic regression or probabilistic SVMs (Platt, 1999)), a natural measure of distance between probability density functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ defined with respect to a random variable $\boldsymbol{x}$ is represented by the Kullback-Leibler divergence $D_{f_{\boldsymbol{x}}||g_{\boldsymbol{x}}}$:

$$D_{f_{\boldsymbol{x}}||g_{\boldsymbol{x}}} = \int_{-\infty}^{\infty} f(\boldsymbol{x}) \log \left(\frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \right) d\boldsymbol{x} \tag{2}$$

In the context of reconciliation methods we need to consider a discrete version of the Kullback-Leibler divergence, yielding the following optimization problem:

$$\begin{aligned} min_{\boldsymbol{p}} D_{\widehat{\boldsymbol{p}}||\boldsymbol{p}} = \ & min_{p_i, i \in I} \sum_{i \in I} \widehat{p}_i \log \left(\frac{\widehat{p}_i}{p_i} \right) \\ s.t. \qquad & p_j \leq p_i, \quad (i,j) \in E \end{aligned} \tag{3}$$

The algorithm finds the probabilities closest to the probabilities $\widehat{\boldsymbol{p}}$ obtained from logistic regression according to the Kullback-Leibler divergence and obeying the constraints that probabilities cannot increase descending the hierarchy underlying the ontology.

As outlined in the main paper, other variants able to "weigth" the contribution of the children with respect heir parent can be designed, by extending previous works on True Path Rule algorithms for tree-structured taxonomies Valentini and Re (2009); Cesa-Bianchi *et al.* (2012). FThsi can be achieved by simply substituting row 10 of the TPR-DAG algorithm (Fig. 3) with the following line of pseudocode:

$$\bar{y}_i := w\hat{y}_i + \frac{(1-w)}{|\phi_i|} \sum_{j \in \phi_i} \bar{y}_j \tag{4}$$

In this approach a weight $w \in [0,1]$ is added to balance between the contribution of the node i and that of its "positive" children.

As shown in Valentini (2011), the contribution of the descendants of a given node decays exponentially with their distance from the node itself. To enhance the contribution of the most specific nodes to the overall decision of the ensemble, a linear decaying or a constant contribution of the "positive" descendants could be considered instead:

$$\bar{y}_i := \frac{1}{1 + |\Delta_i|} (\hat{y}_i + \sum_{j \in \Delta_i} \bar{y}_j) \tag{5}$$

where

$$\Delta_i = \{ j \in desc(i) | \bar{y}_j > t_j \} \tag{6}$$

In this way all the "positive" descendants of node i provide the same contribution to the ensemble prediction $\bar{y}_i$.

Analogously, we can design "positive" descendants whose contributions to $\bar{y}_i$ decays linearly with their distance from the root. An opposite strategy could consist in an increment of the weights from bottom to top, to put more weights on predictions made on the most specific terms.

A weighting strategy could be also pursued not only considering balancing between the predictions on node i and nodes $j \in child(i)$, but inlcuding also weighting with respect to the estimated accuracy of each base learner, estimated e.g. by internal cross-validation.

2 Base learners of the hierarchical ensemble methods used in the experiments

We used a semi-supervised (*RANKS* Valentini *et al.* (2016)) and a supervised (Support Vector Machines – SVM) machine learning method to implement the base learners of the proposed hierarchical ensemble methods.

RANKS (RAnking of Nodes with Kernelized Score functions) is a semi-supervised network-based method successfully applied to gene disease prioritization (Valentini *et al.*, 2014), gene function prediction (Re *et al.*, 2012) and drug repositioning (Re and Valentini, 2013). *RANKS* adopts both a local and a global learning strategy. Local learning is accomplished through the introduction of different score functions to quantify the similarity between a gene and its neighbours. Global learning is introduced by graph kernels that capture the overall topology of the underlying biomolecular network. In principle any valid kernel function can be applied, but in our experiment we applied *RANKS* with the *average score function* and the *random walk kernel* at 1, 2 and 3 steps (Smola and Kondor, 2003), i.e. kernels able to evaluate the direct neighbours and those far away 2 and 3 steps from each gene in the GGI network.

It is worth nothing that *RANKS* returns a score and not a probability: the higher the score, the higher the likelihood that a gene belongs to a given class, but the "magnitude" of the scores may vary across different classes (Re *et al.*, 2012). To make comparable the scores computed for each class, we considered two distinct normalization procedures:

1. Normalization in the sense of the maximum: the score of each class are normalized by dividing the score values for the maximum score of that class;
2. Quantile normalization: a method originally designed for the normalization of probe intensity levels for high density oligonucleotide microarray data across multiple experiments (Bolstad *et al.*, 2003). In our case we applied quantile normalization to make comparable the scores across the different HPO terms.

SVMs were trained for each term using the R interface of the machine learning library *LiblineaR* (Fan *et al.*, 2008) with default parameter settings. Because of the high running time of SVMs we implemented a *multicore* version of *LiblineaR* using *doParallel* and *foreach* R packages.

3 Supplementary Experimental Results

[NOTA: Marco riduci la tabella alla sola Inheritance e Onset, segendo lo stesso ordine dei metodi dela tabella Organ del main paper].

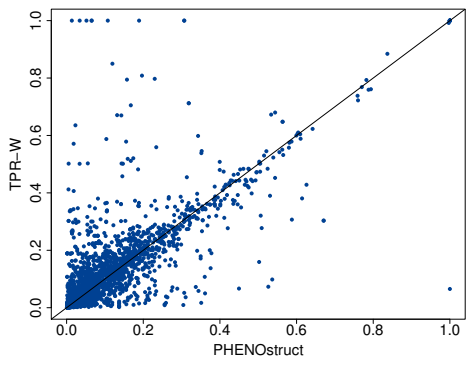

Fig. 1. AUPRC Scatter Plot. Paired AUPRC comparison between our best hierarchical TPRvariants TPR-W and PHENOstruct. We considered all the 2444 classes.

Table 1. Prediction of genes associated to HPO terms: average AUROC across terms and average F-max, Precision and Recall across genes of HTD and TPR-W ensembles and state-of-the-art methods. Best results for each metric are highlighted in bold. [NOTA: Marco, controlla che i valori di Struct->Dis->SVM e PhenoPPIOrth siano corretti]

Organ subontology				
	AUROC	F-max	Precision	Recall
RANKS	0.879	0.305	0.235	0.435
SVMs	0.747	0.419	0.360	0.501
HTD-RANKS	0.881	0.374	0.304	0.487
HTD-SVMs	0.748	0.425	0.374	0.492
TPR-RANKS	**0.885**	0.400	0.343	0.481
TPR-SVMs	0.772	**0.434**	0.378	0.511
PhenoPPIOrth	0.52	0.20	0.27	0.15
Struct->Dis->HPO	0.49	0.23	0.16	0.41
Binary SVMs	0.66	0.35	0.32	0.40
Clus-HMC-Ens	0.65	0.41	**0.39**	0.43
PHENOstruct	0.73	0.42	0.35	**0.56**

Inheritance subontology				
	AUROC	F-max	Precision	Recall
RANKS	0.911	0.560	0.429	0.806
SVMs	0.816	0.683	0.588	0.817
HTD-RANKS	0.913	0.568	0.439	0.805
HTD-SVMs	0.810	0.687	0.590	0.823
TPR-RANKS	**0.915**	0.572	0.447	0.795
TPR-SVMs	0.825	0.690	0.590	**0.823**
PhenoPPIOrth	0.55	0.12	0.16	0.10
Struct->Dis->HPO	0.46	0.11	0.07	0.25
Binary SVMs	0.72	0.69	0.62	0.78
Clus-HMC-Ens	0.73	0.73	0.64	0.84
PHENOstruct	0.74	**0.74**	**0.68**	0.81

Onset subontology				
	AUROC	F-max	Precision	Recall
RANKS	0.856	0.414	0.300	0.668
SVMs	0.737	0.466	0.369	0.631
HTD-RANKS	**0.861**	0.417	0.300	0.686
HTD-SVMs	0.743	0.458	0.365	0.616
TPR-RANKS	0.857	0.440	0.326	**0.700**
TPR-SVMs	0.746	**0.477**	**0.374**	0.664
PhenoPPIOrth	0.53	0.25	0.25	0.24
Struct->Dis->HPO	0.49	0.07	0.06	0.10
Binary SVMs	0.62	0.33	0.24	0.51
Clus-HMC-Ens	0.58	0.35	0.27	0.48
PHENOstruct	0.64	0.39	0.31	0.52

4 Consistency of the prediction: details and theorem proofs

To prove the consistency of the predictions of the *HTD-DAG* algorithm, we first introduce a property of the level function ψ. We recall here the definition of ψ, just given in the main paper. If $p(r,i)$ represents a path from the root node r and a node $i \in V$, $l\left(p(r,i)\right)$ the length of $p(r,i)$, $\mathcal{L} = \{0,1,\ldots,\xi\}$ the set of observed levels, with ξ the maximum node level, then $\psi : V \longrightarrow \mathcal{L}$ is a level function which assigns each node $i \in V$ to its level $\psi(i)$:

$$\psi(i) = \max_{p(r,i)} l\left(p(r,i)\right) \tag{7}$$

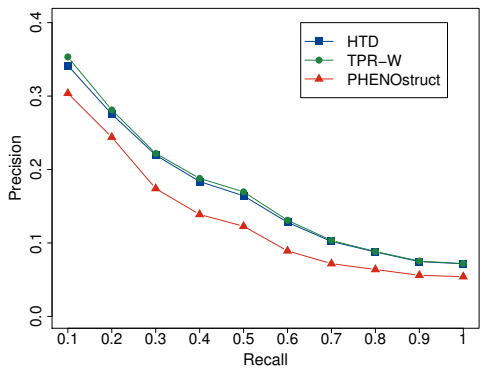

Fig. 2. Precision at different recall levels considering only the HPO terms predicted with AUROC>0.7 by TPR-W (779 terms).

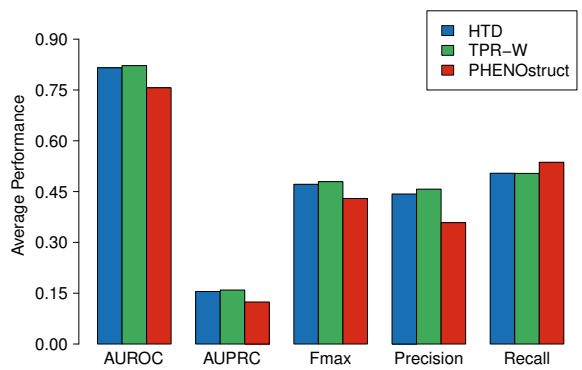

Fig. 3. Hierarchical Ensemble Methods vs PHENOstruct. Term-centric and protein-centric average performance comparison between HTD ensemble (blue bar), TPR-W ensemble variant (green bar) and PHENOstruct (red bar).

Nodes $\{i|\psi(i) = 0\}$ correspond to the root nodes, $\{i|\psi(i) = 1\}$ is the set of nodes with a maximum path length from the root (distance) equal to 1, and $\{i|\psi(i) = \xi\}$ are nodes that lie at a maximum distance ξ from the root.

To prove the consistency, we need the following lemma:

Lemma 1. *Given a DAG* $G =< V, E >$*, a level function* ψ *that assigns to each node its maximum path length from the root, it holds that* $\forall i \in V$, $\psi(j) < \psi(i)\ \forall j \in par(i)$.

Proof. The proof is based on the optimal-substructure property holding for the longest path problem in DAGs, that is a longest path between two vertices contains other longest path within it Dasgupta *et al.* (2008). Indeed, let $\bar{p}(r, i)$ be the longest path from $r = root(G)$ to node $i \in V$, and suppose that there exists $j \in par(i)$ such that $\psi(j) \geq \psi(i)$. Let $\bar{p}(r, j)$ be the path between r and j whose length is $\psi(j)$ (that is the longest path between them). Note that the path $p(r, j)$ does not contain the node i, otherwise the DAG would contain a cycle. By adding the edge (j, i) to $\bar{p}(r, j)$, we obtain a path from r to i whose length is $\psi(j) + 1 > \psi(i)$, which contradicts the hypothesis that $\bar{p}(r, i)$ is the longest path between nodes r and i.

By using Lemma 1, we can prove that the the top-down visit of the DAG obeys the true path rule:

Theorem 1. *Given a DAG* $G =< V, E >$*, a level function* ψ *that assigns to each node its maximum path length from the root and the set of* HTD-DAG *flat predictions* $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >$*, the top-down hierarchical correction of the* HTD-DAG *algorithm assures that the set of ensemble predictions* $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ *satisfies the following property:*

$$\forall i \in V,\ j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$$

Proof. For an arbitrary node $i \in V$, when it is processed by the top-down step of *HTD-DAG* algorithm, we may have two basic cases:

1. $i \in root(G)$. By applying the rule (5) we set $\bar{y}_i := \hat{y}_i$ and the property $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$ trivially holds, since $par(i) = \emptyset$.
2. $i \notin root(G)$. We may have two cases:
   a. $\hat{y}_i \leq \min_{j \in par(i)} \hat{y}_j$. In this case the rule (5) sets $\bar{y}_i := \hat{y}_i$ and hence it holds that $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.
   b. $\hat{y}_i > \min_{j \in par(i)} \bar{y}_j$. In this case by applying (5) we have $\bar{y}_i := min_{j \in par(i)} \bar{y}_j$ and hence also in this case the property $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$ holds.

Summarizing, in all cases we have that $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$, after the node i has been processed. Moreover, we note that for the currently processed node i both $\bar{y}_i$ and $\bar{y}_j,\ j \in par(i)$ will not be further changed by the "per level" top-down visit of the *HTD-DAG* algorithm. Indeed, the score $\bar{y}_i$ is modified only once, since each node is visited exactly one time (each node belongs to one and only one level of the hierarchy); moreover, since the visit is top-down, Theorem 1 implies that parent nodes are processed before their children, and hence also the scores $\bar{y}_j$ of the nodes $j \in par(i)$ will not be further changed, since $j \in par(i)$ have just been visited and their scores $\bar{y}_j$ have just been set before visiting node i. As a consequence, once a node i is visited the property $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$ will hold till to the end of the algorithm.

Finally, since the top-down step of the algorithm visits each node exactly one time, at the end of this step the property $j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$ holds for each node $i \in V$.

From Theorem 1 it is easy to prove that the consistency of the predictions holds for all the ancestors of a given node $i \in V$.

Corollary 1. *Given a DAG* $G =< V, E >$*, the level function* ψ *and the set of flat predictions* $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >$*, the* HTD-DAG *algorithm assures that for the set of ensemble predictions* $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ *the following property holds:* $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.

Proof. The corollary can be proven by "reductio ad absurdum" from Theorem 1. We suppose that for an arbitrary node i does exist a node $z \in anc(i)$ such that $\bar{y}_z < \bar{y}_i$. Let us consider all the edges (k, l) included in the path $\bar{p}(z, i)$ connecting node z with node i. Without loss of generality, we focus on a specific path, since we can repeat the same reasoning for any path connecting z with i. We claim that $\exists (k, l) \in \bar{p}(z, i)$ such that $\bar{y}_k < \bar{y}_l$, and we show this again by "reductio ad absurdum". By absurd we suppose that $\forall (k, l) \in \bar{p}(z, i)$ we have $\bar{y}_k \geq \bar{y}_l$. By transitivity along the path $\bar{p}(z, i)$, we obtain that $\bar{y}_z \geq \bar{y}_i$, but this contradicts our first hypothesis that $\bar{y}_z < \bar{y}_i$ and hence it does exist an edge $(k, l) \in \bar{p}(z, i)$ such that $\bar{y}_k < \bar{y}_l$. But for Theorem 1 it is not possible that $\bar{y}_k < \bar{y}_l$, since $k \in par(l)$. Since this contradiction comes from the assumption that does exist a node $z \in anc(i)$ such that $\bar{y}_z < \bar{y}_i$, it follows that $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.

Independently of the choice of the positive children, the following consistency theorem holds for *TPR-DAG*:

Theorem 2. *Given a DAG* $G =< V, E >$*, a set of flat predictions* $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >$ *for each class associated to each node* $i \in$

$\{1, \ldots, |V|\}$, *the* TPR-DAG *algorithm assures that for the set of ensemble predictions* $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ *the following property holds:* $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.

The proof is substantially the same of Theorem 1 and is omitted for brevity.

It is worth noting that the following properties hold for *HTD-DAG* and *TPR-DAG* algorithms:

Lemma 2. *Given a DAG* $G =< V, E >$*, a set of flat predictions* $\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >$ *for each class associated to each node* $i \in \{1, \ldots, |V|\}$*, a set of ensemble predictions* $\bar{\boldsymbol{y}} =< \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} >$ *for the* HTD-DAG *and a set of ensemble predictions* $\tilde{\boldsymbol{y}} =< \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{|V|} >$ *for the* TPR-DAG *with "positive" children selected according to (10), we have that* $\forall i \in V, \quad \tilde{y}_i \geq \bar{y}_i$.

The proof is based on the fact that the bottom-up step of *TPR-DAG* can only increment the scores $\tilde{y}_i$ with respect to the flat predictions $\hat{y}_i$. Hence the successive top-down step of the *TPR-DAG* starts from higher scores than that of the *HTD-DAG*, and the applied top-down procedure is the same for both algorithms.

A good property of *TPR-DAG* is that its sensitivity is always equal or better than that of the *HTD-DAG*:

Theorem 3. *The* TPR-DAG *ensemble algorithm with "positive" children selected according to (10) achieves always a sensitivity equal or higher than the* HTD-DAG *ensemble algorithm.*

Proof: From Lemma 2 we have that $\forall i \in V, \quad \tilde{y}_i \geq \bar{y}_i$. Hence the *TPR-DAG* ensemble algorithm, with respect to the *HTD-DAG* algorithm: a) increments or maintains equal the number of true positives; b) decreases or maintains equal the number of false negatives. By definition of the sensitivity *TPR-DAG* achieves a sensitivity equal or higher than the *HTD-DAG*.

Unfortunately there is no guarantee that the precision of *TPR-DAG* is always larger or equal than that of the *HTD-DAG* algorithm.

References

Barlow, R. and Brunk, H. (1972). The isotonic regression problem and its dual. *Journal of the American Statistical Association*, **67**(337), 140–147.

Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**(2), 185–193.

Burdakov, O., Sysoev, O., Grimvall, A., and Hussian, M. (2006). An $\mathcal{O}(n^2)$ algorithm for isotonic regression. In *Large-Scale Nonlinear Optimization*, number 83 in Nonconvex Optimization and Its Applications, pages 25–33. Springer-Verlag.

Cesa-Bianchi, N., Re, M., and Valentini, G. (2012). Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, **88**(1), 209–241.

Dasgupta, S., Papadimitriou, C., and Vazirani, U. (2008). *Algorithms*. McGraw Hill, Boston.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, **9**, 1871–1874.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.

Re, M. and Valentini, G. (2013). Network-based Drug Ranking and Repositioning with respect to DrugBank Therapeutic Categories. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **10**(6), 1359–1371.

Re, M., Mesiti, M., and Valentini, G. (2012). A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, **9**(6), 1812–1818.

Smola, A. and Kondor, I. (2003). Kernel and regularization on graphs. In B. Scholkopf and M. Warmuth, editors, *Proc. of the Annual Conf. on Computational Learning Theory*, Lecture Notes in Computer Science, pages 144–158. Springer.

Valentini, G. (2011). True Path Rule hierarchical ensembles for genome-wide gene function prediction. *IEEE ACM Transactions on Computational Biology and Bioinformatics*, **8**(3), 832–847.

Valentini, G. and Re, M. (2009). Weighted True Path Rule: a multilabel hierarchical algorithm for gene function prediction. In *MLD-ECML 2009, 1st International Workshop on learning from Multi-Label Data*, pages 133–146, Bled, Slovenia.

Valentini, G., Paccanaro, A., Caniza, H., Romero, A., and Re, M. (2014). An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine*, **61**(2), 63–78.

Valentini, G., Armano, G., Frasca, M., Lin, J., Mesiti, M., and Re, M. (2016). RANKS: a flexible tool for node label ranking and classification in biological networks. *Bioinformatics*, **32**(18).