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

OXFORD

Subject Section

A hierarchical ensemble approach for predicting genes associated to abnormal human phenotypes

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

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation:
Results:
Contact: name@bio.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In medical contexts, the word "*phenotype*" is defined as a deviation from normal morphology, physiology, or behavior (Robinson, 2012). The analysis of phenotype is essential for understanding the pathophysiology of cellular networks and plays a key role in medical research and in the mapping of disease genes (Robinson *et al.*, 2011). The Human Phenotype Ontology project (Robinson *et al.*, 2008) aims at providing a standard categorization of the abnormalities associated to human diseases and the semantic relationships between them. It's worth noting that each HPO term does not represent a disease, but rather it denotes individual signs or symptoms or other clinical abnormalities that characterize a disease. Thus, one disease is characterized by one or more HPO terms and many HPO terms are associated with multiple distinct diseases. The HPO is currently developed using the medical literature, and OMIM (Hamosh *et al.*, 2002), Orphanet (Aymé and Schmidtke, 2007) and DECIPHER (Bragin *et al.*, 2014) databases, and contains approximately $11,000$ terms and over $115,000$ annotations to hereditary diseases. The HPO is structured according to a Direct Acyclic Graph (DAG), where more general terms are found on the top levels of hierarchy and the term specificity increases moving towards the lower levels of hierarchy, i.e. from root to leaves. The HPO is governed by *true-path-rule* (also known as *annotation propagation rule*) (Robinson *et al.*, 2011): if a gene is annotated with a given functional term, then it is annotated with all the "parent" classes, and with all its ancestors in a recursive way. On the contrary if a gene is not annotated to a class, it cannot be annotated to all its offspring. According to this rule, a positive prediction for a class and a negative prediction for its parent classes are not allowed, since this violates the inclusion relationship between them.

While the problem of the prediction of the associations gene - disease has been widely investigated (Moreau and Tranchevent, 2012), the related problem of the gene - HPO term prediction has been only considered in a few studies (Kahanda *et al.*, 2015), despite no HPO term associations are known for most human genes, and the quickly growing application of the HPO to relevant medical problems (Zemojtel *et al.*, 2014; Smedley *et al.*, 2016).

"Flat" classification methods, that predict labels independently of each other, can in principle be applied to the prediction of gene - HPO terms associations (Wang *et al.*, 2013), but they may introduce significant inconsistencies in the classification, since the predictions made are unaware of the hierarchical relationship between the different phenotypes, both in human and model organism (Musso *et al.*, 2014). For instance, if we adopt the HPO to catalogue human phenotypes and we try to predict HPO terms independently of each other, we could associate to a human gene the HPO term "Hypoplasia of metatarsal bones" but not the term "Abnormalities of the metatarsal bones", introducing thus an inconsistency since "Hypoplasia of metatarsal bones" is obviously a subclass of "Abnormalities of the metatarsal bones". Besides inconsistency, flat methods may lose important a priori knowledge about the constraints of the hierarchical labeling that could enhance the accuracy of the predictions.

To properly handle the hierarchical relationships between terms that characterize ontologies (including the HPO), we can apply two main classes of structured output methods, i.e. methods able to exploit in the learning process the hierarchical structure of terms (Valentini,

2014). The first general approach is based on the kernelization of both the input and the output space through the introduction of techniques based on large margin methods for structured and interdependent output variables (Tsochantaridis *et al.*, 2005). The second general approach is based on ensemble methods able to exploit the hierarchical relationship between classes (Silla and Freitas, 2011).

In the context of HPO, Kahanda *et al.* (2015) proposed a structured output method, based on a joint kernel constructed through the product of the input and the output kernel, and showed that the proposed approach outperforms existing methods.

To our knowledge no methods based on hierarchical ensembles have been proposed in the context of structured output prediction of HPO terms associated with human genes. Indeed most of the hierarchical ensemble methods proposed in literature are conceived for tree-structured taxonomies (Valentini, 2014), and the few ones specific for DAGs have been mainly applied to the prediction of the gene and protein functions (Obozinski *et al.*, 2008; Guan *et al.*, 2008).

To fill this gap, we propose two distinct hierarchical ensemble methods (Section 2) able to provide consistent predictions of HPO terms and to scale nicely both in terms of the complexity of the taxonomy and the cardinality of the examples. The proposed approaches present several advantages with respect to structured output methods based on the joint kernelization of input and output:

1. Their computational complexity is significantly lower, while the results are competitive with respect to state-of-the art methods
2. The methods are modular, in the sense that being ensembles, can be applied with different base learners, thus allowing to enhance the predictions of any flat learning method, used as base learner in the hierarchical ensemble. Indeed hierarchical ensemble methods use as input the output of a generic flat method and are able to provide a structured prediction of the terms of the HPO according to the hierarchical relationships between terms.
3. These methods can be applied to improve the prediction of any base learner and are not constrained to apply a specific learning algorithm as it is done by kernelized method with structured output.
4. They are guaranteed to provide consistent predictions and can provide any structured prediction obeying the true path rule, differently from structured output kernelized methods that for computational complexity reasons are only able to provide a predefined set of possible structured predictions (Kahanda *et al.*, 2015).

In the next Section we describe the details of the proposed algorithm and we prove the consistency of the structured prediction of HPO terms. Then in Section 3 we present a genome-wide experimental comparison of our proposed hierarchical ensembles with state-of-the-art methods for HPO term prediction, and we show that hierarchical ensemble methods can accurately predict novel potential associations gene - HPO term.

[**Note**: We could try to predict potential novel associations for currently not annotated genes, but we need to perform the experiments ... Do you think that could be useful to provide such results?]

2 Methods

We present two algorithms, *Hierarchical Top-Down (HTD-DAG)* and *True Path Rule (TPR-DAG)* for *Directed Acyclic Graphs (DAG)*, specifically designed to exploit the DAG structure of the relationships between HPO terms to predict associations between genes and sets of HPO terms.

Both these hierarchical ensemble methods (Silla and Freitas, 2011) adopt a two-step learning strategy:

1. *Flat learning of the terms of the ontology*. In the first step each base classifier learns a specific class (HPO term). This yields a set of independent classification problems, where each base learning machine is trained to learn a specific class, independently of the other base learners.
2. *Hierarchical combination of the predictions*. In the second step the predictions provided by the trained classifiers are combined by considering the hierarchical relationships between the base classifiers modeled according to the hierarchy of the HPO.

This ensemble approach is highly modular: in principle any learning algorithm can be used to train the classifiers in the first step, and in the second step the hierarchical relationships between the HPO terms are exploited to achieve the final ensemble predictions of the set of HPO terms associated to a specific gene.

2.1 Basic notation and definitions

Let $G =< V, E >$ a Directed Acyclic Graph (DAG) with vertices $V = \{1, 2, \ldots, |V|\}$ and edges $e = (i, j) \in E, i, j \in V$. G represents the HPO taxonomy structured as a DAG, whose nodes $i \in V$ represent classes (terms) of the ontology and a directed edge $(i, j) \in E$ the hierarchical relationships between i and j: i is the parent term and j is the child term.

The set of children of a node i is denoted by $child(i)$, the set of its parents by $par(i)$, the set of its ancestors by $anc(i)$ and the set of its descendants by $desc(i)$.

A "flat multi-label scoring" predictor $f : X \to \mathbb{Y}$ provides a score $f(x) = \hat{\boldsymbol{y}}, \hat{\boldsymbol{y}} \in \mathbb{Y} = [0, 1]^{|V|}$ for a given example $x \in X$, where X is a suitable input space for the predictor f. In other words a flat predictor provides a score $\hat{y}_i \in [0, 1]$ for each node/class $i \in V$ of the DAG G:

$$\hat{\boldsymbol{y}} =< \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} > \tag{1}$$

We say that the multi-label scoring $\boldsymbol{y}$ is consistent if it obeys the *true path rule*:

$$\boldsymbol{y} \text{ is consistent } \iff \forall i \in V, j \in par(i) \Rightarrow y_j \geq y_i \tag{2}$$

From eq. 2 descends that if we predict that a protein is annotated with a given term i then to provide consistent predictions the same protein must be annotated also with all the ancestor terms of i.

In real cases it is very unlikely that a flat multi-label scoring predictor satisfies the true path rule, since by definition the predictions are performed without considering the hierarchy of the classes. Nevertheless, by adding a further label/score modification step that takes into account the hierarchy of the classes, we can modify the labeling or the scores of the flat predictors to obtain a hierarchical classifier that obeys the true path rule. More precisely, we can provide a function $h(f(x)) : \mathbb{Y} \to \mathbb{Y}$ such that the *true path rule* (2) holds for all the predictions $h(f(x)) = \bar{\boldsymbol{y}}$:

$$\forall i \in V, j \in par(i) \Rightarrow \bar{y}_j \geq \bar{y}_i \tag{3}$$

2.2 Flat learning of the terms of the ontology

At first the algorithm adopts a flat ensemble learning strategy by which each term $i \in V$ of the HPO is independently learned through a term specific predictor $f_i : X \to [0, 1]$. Accordingly, the output of the flat classifier $f : X \to \mathbb{Y}$ on the instance $x \in X$ is $f(x) = \hat{\boldsymbol{y}}$:

$$f(x) =< f_1(x),\ f_2(x),\ \ldots,\ f_{|V|}(x) >=< \hat{y}_1, \hat{y}_2,\ \ldots,\ \hat{y}_{|V|} > \tag{4}$$

To this end any supervised or semi-supervised base predictor can be used, including also flat binary classifiers, even if flat predictors f_i estimating a score or a probability that gene x belongs to a HPO term i are better suited to this task. Note that the training of per-class predictors $f_1,\ f_2,\ \ldots, f_{|V|}$ can be performed in parallel, and it is easy to achieve a linear speed-up in the number of the available processors by adopting simple parallel computational techniques.

2.3 The Hierarchical Top-Down (HTD) algorithm

The main idea behind the Hierarchical top-down algorithm *(HTD-DAG)* consists in modifying the predictions of each base learners from "top to bottom", i.e. from the least to the most specific terms by exploiting at each step the predictions provided by the less specific predictors, e.g. predictors associated to parent HPO terms. This is performed in a recursive way by transmitting the predictions from each node to the their children, and from the children to the children of the children through a propagation of the information towards the descendants of each node of the ontology. For instance in Fig. 1a the information can flow along the path traversing nodes $1, 5, 6, 7$ or $1, 3, 7$ (numbers represent different HPO terms), and a prediction for e.g. the node 5 depends on the predictions performed by the base learners for the parent nodes 4, 1 and 3. This operating mode of the ensemble is performed neatly from the top to the bottom nodes (Fig. 1a).

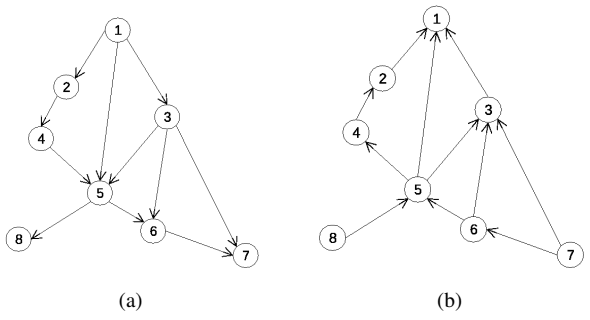


Fig. 1. Flow of information in hierarchical ensembles. (a) Top-down flow (b) Bottom-up flow. See text for more explanations.

More precisely, the *HTD-DAG* algorithm modifies the flat scores according to the hierarchy of a DAG through a unique run across the nodes of the graph. For a given example $x \in X$, the flat predictions $f(x) = \hat{\boldsymbol{y}}$ are hierarchically corrected to $\bar{\boldsymbol{y}}$, by per-level visiting the nodes of the DAG from top to bottom, according to the following simple rule:

$$\bar{y}_i := \begin{cases} \hat{y}_i & \text{if} \quad i \in root(G) \\ \min_{j \in par(i)} \bar{y}_j & \text{if} \quad \min_{j \in par(i)} \bar{y}_j < \hat{y}_i \\ \hat{y}_i & \text{otherwise} \end{cases} \tag{5}$$

The node levels correspond to their maximum path length from the root. If $p(r, i)$ represents a path from the root node r and a node $i \in V$, $l\,(p(r, i))$ 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 $\psi(i)$:

$$\psi(i) = \max_{p(r,i)} \; l\,(p(r, i)) \tag{6}$$

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.

Fig. 2 shows the pseudo code of the second step of *HTD-DAG* algorithm, by which the flat predictions $\hat{\boldsymbol{y}}$ computed in the first step are combined and updated according to top-down per-level traversal of the DAG.

The block A of the algorithm (row 1) computes the maximum distance of each node from the root; to this end the classical Bellman-Ford algorithm or the methods based on the Topological Sorting algorithm can be applied Cormen *et al.* (2009).

The block B of the algorithm implements a per-level top-down visit of the graph (rows $2 - 13$). Starting from the children of the root (level 1) for each level of the graph the nodes are processed and the hierarchical top-down correction of the flat predictions $\hat{y}_i$, $i \in \{1, \ldots, |V|\}$ is performed according to (5) thus obtaining the *HTD-DAG* ensemble prediction $\bar{y}_i$. More precisely, the nested loops starting respectively at line 04 and 06 ensure that nodes are processed by level in an increasing order. Lines 07–11, which implement (5), perform the hierarchical correction of the flat predictions $\hat{y}_i$, $i \in \{1, \ldots, |V|\}$. The algorithm ends when, at the last iteration of the external loop (lines 04–13), nodes at distance ξ from the root are processed, and provides as output the corrected predictions $\bar{\boldsymbol{y}}$.

The complexity of block A is $\mathcal{O}(|V| + |E|)$ (if the Topological Sort algorithm is used to implement *ComputeMaxDist*), while it is easy to see that the complexity of block B (rows $3 - 13$) is $\mathcal{O}(|V| + |E|)$. Hence the overall complexity of the top-down step of *HTD-DAG* is $\mathcal{O}(|V| + |E|))$, that is linear in the number of vertices for sparse graphs.

2.4 Hierarchical True Path Rule algorithm for DAGs (TPR-DAG)

The *HTD* algorithm takes only into account the predictions of the parent and recursively of the ancestor nodes. By considering the opposite flow of information "from bottom to top", we can construct the prediction of the ensemble by using the predictions made by the children nodes and recursively by the descendant more specific nodes. For instance in Fig. 1b) a possible flow of information could be along the path $8, 5, 4, 2, 1$ or $7, 6, 5, 1$, and the prediction of the ensemble for e.g. node 3 depends on children nodes $5, 6$ and 7. The proposed True Path Rule for DAG *TPR-DAG* adopts this bottom-up flow of information, to take into account the predictions of the most specific HPO terms, but also the opposite flow from top to bottom to consider the predictions of the most specific terms. This second algorithm can be considered a DAG extension of the *TPR* algorithm, originally proposed for tree-structured taxonomies (Valentini, 2011). The main difference with respect to the original tree-version consists in the fact that the per-level traversal of the DAG is now performed through two completely distinct steps: a bottom-up per level visit of the graph followed by a top-down visit, while in the original tree-version the per-level traversal is performed in an "interleaved" fashion (that is the bottom-up and top-down traversal are alternated at each level Valentini (2011)). In the DAG version the separation of the bottom-up and top-down steps is necessary

Fig. 2. The Hierarchical Top-Down algorithm for DAGs (HTD-DAG)

```
Input:
- G =< V, E >
- ŷ =< ŷ_1, ŷ_2, ..., ŷ_|V| > (flat predictions)
begin algorithm
01:    A. dist := ComputeMaxDist (G, root(G))
02:    B. Per-level top-down visit of G:
03:       ȳ_root(G) := ŷ_root(G)
04:       for each d from 1 to ξ do
05:          N_d := {i|dist(i) = d}
06:          for each i ∈ N_d do
07:             x := min_{j∈par(i)} ȳ_j
08:             if (x < ŷ_i)
09:                ȳ_i := x
10:             else
11:                ȳ_i := ŷ_i
12:          end for
13:       end for
end algorithm
Output:
- ȳ =< ȳ_1, ȳ_2, ..., ȳ_|V| >
```

to assure the true path rule consistency of the predictions (see Section 2.5 for details).

Fig. 3. Hierarchical True Path Rule algorithm for DAGs (TPR-DAG)

```
Input:
- G =< V, E >
- V = {1, 2, ..., |V|}
- ŷ =< ŷ_1, ŷ_2, ..., ŷ_|V| >,   ŷ_i ∈ [0, 1]
begin algorithm
01:    A. Compute ∀i ∈ V the max distance from root(G):
02:        E' := {e'|e ∈ E, e' = −e}
03:        G' :=< V, E' >
04:        dist := Bellman.Ford(G', root(G'))
05:    B. Per-level bottom-up visit of G:
06:        for each d from max(dist) to 0 do
07:           N_d := {i|dist(i) = d}
08:           for each i ∈ N_d do
09:              Select the set φ_i of "positive" children
10:              ȳ_i := 1/(1+|φ_i|) (ŷ_i + Σ_{j∈φ_i} ȳ_j)
11:           end for
12:        end for
13:    C. Per-level top-down visit of G:
14:        ŷ := ȳ
15:        for each d from 1 to max(dist) do
16:           N_d := {i|dist(i) = d}
17:           for each i ∈ N_d do
18:              x := min_{j∈par(i)} ȳ_j
19:              if (x < ŷ_i)
20:                 ȳ_i := x
21:              else
22:                 ȳ_i := ŷ_i
23:           end for
24:        end for
end algorithm
Output:
- ȳ =< ȳ_1, ȳ_2, ..., ȳ_|V| >
```

The other main difference consists in the way the levels are computed: in this new DAG version the levels are constructed according to the maximum distance from the root, since this guarantees that in the top-down step all the ancestor nodes have been processed before their descendants, thus assuring the true path rule consistency of the predictions.

Similarly to the tree-based version the *TPR* algorithm, the basic *TPR-DAG* adopts a per-level bottom-up traversal of the DAG, starting from the nodes most distant (in the sense of the maximum distance) from the root to correct the flat predictions $\hat{y}_i$:

$$\bar{y}_i := \frac{1}{1+|\phi_i|}(\hat{y}_i + \sum_{j \in \phi_i} \bar{y}_j) \tag{7}$$

where ϕ_i are the "positive" children of i.

Here we consider only positive predictions of the children to obey the true path rule. Indeed, according to this rule, we may have a gene annotated to a term t, but not annotated to a terms $s \in child(t)$. Hence if we have a negative prediction for the terms s it is not meaningful to use this prediction to predict the term t. Different strategies to select the "positive" children ϕ_i can be applied, according to the usage of a specific threshold to separate positive form negative examples:

1. *Constant Threshold (CT) strategy.* For each node the same threshold $\bar{t}$ is a priori selected: $t_j = \bar{t}, \quad \forall j \in V$. In this case $\forall i \in V$ we have:

$$\phi_i := \{j \in child(i) | \bar{y}_j > \bar{t}\} \tag{8}$$

   For instance if the predictions represent probabilities it could be meaningful to a priori select $\bar{t} = 0.5$.
2. *Adaptive Threshold (AT) strategy.* The threshold is selected to maximize some performance metric $\mathcal{M}$ estimated on the training data, e.g. the F-score or the AUC. In other words the threshold is selected to maximize some measure of accuracy of the predictions $\mathcal{M}(j,t)$ on the training data for the class j with respect to the threshold t. The corresponding set of positives $\forall i \in V$ is:

$$\phi_i := \{j \in child(i) | \bar{y}_j > t_j^*, t_j^* = \arg\max_t \mathcal{M}(j,t)\} \tag{9}$$

   For instance t_j^* can be selected from a set of $t \in (0,1)$ through internal cross-validation techniques.
3. *Threshold Free (TF) strategy.* A different solution, that does not require an a priori or an experimentally selected threshold could consists in choosing those children that can increment the score of the node i (that is positive nodes are those that achieve a higher score than that of their parent):

$$\phi_i := \{j \in child(i) | \bar{y}_j > \hat{y}_i\} \tag{10}$$

According to the above positives selection strategies we can derive three different algorithmic variants of the basic *TPR*:

1. *TPR-CT*: *TPR* with constant threshold, corresponding to the above positives selection strategy a)
2. *TPR-AT*: *TPR* with adaptive thresholds, corresponding to the selection strategy b)
3. *TPR-TF*: *TPR* threshold-free, corresponding to the selection strategy c)

With all the variants of the basic *TPR*, predictions are bottom-up propagated, thus moving positive predictions towards the parents and recursively towards the ancestors of each node.

Fig. 3 shows the high-level pseudo-code of the *TPR-DAG* algorithm. The first four rows compute the maximum distance of each node from the root, using the Bellman-Ford algorithm. The block B (rows 5-12) performs a bottom-up visit of the graph and updates the predictions $\bar{y}_i$ of the *TPR-DAG* ensemble according to eq. 7 and one of the positives selection strategies described previously in this section. Note that this step propagates the "positive" predictions from bottom to top of the DAG, but does not assure the true path rule consistency of the predictions. This is accomplished by the third block (rows $13-24$) that simply executes a hierarchical top-down step, in the same way of the *HTD-DAG* algorithm.

The complexity of the *TPR-DAG* algorithm is quadratic in the number of nodes for the block A (but can be $\mathcal{O}(|V|+|E|)$ if the Topological Sort algorithm is used instead). It is easy to see that the complexity is $\mathcal{O}(|V|)$ for both the B and C blocks when graphs are sparse.

A *TPR-DAG* variant similar to the weighted True Path Rule algorithms for tree-structured taxonomies Cesa-Bianchi *et al.* (2012) can be designed for DAGs. The *TPR-W* can be obtained by substituting row 10 of the *TPR-DAG* algorithm (Fig. 3) with the following line of pseudocode:

$$\bar{y}_i := w\hat{y}_i + \frac{(1-w)}{|\phi_i|} \sum_{j \in \phi_i} \bar{y}_j \tag{11}$$

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. If $w = 1$ no

weight is attributed to the children and the *TPR-DAG* reduces to the *HTD-DAG* algorithm, since in this way only the prediction for node i is used in the bottom-up step of the algorithm. If $w = 0$ only the predictors associated to the children nodes "vote" to predict node i. In the intermediate cases for increasing values of w we attribute more importance to the predictor for the node i with respect to its children and of course we put more weights on children for decreasing values of w.

Other variants of the *TPR-DAG* algorithm can be designed by using at each step all the descendants of a given node instead of simply its children, or taking into account the accuracy of each base learner at each step of the weighted combination of the predictor (see Supplementary Information for more details).

2.5 Consistency of the predictions

To obtain consistent predictions, we need to visit the hierarchy per level in the sense of the maximum and not of the minimum distance from the root.

Fig. 4 provides a simple example to show this fact. Indeed looking

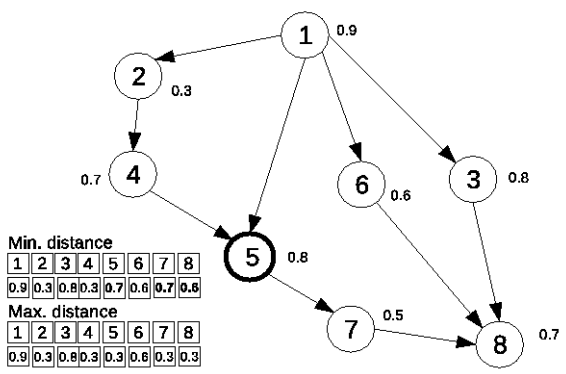


Fig. 4. Levels defined in terms of the minimum distance from the root (node 1) lead to inconsistent predictions. The small numbers close to nodes correspond to the $\hat{y}_i$ scores of the flat predictions. The Hierarchical top-down scores obtained respectively by crossing the levels according to the minimum and the maximum distance from the root are shown in the bottom-left. Scores in boldface represent inconsistent predictions.

at the *HTD-DAG* scores obtained respectively with the minimum and maximum distance from the root (bottom-left of Fig. 4), we see that only the maximum distance preserves the consistency of the predictions. For instance, focusing on node 5, by traversing the DAG levels according to the minimum distance from the root, we have that the level of node 5 is 1 ($\psi^{min}(5) = 1$) and in this case by applying the *HTD* rule (5) the flat score $\hat{y}_5 = 0.8$ is wrongly modified with the *HTD* ensemble score $\bar{y}_5 = 0.7$. If we instead traverse the DAG levels according to the maximum distance from the root, we have $\psi^{max}(5) = 3$ and the *HTD* ensemble score is correctly set to $\bar{y}_5 = 0.3$. In other words at the end of the *HTD*, by traversing the levels according to the minimum distance we have $\bar{y}_5 = 0.7 > \bar{y}_4 = 0.3$, that is a child note has a score larger than that of its parent, and the true path rule is not preserved. On the contrary by traversing the levels according to the maximum distance we achieve $\bar{y}_5 = 0.3 \leq \bar{y}_4 = 0.3$ and the true path rule consistency is assured. This is due to the fact that by adopting the minimum distance when we visit node 5, node 4 has not just been visited, and hence the value 0.4 has not been transmitted by node 2 to node 4; on the contrary if we visit the DAG according to the maximum distance all the ancestors of node 5 (including node 4) have just been visited and the score 0.4 is correctly transmitted to node 5 along the path $2 \rightarrow 4 \rightarrow 5$.

We proved the consistency of the predictions of both the *HTD-DAG* and *TPR-DAG* through the following theorems:

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:*

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

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:* $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.

Independently of the choice of the positive children (Section 2.4), 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:* $\forall i \in V,\ j \in anc(i) \Rightarrow \bar{y}_j \geq \bar{y}_i$.

Finally 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.*

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

All the proofs and details about the above theorems are available in the Supplementary Information.

3 Experimental results

We performed two different sets of experiments to compare our proposed hierarchical ensemble approach with state-of-the-art methods for the prediction of abnormal human phenotypes structured according to the HPO. In the first set of experiments (Section 3.1) we compared *HTD* and *TPR* ensembles with state-of-the-art methods using the same data and experimental set-up adopted by Kahanda *et al.* (2015). In the second set of experiments (Section 3.2) we evaluated the ability of our proposed hierarchical ensemble methods to predict newly annotated genes of the April 2016 HPO release, by using annotations of a previous release (January 2014).

3.1 Prediction of Human Phenotype Ontology terms

We compared the hierarchical ensemble methods for DAGs (*HTD*, Section 2.3) and *TPR* and its variants, (Section 2.4) against several state-of-the-art and baseline methods:

- PHENOstruct, a state-of-the art joint-kernel structured support vector machine approach (Kahanda *et al.*, 2015);
- Clus-HMC-Ens, a state-of-the art Hierarchical Multilabel classification (HMC) based on decision tree ensemble (Schietgat *et al.*, 2010);

- *SSVM* $\rightarrow$ *disease* $\rightarrow$ *HPO method*, an indirect two-step method that first predicts gene-disease associations and then maps them to HPO terms using the associations available on the HPO website (Kahanda *et al.*, 2015);
- *PhenoPPIOrth*, a computational tool that can predict a set of OMIM diseases for given human genes using protein-protein interaction and orthology data and then maps the predicted OMIM terms to HPO terms by direct mapping (Wang *et al.*, 2013).
- Probabilistic support vector machines (SVMs) (Platt, 1999);
- RANKS, a semi-supervised method base on kernelized score functions (Valentini *et al.*, 2016), resulted one of the top-ranked methods in the recent CAFA2 challenge for HPO term prediction (Jiang *et al.*, 2016).

We used both 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 (see Supplementary Information for more details).

We applied the same experimental set-up adopted by Kahanda *et al.* (2015) to provide a fair comparison with previously proposed methods (Schietgat *et al.*, 2010; Wang *et al.*, 2013; Kahanda *et al.*, 2015).

3.1.1 Data

Data sources. We used the same version of the the STRING (v. 9.1, Franceschini *et al.* (2013)) and BioGRID (v. 3.2.106, Chatr-Aryamontri *et al.* (2013)) databases for a fair comparison with the results presented by Kahanda *et al.* (2015). More precisely we downloaded physical and genetic experimental interactions relative to 4970 proteins from BioGRID 3.2.106, and the integrated protein-protein interaction and functional association data for $18,172$ proteins from STRING 9.1. From the same STRING website we downloaded also the protein aliases file to map proteins to genes. [NOTA: SPECIFICARE FRA QUALI IDENTIFICATORI HAI EFFETTUTATO IL MAPPING].

In STRING, each protein-protein interaction is annotated with a "score" ranging from 0 to 1 that indicates the confidence level of an interaction according to the available evidence and we used the full data available for *H. sapiens*.

We used also 4 different sources of gene annotation (GO BP, MF, CC and OMIM) to construct 4 sets of similarity features between proteins through the classical Jaccard index, thus resulting in 4 additional functional networks. The resulting $n = 6$ networks have been also integrated by averaging the edge weights w_{ij}^d between the genes i and j, of each network $d \in \{1, n\}$, after normalizing their weights in the same range of values $w_{ij}^d \in [0, 1]$ (*Unweighted Average* (UA) network integration, Valentini *et al.* (2014)):

$$\bar{w}_{ij} = \frac{1}{n} \sum_{d=1}^{n} w_{ij}^d \tag{12}$$

More details about the data are available in the Supplementary Information.

HPO DAG and Annotations. Following the same experimental set-up of Kahanda *et al.* (2015), we considered separately the three main subontologies of the HPO (January 2014 release): *Organ Abnormality*, *Mode of Inheritance* and *Onset and Clinical Course*. *Organ Abnormality* is the main ontology and includes terms related to clinical abnormalities. *Mode of Inheritance* is a relatively small ontology and describes the inheritance pattern of the phenotypes. *Onset and Clinical Course* contains classes that describe typical modifiers of clinical symptoms, as the speed of progression, and the variability or the onset. For the sake of simplicity for the rest of the paper we refer to these subontologies respectively as *Organ*, *Inheritance* and *Onset*. We pruned the HPO terms having less than 10 annotations in the January 2014 release. For the experiments we report the results obtained with STRING (the most informative source of information, as reported by the analysis performed by Kahanda *et al.* (2015)) and the UA integrated network. The general characteristics of the resulting DAGs are reported in Table 1.

Table 1. HPO DAG terms and between-terms relationship for each HPO subontology referred to the STRING and UA Integrated gene networks.

Subontology	HPO Features	STRING Network	UA Network
Organ	*nodes number*	2134	2615
	edges number	2154	2641
Inheritance	*nodes number*	13	13
	edges number	12	12
Onset	*nodes number*	23	24
	edges number	22	23

3.1.2 Experimental set-up and performance metrics

The generalization performance of the compared methods have been evaluated through a classical 5-fold cross-validation procedure, according to Kahanda *et al.* (2015).

We applied both *gene-centric* measures to provide metrics that depend on how much we can predict the HPO terms associated to each specific gene, and *term-centric* measures that focus on the accuracy achieved on each specific term of the ontology. As *gene-centric* measure we used the hierarchical F-score, i.e. the harmonic mean of the hierarchical precision and recall computed separately for each specific gene. More precisely we applied the F_{max} measure, i.e. the maximum hierarchical F-score achievable by "a posteriori" setting the optimal decision threshold (Jiang *et al.* (2016), see Supplementary Information for more details) As *term-centric* measures we computed the classical Area Under the Receiving Operating Characteristic (AUROC). For both measures, by averaging across genes or terms, we may have an overall picture of the prediction performance of the methods.

3.1.3 Results and Discussion

We implemented five different *TPR* variants: *TPR-TF* (True Path Rule Threshold-Free), *TPR-CT* (True Path Rule with Constant Threshold), *TPR-W* (True Path Rule Weighted), *TPR-WT* (True Path Rule Weighted with Threshold) and *TPR-D* (True Path Rule Descendants) (see Section 2.4 and Supplementary Information for more details). [NOTA: Marco, controlla che le sigle e lo stile dei caratteri siano coerenti in TUTTO il paper e le Supplementary].

Here we report only the results obtained with the STRING network (Section 3.1.1) obtained with *HTD* and *TPR-W* ensemble methods, while the detailed results obtained with the other variants of the *TPR* algorithm as well as those obtained with the *UA* integrated network are available in the Supplementary Information. Indeed the best results have been obtained with the STRING network; this is not so surprising since STRING already provides a careful integration of different sources of information, and additional sources do not seem to add further valuable information for the prediction of HPO terms.

Table 2 summarizes the results achieved by the proposed hierarchical ensemble methods *HTD* and *TPR* with the main *Organ* subontology, using as base learner *RANKS* (*TPR-RANKS* and *HTD-RANKS*) and a binary linear SVM (*TPR-SVM* and *HTD-SVM*). The results have been compared with those achieved by state-of-the-art methods and the two flat methods used as base learner by the hierarchical ensembles (*RANKS* and *SVM*). *HTD* and *TPR* ensembles significantly outperform state-of-the-art methods in terms of term-centric measures (Wilcoxon rank sum test, p-value$< 10^{-6}$), independently of the base learner used. Also in terms of the gene-centric F_{max} measure hierarchical ensemble methods

achieve significantly better results, but only with *TPR* having linear SVMs as base learner. The best precision is obtained by *Clus-HMC-Ens* and the best recall by *PHENOstruct*, but the best compromise between these measures is obtained by *TPR-SVM*, thus resulting in the best overall F_{max} score. Interestingly enough, the hierarchical ensemble methods are always able to improve the results of the flat methods used as base learner; in particular we have a very large improvement of the F_{max} when *RANKS* is used as base learner, while the improvement is smaller with the AUROC, for which *RANKS* just achieves relatively high values. With the *Onset* subontology we obtain similar results, while with the smallest subontology (*Inheritance*) including only 13 terms, *PHENOstruct* and *Clus-HMC-Ens* achieve the best performances. Overall, these results show that the proposed hierarchical ensemble methods are competitive with state-of-the-art methods such as *PHENOstruct* and *Clus-HMC-Ens* and moreover show that they can improve the results of different flat methods, such as the network-based semi-supervised *RANKS* algorithm and the supervised *SVM* classifier. Detailed experimental results available in the Supplementary Information confirm these findings.

Table 2. Prediction of genes associated to HPO terms of the main Organ subontology: 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
TPR-RANKS	**0.89**	0.40	0.34	0.48
TPR-SVM	0.77	**0.44**	0.38	0.51
HTD-RANKS	0.88	0.37	0.30	0.49
HTD-SVM	0.75	0.43	0.37	0.49
PHENOstruct	0.73	0.42	0.35	**0.56**
Clus-HMC-Ens	0.65	0.41	**0.39**	0.43
PhenoPPIOrth	0.52	0.20	0.27	0.15
Struct->Dis->HPO	0.49	0.23	0.16	0.41
RANKS	0.87	0.30	0.23	0.43
SVM	0.74	0.41	0.36	0.49

3.2 HPO Prediction of Newly Annotated Genes

In this section we assess the capacity of our proposed hierarchical ensemble methods to predict novel HPO annotations for human genes. To this end we used annotations of an old HPO release (January 2014) to predict the newly annotated genes of a recent HPO release (April 2016).

3.2.1 Data

Data Source. As data source we used the STRING 9.1 network, i.e one of the data sets used in the previous experiments (Section 3.1.1). Indeed STRING 9.1 has been constructed by integrating different sources of information, and the previous experiments as well as the experiments performed by Kahanda *et al.* (2015) revealed to be the most informative source of information for the prediction of HPO terms. We did not use the most recent release of the STRING database (v.10, Szklarczyk *et al.* (2015)), since we might introduce an indirect bias in the prediction, considering that STRING 10 was not available when the January 2014 HPO version has been released.

HPO DAG and Annotations. The experiments presented here are based both on the January 2014 HPO release ($10,320$ terms and $13,549$ between-term relationships) and on the April 2016 HPO release ($11,673$ terms and $15,459$ between-term relationships). Since in different releases some terms could have been removed, other changed or become obsolete, we mapped the old HPO terms to the new ones by parsing the annotation file of the January 2014 HPO release using as key the alt-ID taken from the obo file of the April 2016 HPO release. From the same HPO releases we downloaded all the corresponding gene-terms associations. Then we pruned HPO terms having less than 10 annotations obtaining a final HPO DAG composed by 2445 terms and 3059 between-terms relationships. Unlike the previous experimental part (Section 3.1), in the experiments presented here we considered the whole HPO, without splitting it up in its three main subontologies.

3.2.2 Experimental set-up and performance metrics.

We compared the generalization performance of *HTD* and *TPR* hierarchical ensemble methods versus PHENOstruct, the best performing state-of-the-art method in the previous set of experiments (Section 3.1).

We denote with T the set of genes having at least 1 annotation with an HPO term of the "old" January 2014 HPO release (2804 genes) and with S the set of newly annotated genes, i.e. genes having at least one new annotation in the "new" April 2016 HPO release, but previously unannotated in the January 2014 HPO release (608 genes). Hence we have that $S \cap T = \emptyset$. We used the set T as training set and the set S as test set, and we applied a classical hold-out procedure to assess the capability of predicting newly annotated genes using only the annotations of the previous HPO release. For the *HTD* and *TPR* methods we used the SVMs as base learners. To evaluate the performance of PHENOstruct, we downloaded and adapted the freely available C++ PHENOstruct code to perform the hold-out procedure described above.

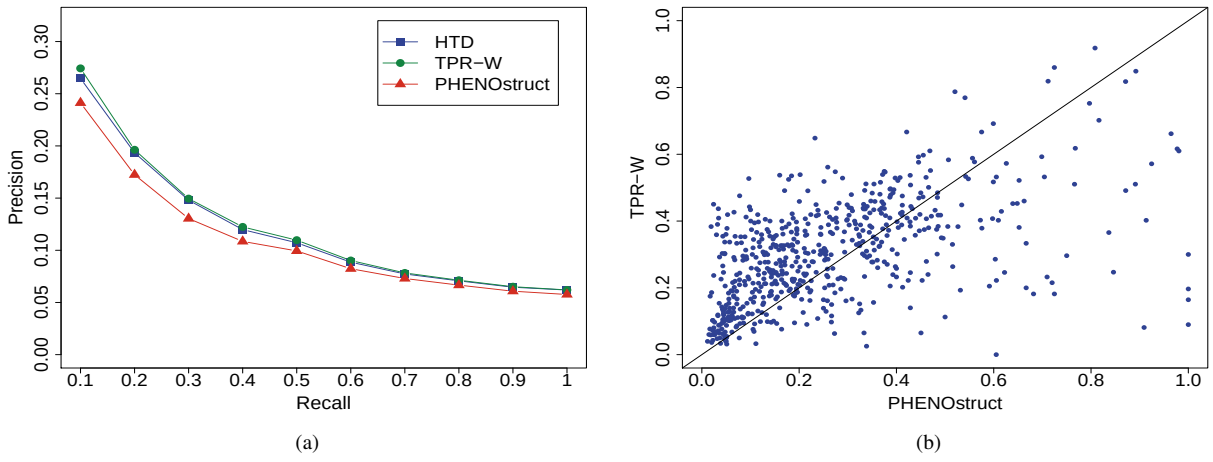
As performance metrics we used the same *gene-centric* and *term-centric* measures mentioned in subsection 3.1.2. In addition we added a further *term-centric* measure: the Area Under the Precision Recall Curve (AUPRC), to take into account the imbalance of annotated vs unannotated HPO terms (Saito and Rehmsmeier, 2015).

3.2.3 Results and Discussion

Table 3 shows that *TPR-W* and *HTD* are able to predict newly annotated genes, even if with a certain decay in the overall performance, as expected, with respect to the cross-validated results of Table 2. Hierarchical ensembles are competitive with respect to the state-of-the-art *PHENOstruct* algorithm. Indeed hierarchical ensemble methods attain significantly better results both in terms of average AUPRC and F_{max} (Wilcoxon paired rank sum test, p-value $< 10^{-9}$), while *PHENOstruct* achieves the best AUROC results. It is worth noting that the precision of *TPR-W* and *HTD* is higher than that of *PHENOstruct* at any recall level (Fig. 5a), and these results are confirmed also by the "per-gene" hierarchical F_{max} score: *TPR-W* "wins" with 431 and "loses" with 177 human genes (Fig. 5b).

Table 3. Prediction of newly annotated human genes. Average AUROC and AUPRC across terms and average F_{max}, Precision and Recall across genes. Results significantly better than the others according to the Wilcoxon Rank Sum test ($\alpha = 10^{-9}$) are highlighted in bold.

Meas. / Methods	AUROC	AUPRC	F_{max}	Precision	Recall
HTD	0.6464	0.1207	0.3794	**0.3581**	0.4033
TPR-W	0.6512	**0.1237**	**0.3826**	0.3512	0.4202
PHENOstruct	**0.6661**	0.1089	0.3635	0.3040	**0.4519**



(a) (b)

Fig. 5. (a) Compared precision at different recall levels averaged across 2444 HPO terms (b) Scatterplot of F_{max} values. Each point represent one of the 608 genes of the test set. PHENOstruct values are in abscissa, TPR-W values in ordinate.

By restricting the evaluation of the results only to the classes and genes best predicted by the best method (*TPR-W*), i.e. HPO terms having AUROC$>$ 0.7 and genes having $F_{max} > 0.3$, we obtain a relatively large set of "well predicted" HPO terms (779) and newly annotated genes (296, about the half of the overall newly annotated genes, Table 4). Fig. 6 shows the distribution of the best "per-term" AUROC and AUPRC results of *HTD* and different variants of *TPR* ; detailed results about the prediction of newly annotated genes are available in the Supplementary Information

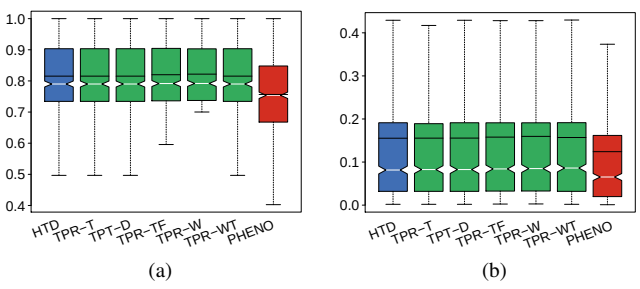
The overall computational time of hierarchical ensemble methods ...

While the results of Table 4 and Fig. 6 are biased in favour of *TPR-W*, Fig. 7 shows that hierarchical ensemble methods outperform *PHENOstruct* in precision at any recall level independently if the best predicted HPO terms are selected with respect to *TPR-W* or *PHENOstruct* best predictions.

This second set of experiments resembles those of the recent CAFA2 challenge, since we considered the predictions of newly annotated genes. For both the main metrics used (AUROC and F_{max}), hierarchical

Table 4. Prediction of newly annotated human genes considering only the best predictions. Average AUROC and AUPRC across terms and average F_{max}, Precision and Recall across genes considering only HPO terms with AUROC > 0.7 (779 terms) and $F_{max} > 0.3$ (296 genes). Results significantly better than the others according to the Wilcoxon Rank Sum test ($\alpha = 10^{-9}$) are highlighted in bold.

Meas. / Methods	AUROC	AUPRC	F_{max}	Precision	Recall
HTD	0.8155	0.1551	0.4716	0.4429	0.5042
TPR-W	**0.8219**	**0.1594**	**0.4793**	**0.4572**	0.5037
PHENOstruct	0.7565	0.1241	0.4297	0.3583	**0.5366**



(a) (b)

Fig. 6. Distribution of the AUROC and AUPRC values across the best predicted terms (779 HPO terms). (a) AUROC (b) AUPRC. HTD and different TPR variants are compared with PHENOstruct

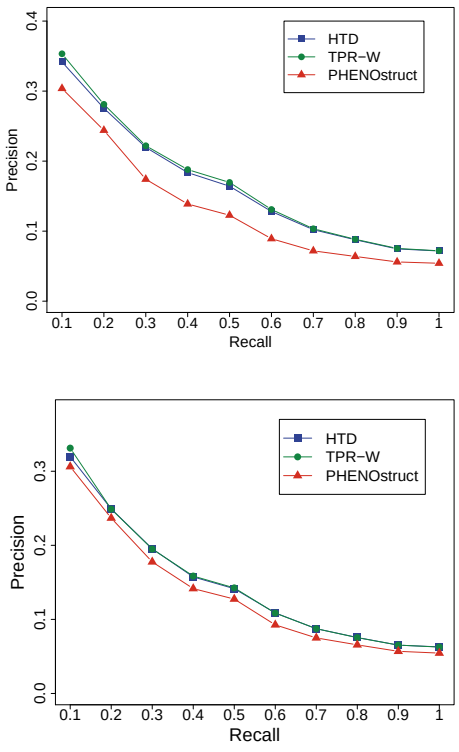


Fig. 7. Precision at different recall levels, considering only the best predicted terms. Top: Results considering only the HPO terms predicted with AUROC>0.7 by TPR-W (779 terms); Bottom: Results considering only the HPO terms predicted with AUROC>0.7 by PHENOstruct (852 terms).

ensemble methods achieved sligtly better results than those obtained by the best CAFA2 methods (Jiang *et al.*, 2016). Neverthless this comparison should be considered with caution, since our data, updated at the April 2016 release, are different from those used in the CAFA2 challenge (updated at September 2014).

4 Conclusion

NOT SO GOOD: IT SHOULD BE IMPROVED

The experimental results show that hierarchical ensemble methods are able to predict associations between genes and abnormal phenotypes with results competitive with state-of-the-art algorithms. The low computational complexity of the hierarchical correction step of both *HTD* and *TPR* (linear with respect the number of nodes of the HPO) enables its efficient application using different types of base learners. Indeed we showed that the proposed hierarchical algorithms are able to improve the predictions of flat methods, such as the *RANKS* semi-supervised network-based algorithm, that resulted one of the top ranked method in the recent CAFA2 challenge for HPO term prediction (Jiang *et al.*, 2016), as well as supervised methods such as SVM. By exploiting the modular structure of *HTD* and *TPR*, we guess that in principle any flat method, used as base learner within our proposed hierarchical methods, can in principle improves its performance for the prediction HPO terms. We also proved that both *HTD* and *TPR* always provide consistent predictions that obey the true path rule. Finally experimental results show that these methods can predict newly annotated genes, and as just outlined in the recent CAFA2 challenge, there is room to improve predictions, as soon as new annotations will be available for human genes, since at the moment only a minority of them is annotated with HPO terms. The prediction of human gene-abnormal phenotype association is a crucial challenge in discovering novel genes related to genetic diseases and cancer (Boycott *et al.*, 2013; Kohler *et al.*, 2014). Hierarchical ensemble methods can help the discovery of new gene-HPO term associations, thus getting insights into novel genes involved in diseases characterized by abnormal phenotypes for which no knowledge about disease genes is available.

Funding

References

Aymé, S. and Schmidtke, J. (2007). Networking for rare diseases: a necessity for europe. *Cin. Genet.*, **52**, 1477 – 1483.

Boycott, K., Vanstone, M., Bulman, D., and MacKenzie, A. (2013). Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics*, **14**, 681âĹ“691.

Bragin, E., Chatzimichali, E. A., Wright, C. F., Hurles, M. E., Firth, H. V., Bevan, A. P., and Swaminathan, G. J. (2014). DECIPHER: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. *Nucleic Acids Research*, **42**(Database-Issue), 993–1000.

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.

Chatr-Aryamontri, A., Breitkreutz, B.-J., Heinicke, S., Boucher, L., Winter, A. G., Stark, C., Nixon, J., Ramage, L., Kolas, N., O'Donnell, L., Reguly, T., Breitkreutz, A., Sellam, A., Chen, D., Chang, C., Rust, J. M., Livstone, M. S., Oughtred, R., Dolinski, K., and Tyers, M. (2013). The biogrid interaction database: 2013 update. *Nucleic Acids Research*, **41**(Database-Issue), 816–823.

Cormen, T., Leiserson, C., Rivest, R., and RL, S. (2009). *Introduction to Algorithms*. MIT Press, Boston.

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C., and Jensen, L. J. (2013). STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, **41**(Database-Issue), 808–815.

Guan, Y., Myers, C., Hess, D., Barutcuoglu, Z., Caudy, A., and Troyanskaya, O. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, **9**(S2).

Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., Valle, D., and McKusick, V. A. (2002). Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, **30**(1), 52–55.

Jiang, Y. *et al.* (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, **17**(184).

Kahanda, I., Funk, C., Verspoor, K., and Ben-Hur, A. (2015). Phenostruct: Prediction of human phenotype ontology terms using heterogeneous data sources. *F1000Research*, **4**, 259.

Kohler, S., Doelken, S., Mungall, C., Bauer, S., Firth, H., Bailleul-Forestier, I., Black, G., Brown, D., Brudno, M., Campbell, J., FitzPatrick, D., Eppig, J., Jackson, A., Freson, K., Girdea, M., Helbig, I., Hurst, J., Jahn, J., Jackson, L., Kelly, A., Ledbetter, D., Mansour, S., Martin, C., Moss, C., Mumford, A., Ouwehand, W., Park, S., Riggs, E., Scott, R., Sisodiya, S., Van Vooren, S., Wapner, R., Wilkie, A., Wright, C., Vulto-van Silfhout, A., de Leeuw, N., de Vries, B., Washingthon, N., Smith, C., Westerfield, M., Schofield, P., Ruef, B., Gkoutos, G., Haendel, M., Smedley, D., Lewis, S., and Robinson, P. (2014). The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic Acids Research*, **42**((Database issue)), D966–74.

Moreau, Y. and Tranchevent, L. (2012). Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Rev. Genet.*, **13**(8), 523–536.

Musso, G. *et al.* (2014). Novel cardiovascular gene functions revealed via systematic phenotype prediction in zebrafish. *Development*, **141**, 224–235.

Obozinski, G., Lanckriet, G., Grant, C., M., J., and Noble, W. (2008). Consistent probabilistic output for protein function prediction. *Genome Biology*, **9**(S6).

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.

Robinson, P. (2012). Deep phenotyping for precision medicine. *Human Mutations*, **32**(5), 777–780.

Robinson, P., Kohler, S., Bauer, S., Seelow, D., Horn, D., and Mundlos, S. (2008). The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease. *Am. J. Hum. Genet.*, **83**, 610–615.

Robinson, P., Krawitz, P., and Mundlos, S. (2011). Strategies for exome and genome sequence data analysis in disease-gene discovery projects. *Cin. Genet.*, **80**, 127 – 132.

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, **10**, e0118432.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., and Dzeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, **11**(2).

Silla, C. and Freitas, A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, **22**(1-2), 31–72.

Smedley, D., Schubach, M., Jacobsen, J. O., Köhler, S., Zemojtel, T., Spielmann, M., Jäger, M., Hochheiser, H., Washington, N. L., McMurry, J. A., Haendel, M. A., Mungall, C. J., Lewis, S. E., Groza, T., Valentini, G., and Robinson, P. N. (2016). A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease. *The American Journal of Human Genetics*, **99**(3), 595–606.

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K., Kuhn, M., Bork, P., Jensen, L. J., and von Mering, C. (2015). String v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, **43**(Database-Issue), 447–452.

Tsochantaridis, I., Joachims, T., Hoffman, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, **6**, 1453–1484.

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. (2014). Hierarchical Ensemble Methods for Protein Function Prediction. *ISRN Bioinformatics*, **2014**(Article ID 901419), 34 pages.

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).

Wang, P. *et al.* (2013). Inference of gene-phenotype associations via protein-protein interaction and orthology. *PLoS ONE*, **8**(10).

Zemojtel, T., Köhler, S., Mackenroth, L., Jäger, M., Hecht, J., Krawitz, P., Graul-Neumann, L., Doelken, S., Ehmke, N., Spielmann, M., Oien, N. C., Schweiger, M. R., Krüger, U., Frommer, G., Fischer, B., Kornak, U., Flöttmann, R., Ardeshirdavani, A., Moreau, Y., Lewis, S. E., Haendel, M., Smedley, D., Horn, D., Mundlos, S., and Robinson, P. N. (2014). Effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome. *Sci Transl Med*, **6**(252), 252ra123.