Hierarchical Ensemble Methods for Structured Prediction with Applications in Computational Biology

Marco Notaro

https://homes.di.unimi.it/notaro/
Bioinformatics vs Computational Biology

- **Computational Biology**: is the study of Biology using computational techniques. The main goal of a computational biologist is to make new insights about Biology and living system. Then *Computational Biology* is about *Science*.

- **Bioinformatics**: is about the creation of new algorithms able to solve problems. The main goal of a bioinformatician is to build tools that can work on biological, medical and pharmaceutical data. Then *Bioinformatics* is about *Computer Science*.

![Cartoon of a computational biologist and a bioinformatician]
• Bio-Ontologies (e.g. HPO, GO, DO): what are and why are useful in a bio-medical contest;

• State-of-the-art approaches ontology-based: Flat vs Hierarchy-aware learning methods;

• Proposed approaches: Hierarchical Ensemble Methods (HEM);

• Behavior of HEM in a state-of-the-art scenarios: HEM vs joint-kernel structured output method;

• Ongoing and Future Developments;
An ontology is an high-level representation of a domain of knowledge that describes concepts and semantic relationships between them in a form of Directed Acyclic Graph (DAG).
• **Human Phenotype Ontology** (HPO): provides a standardized categorization of the abnormalities associated to human diseases;

• **Disease Ontology** (DO): describes the classification of human diseases organized by etiology;

• **Gene Ontology** (GO): describes the function of genes and gene products;

• **Chemical Entities of Biological Interest** (ChEBI): structured dictionary of molecular entities focused on ‘small’ chemical compound;

• **MErged Disease voCabulary** (MEDIC): map the flat list of OMIM disease terms into the hierarchical nature of the MeSH vocabulary;

• **Anatomical Ontologies**: structured controlled vocabulary of the anatomy and development of the Zebrafish (ZFO), Xenopus (XAO), Mouse (MA);

More at OBO Foundry (Open Biological and Biomedical Ontologies):

http://www.obofoundry.org/

OBO-EDIT (http://oboedit.org/): open source ontology editor
Human Phenotype Ontology (HPO) (Köhler et al., 2017)


What is: standardized categorization of the phenotypic abnormalities associated to human diseases

- all relationships in the HPO are is-a relationships, i.e. simple class-subclass relationships

HPO Easter Release:
- Tot. Number of Nodes: 12,226
- Tot. Number of Edges: 16,044
Human Disease Ontology (DO) (Schriml et al., 2015)
Link: http://disease-ontology.org/

What is: controlled classification of human diseases

: “is-a” relationships are transitive, meaning that they are inherited up all paths to the root

DO March Release:
Tot. Number of Nodes: 8,240
Tot. Number of Edges: 8,437
**Gene Ontology (GO)** (Ashburner et al., 2000)

What is: three structured ontologies that describe gene products in terms of their association with BP, MF and CC in a species-independent manner.

**Biological Process** (BP) describes a collection of events carried out by one or more molecular functions (lipid metabolic process, Krebs acid cycle, antibiotic response).

BP April release:
Number of Nodes = 29,531
Number of Edges = 56,880
**Molecular function** (MF) describes activities that occur at molecular level, such as catalytic or binding activities.

**MF April release:**
Number of Nodes = 10,895
Number of Edges = 13,156
**Cellular component** (CC) ontology describes locations, at the levels of subcellular structures or macromolecular complexes, in which a specific gene product is located (e.g. nucleus, nuclear inner membrane, ribosome, synapse).

CC April release:
Number of Nodes = 4,096  
Number of Edges = 5,971
Ontologies provide predefined taxonomies for solving several relevant computational biology problems as:

- Protein Function prediction (GO);
- Prediction of human gene – abnormal phenotype associations (HP);
- Prediction of gene – disease associations (DO);

In silico methods unlike in vitro methods are not costly in time and in money and can support the molecular biologist in solving several bioomedical problem:

- understanding the role of a protein in a BP;
- annotating a new gene/protein at high level of accuracy;
- solving a functional genomics problem;
AFP: complex prediction problem characterized by several issues

• **Data Preparation**: construction, selection and normalization of the input data are complex and time-consuming. *Data preparation is relevant as algorithm design*;

• **Data-Fusion methods**: integration of multiple heterogeneous sources of data;

• **Unbalanced classification**: low number of positive and large number of negative examples;

• **Labels at different level of reliability**: each annotation is labeled with an *evidence code that* indicates how the annotation to a particular term is supported:
  - IPI/IGI: Inferred from Physical/Genetic Interaction (Experimental Evidence);
  - ISS: Inferred from Sequence Similarity (Computational Analysis Evidence)
  - TAS: Traceable Author Statement (annotation made on the basis of a statement made by the authors in the reference cited)
  - ... and much more. Full set of available evidence codes at [GO website](https://www.geneontology.org);

• **Multi-class and multi-label**: thousand functional classes and multiple annotations for each gene/protein;

• **Structured multi-label classification**: terms are structured in a hierarchy;
AFP: complex prediction problem characterized by several issues

- **Data Preparation**: construction, selection and normalization of the input data are complex and time-consuming. *Data preparation is relevant as algorithm design;*

- **Data-Fusion methods**: integration of multiple heterogeneous sources of data;

- **Unbalanced classification**: positive examples usually largely lower than negatives;

- **Labels at different level of reliability**: each annotation is labeled with an evidence code that indicates how the annotation to a particular term is supported:
  - IPI/IGI: Inferred from Physical/Genetic Interaction (Experimental Evidence);
  - ISS: Inferred from Sequence Similarity (Computational Analysis Evidence);
  - TAS: Traceable Author Statement (annotation made on the basis of a statement made by the authors in the reference cited);
  - ... and much more. Full set of available evidence codes at [GO website](#);

- **Multi-class and multi-label**: thousand of functional classes and multiple annotations for each gene/protein;

- **Structured multi-label classification**: hierarchical relationship between classes;

Can we design computational methods able to exploit the hierarchical relationships between ontology terms to provide biologically consistent predictions?
**Problem**: Hierarchical Prediction of Phenotypic Abnormalities associated to human diseases

**Problem**: Hierarchical Prediction of Protein Functions

Directed Acyclic Graph (DAG)

- molecular_function
- binding
- small molecule binding
- ion binding
- organic cyclic compound binding
- organic acid binding
- anion binding
- cation binding
- carboxylic acid binding
- amino acid binding
- tyrosine binding

**Different classification problem**
**Flat Classifier**: predict each class separately

Advantage: simplicity → makes prediction just for one class/term

- a priori loss of information

Drawbacks:
- neglect structural information between classes

Hierarchical constraint:
positive instance “P” for a class **implies**
positive instance for all ancestors of that class

A Toy Example: Flat Classification

Classifies protein “P”

- amino acid binding
- tyrosine binding

Violated
Hierarchy-aware approaches proposed in literature:

• Kernel-based structured output methods (Sokolov and Benhur 2010, Kahanda et al. 2015);

• Hierarchical Ensemble Methods (Silla et al. 2011, Valentini 2014);
Step 1: Flat Learning of the Ontology Terms

- **class 1**: D1 → LA → C1
- **class 2**: D2 → LA → C2
- **class 3**: D3 → LA → C3
- **class 4**: D4 → LA → C4
- **class n**: Dn → LA → Cn

Data → Flat Base Classifiers → Predictions

Step 2: Flat Predictions are Hierarchical Combined

Diagram showing the hierarchical combination of flat predictions.
State-of-the-art Hierarchical ensemble methods

- Most ensembles are conceived for tree-structured taxonomies \cite{Valentini2011, Cesa-Bianchi2012, Paes2012, Hernandez2013};

- Only a few for DAG-structured taxonomies \cite{Obozinski2008, Schietgat2010};

- With DAG-structured taxonomies it is difficult to achieve results comparable with flat methods \cite{Obozinski2008};

- DAGs are more complex than trees:
  - more parents;
  - more edges;
  - multiple paths;
  - nodes may belong to multiple levels;
**HTD-DAG**: Hierarchical Top Down for DAGs

**TPR-DAG**: True Path Rule for DAGs

1. Bottom-Up Step
2. Top-Down Step
HTD-DAG is a two-step learning strategy:

1) **Flat learning phase**: a base learner learns a specific class on a per-term basis (a set of independent classification problem);

2) **Top-Down step**: traversing the DAG by a per-level top-down visit to propagate the *negative predictions* towards the bottom of the hierarchy

- Remove constraint violations;
- Improvement of precision of the predictions;
To preserve the consistency of the predictions the levels must be defined according to the maximum distance from the root:

\[ y \text{ is consistent } \iff \forall i \in V, j \in \text{par}(i) \Rightarrow y_j \geq y_i \]
Input:
- \( G = \langle V, E \rangle \)
- \( \hat{y} = \langle \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} \rangle \) (flat predictions)

begin algorithm
01: A. \( dist := \text{ComputeMaxDist}(G, \text{root}(G)) \)
02: B. Per-level top-down visit of \( G \):
03: \( \bar{y}_{\text{root}(G)} := \hat{y}_{\text{root}(G)} \)
04: for each \( d \) from 1 to \( \xi \) do
05: \( N_d := \{i | dist(i) = d\} \)
06: for each \( i \in N_d \) do
07: \( x := \min_{j \in \text{par}(i)} \bar{y}_j \)
08: if \( x < \hat{y}_i \)
09: \( \bar{y}_i := x \)
10: else
11: \( \bar{y}_i := \hat{y}_i \)
12: end for
13: end for
end algorithm

Output:
- \( \bar{y} = \langle \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} \rangle \)

Max.Dist.: Bellman-Ford or Topological sort algorithm

HTD: the nodes are processed by level in an increasing order and the HTD ensemble predictions are returned

HTD-DAG Computational Complexity:
\( O(|V| + |E|) \)
HTD-DAG:
Flat scores $\hat{y}_i$ are hierarchically corrected to $\bar{y}_i$ according to this simple rule:

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

Limit case: predictions at leaves nodes are negatives.
TPR ensemble for DAGs: double flow of information

Sensitivity improving

Removing violations
In the Bottom-Up Step the ensemble decision is modified in according to this simple rule:

\[
\tilde{y}_i := \frac{1}{1 + |\phi_i|} \left( \hat{y}_i + \sum_{j \in \phi_i} \tilde{y}_j \right)
\]

1) Different strategy can be used to define the positive \( \phi_i \) children of class \( i \):

A) **Adaptive Threshold Strategy**: maximize \( M \) on training data by internal cv

\[
\phi_i := \{ j \in \text{child}(i) | \tilde{y}_j > t^*_j, t^*_j = \arg \max_t M(j, t) \}
\]

B) **Threshold Free Strategy**: positive nodes those that achieve a score higher than that of their parents

\[
\phi_i := \{ j \in \text{child}(i) | \tilde{y}_j > \hat{y}_i \}
\]
TPR-DAG is a family of algorithms

2) **Weighted TPR**: \( w \in [0,1] \) to balance the contribution between node \( i \) and that of its ‘positive’ children

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

3) **Descendants TPR**: to enhance the contribution of the most specific nodes we can consider the descendants instead of children

\[
\bar{y}_i := \frac{1}{1 + |\Delta_i|} (\hat{y}_i + \sum_{j \in \Delta_i} \bar{y}_j) \quad \Delta_i = \{ j \in desc(i)|\bar{y}_j > t_j \}
\]

4) **Descendants-TAU**: \( \tau \in [0,1] \) to balance the contribution between \( \phi_i \) and \( \delta_i \)

\[
\bar{y}_i := \frac{\tau}{1 + |\phi_i|} (\hat{y}_i + \sum_{j \in \phi_i} \bar{y}_j) + \frac{1 - \tau}{1 + |\delta_i|} (\hat{y}_i + \sum_{j \in \delta_i} \bar{y}_j) \quad \delta_i = \Delta_i \setminus \phi_i
\]
Input:
- $G = <V,E>$
- $V = \{1,2,\ldots,|V|\}$
- $\hat{y} = <\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_{|V|}>$, $\hat{y}_i \in [0,1]$

begin algorithm
01: A. Compute $\forall i \in V$ the max distance from root($G$):
02: $E' := \{e'|e \in E, e' = -e\}$
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 $i \in N_d$ do
09: Select the set $\phi_i$ of “positive” children
10: $\bar{y}_i := \frac{1}{1+|\phi_i|}(\hat{y}_i + \sum_{j \in \phi_i} \bar{y}_j)$
11: end for
12: end for

13: C. Per-level top-down visit of $G$:
14: $\hat{y} := \bar{y}$
15: for each $d$ from 1 to max($dist$) do
16: $N_d := \{i|dist(i) = d\}$
17: for each $i \in N_d$ do
18: $x := \min_{j \in \text{par}(i)} \bar{y}_j$
19: if ($x < \hat{y}_i$)
20: $\bar{y}_i := x$
21: else
22: $\bar{y}_i := \hat{y}_i$
23: end for
24: end for
end algorithm

Output:
- $\hat{y} = <\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_{|V|}>$

Propagation of the positive predictions towards the top of the hierarchy in order to enhance the sensitivity of the predictions

TPR-DAG scales linearly with the number of classes (since the graph is sparse)
**Operating mode of the TPR: toy example**

**FLAT SCORES**

- **Nodes at Level 0**
  - A: 0.9
  - B: 0.4
  - C: 0.3
  - D: 0.3
  - E: 0.6
  - F: 0.4
  - G: 0.5
  - H: 0.9
  - I: 0.3
  - J: 0.8
  - K: 0.9

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**BOTTOM UP STEP**

- **Nodes at Level 0**
  - A: 0.9
  - B: 0.5
  - C: 0.7
  - D: 0.7
  - E: 0.9
  - F: 0.7

- **Nodes at Level 1**
  - I: 0.3
  - J: 0.8
  - H: 0.9

- **Nodes at Level 2**
  - D: 0.7
  - C: 0.8
  - E: 0.8

- **Nodes at Level 3**
  - F: 0.4
  - J: 0.8
  - H: 0.9

- **Nodes at Level 4**
  - B: 0.5
  - G: 0.5

**TOP DOWN STEP**

- **Nodes at Level 0**
  - A: 0.9
  - G: 0.7
  - B: 0.5

- **Nodes at Level 1**
  - I: 0.3
  - J: 0.5
  - H: 0.5

- **Nodes at Level 2**
  - F: 0.7
  - D: 0.7
  - C: 0.7

- **Nodes at Level 3**
  - B: 0.5
  - G: 0.7

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**Nodes at Level 0**

- **A**: 0.9
  - **B**: 0.5
  - **C**: 0.7
  - **D**: 0.7
  - **E**: 0.9
  - **F**: 0.7

**Nodes at Level 1**

- **I**: 0.3
  - **J**: 0.8
  - **H**: 0.9

**Nodes at Level 2**

- **D**: 0.7
  - **C**: 0.8
  - **E**: 0.8

**Nodes at Level 3**

- **F**: 0.4
  - **J**: 0.8
  - **H**: 0.9

**Nodes at Level 4**

- **B**: 0.5
  - **G**: 0.5

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4

**Legend**
- Node at Level 0
- Nodes at Level 1
- Nodes at Level 2
- Nodes at Level 3
- Nodes at Level 4
HTD-DAG provides always biologically consistent predictions:

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

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

TPR-DAG provides always biologically consistent predictions:

**Theorem 2.** Given a DAG $G = < V, E >$, a level function $\psi$ that assigns to each node its maximum path length from the root, a set of predictions $\tilde{y} = < \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{|V|} >$ generated by the bottom-up step of the TPR algorithm for each class associated with its corresponding node $i \in \{1, \ldots, |V|\}$, the top-down step of the TPR algorithm assures that for the set of ensemble predictions $\tilde{y} = < \tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{|V|} >$ the following property holds:

$$\forall i \in V, j \in \text{par}(i) \Rightarrow \tilde{y}_j \geq \tilde{y}_i$$
Consistent hierarchical predictions and inconsistent flat predictions for the protein coding gene C1QC (complement C1q C chain) whose deficiency is associated with lupus erythematosus and glomerulonephritis (Lopez-Lera et al., 2014)
Hierarchical ensemble methods can improve flat predictions by reducing the number of FN and FP. 

Hierarchical ensembles recover 4 TP for the protein coding gene RGS9 (regulator of G-protein signalling 9) whose mutations cause bradyopsia (Michaelides et al. 2010)
Hierarchical ensembles recover 6 TN for the protein coding gene ENAM (enamelin) that encodes the largest protein in the enamel matrix whose deficiency is associated with amelogenesis imperfecta (Rajpar et al. 2001).
How is the behaviour of our ensemble methods in a state-of-the-art scenario?

1. Comparison with PHENOstruct: a state-of-the-art joint-kernel structured output approach (Kahanda et. al 2015)

2. Assess the capacity to predict novel HPO annotations for human genes: we used the annotation of an old HPO release (January 2014) to predict the newly annotated genes of a recent HPO release (April 2016). In other words we applied a classical hold-out procedure between two different HPO releases.
Scatter plot of hierarchical $F_{\text{max}}$ values: TPR-W “wins” with 431 gene and “loses” with 177 genes (Tot. genes test set: 608)

Precision-Recall curves across 2444 HPO terms: HEMs significantly improve PHENOstruct in according to Wilcoxon Sum Rank test
Precision-Recall curves of the newly annotated genes considering only the best predicted terms: AUROC > 0.7

Results considering only the HPO terms predicted with AUROC>0.7 by TPR-W (778 terms)

Results considering only the HPO terms predicted with AUROC>0.7 by PHENOMstruct (852 terms)
Distribution of AUROC and AUPRC values across HPO terms between our ensemble methods and PHENOstruct

AUROC distribution

AUPRC distribution
Since for about half of Mendelian disease no causative genes are know (Chong et.al 2015), our ensemble methods may contribute to the discovery such genes and to unveil the full spectrum of phenotypes associated with them.
• Genome-wide and ontology-wide experimental results show that our hierarchical ensemble method are competitive with PHENOStruct;
• Computational time significantly lower than state-of-the-art joint-kernel structured output methods (training + test of hold-out):
  • HTD: **12 minutes**;
  • TPR-W: **3 hours** (tuning of w parameter by 5-fold-cross-validation);
  • PHENOstruct: **18 hours**;

Using an Intel Xeon CPU E5-2630 2.6Hz with 128 GB of RAM

**HTD-DAG and TPR-DAG:**
- a) scale linearly with number of the class;
- b) can be applied to big data;
- c) provide biologically consistent predictions;
- d) improve upon flat predictions;
1. In principle the HEM can significantly improve any flat prediction independently of which flat approach we used: I plan to do more experiments with a large set of base learners (GO ontology and in multi-species environment)

2. Top-Down step: instead of HTD strategy → Isotonic Regression

Input:
- \( G = < V, E > \)
- \( V = \{1, 2, \ldots, |V|\} \)
- \( \hat{y} = < \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|} >, \quad \hat{y}_i \in [0, 1] \)

begin algorithm
01: A. Isotonic correction:
02: \( \bar{y} = \left\{ \begin{array}{l}
\min_{\bar{y}} \sum_{i \in V} (\hat{y}_i - \bar{y}_i)^2 \\
\forall i, \quad j \in \text{par}(i) \Rightarrow \bar{y}_j \geq \bar{y}_i
\end{array} \right. \)

end algorithm

Output:
- \( \bar{y} = < \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{|V|} > \)

• select the closest solution to flat predictions in the sense of the squared error
• constraints are maintained by construction
3. TPR-DAG is a family of algorithm: I plan to design novel TPR-D variants
   i. *linear decay* of weights with respect to the levels;
   ii. *linear increment* of the weights (from bottom to top) in order to
       put more weight on predictions made on the most specific terms;

4. Prediction of coding (mRNA) and non coding (IncRNA, miRNA) RNA
   interactions inferred from an integrated multi-ontology environment
   (GO, HPO, DO)
   i. Hierarchical prediction of GO, DO, HPO terms: prediction of PCGs
      and NCGs performed across multi-ontology;
   ii. Discover novel interaction between ncRNA and PCGs using known
       and predicted annotation → will help us to better understand
       complex genetic diseases and cancer

(Joint Collaboration with Dept. of Computer Engineering of the University of
Granada)
Acknowledgments

Giorgio Valentini

https://anacletolab.di.unimi.it

Peter N. Robinson

https://www.jax.org/

Max Schubach

https://www.charite.de/

II. M. Notaro, M. Frasca, M. Mesiti, M. Schubach, P.N. Robinson and G. Valentini, *Ensembling Descendant Term Classifiers to Improve Gene – Abnormal Phenotype Predictions*, submitted to *Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB 2017)*


THANKS for YOUR ATTENTION!

ANY QUESTIONS?