Structural SVM and Applications in Bioinformatics

Jovana Kovačević

University of Belgrade, Faculty of Mathematics
Machine Learning and Applications Group

November 2017

Structure

1. Structural classification
   - Support vector machine (SVM)
   - Structural support vector machine (SSVM)
2. Protein function prediction problem
   - Protein function representation
   - Structural classification problem
3. Implementation
   - Parameter adjusting
   - Loss function
   - Optimization algorithm
4. Performance

Structure

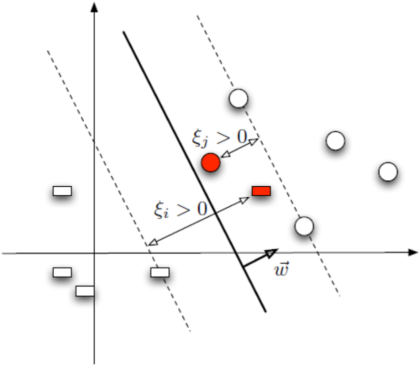
1. Structural classification
    - Support vector machine (SVM)
    - Structural support vector machine (SSVM)
2. Protein function prediction problem
    - Protein function representation
    - Structural classification problem
3. Implementation
    - Parameter adjusting
    - Loss function
    - Optimization algorithm
4. Performance

Support vector machine (SVM)

- Output: $y \in \{-1, 1\}$
- Discriminant function: $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b$
- Inference problem: $y = sgn(f(\boldsymbol{x}))$
- Learning problem: minimize

$$\frac{||\boldsymbol{w}||^2}{2} + C \sum_{i=1}^{n} \xi_i$$

such that $y_i \cdot f(\boldsymbol{x_i}) \geq 1 - \xi_i$, $\xi_i \geq 0$ for each training example $(\boldsymbol{x_i}, y_i)$



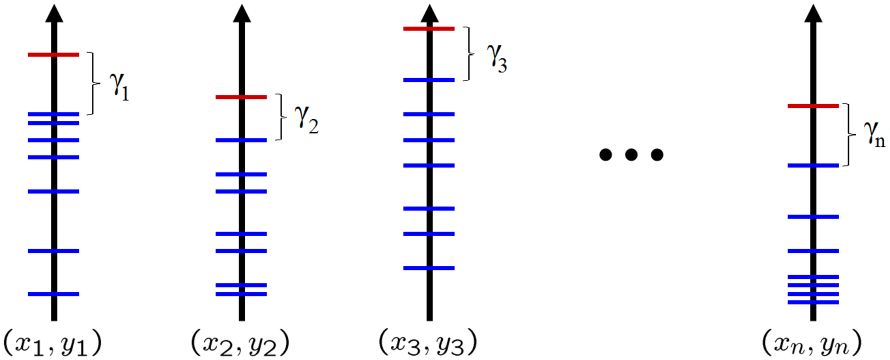
Maximize margin while minimizing training error

Structural support vector machine (SSVM)

- Output: $\boldsymbol{y}$ is a structured object, eg. array, graph, tree, ...
  Set $\boldsymbol{Y}$ of all outputs can be huge or even infinite
- Joint representation: $\Psi : \boldsymbol{X} \times \boldsymbol{Y} \to \mathbb{R}^n$
- Score function: $F_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \langle \Psi(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{w} \rangle$
- Inference problem: $\boldsymbol{y}^* = \mathrm{argmax}_{\boldsymbol{y} \in \boldsymbol{Y}} F_{\boldsymbol{w}}(\boldsymbol{x}^*, \boldsymbol{y})$
- Loss function: $\Delta(\boldsymbol{y}, \hat{\boldsymbol{y}})$ (eg. Hamming distance, Jaccard distance, ...)
- Margin:

$$\gamma_i = F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\}$$

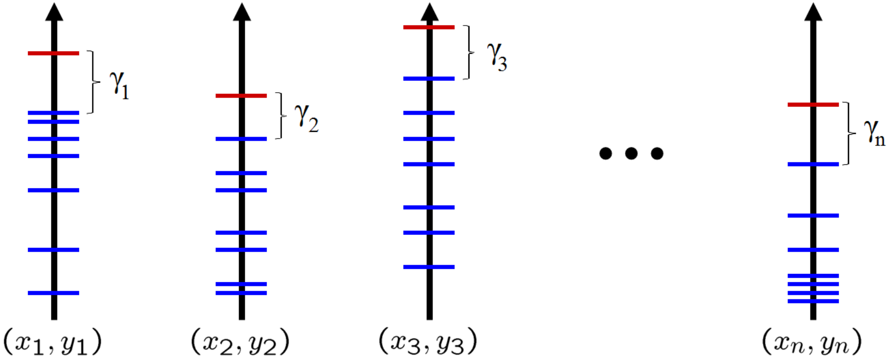
Margin



$$\gamma_i = F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y_i}) - \max_{\mathbf{y} \in Y \setminus \mathbf{y_i}} \{F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y})\}$$

- hard formulation: $\gamma_i \geq 1$

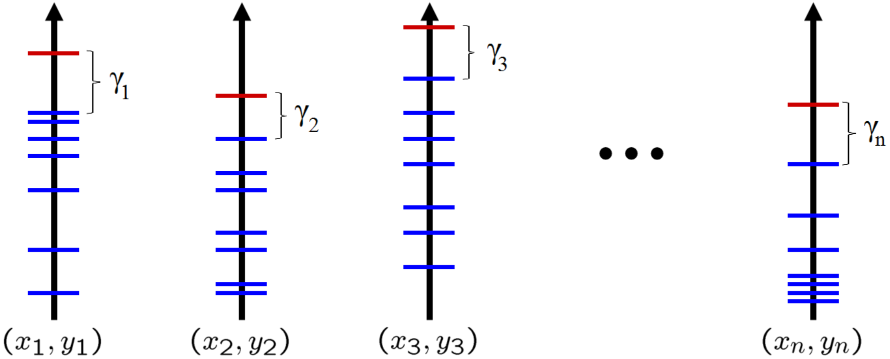
Margin



$$\gamma_i = F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y_i}) - \max_{\mathbf{y} \in Y \setminus \mathbf{y_i}} \{F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y})\}$$

- hard formulation: $\gamma_i \geq 1$
- soft formulation: $\gamma_i \geq 1 - \xi_i$

Margin



$$\gamma_i = F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y_i}) - \max_{\mathbf{y} \in Y \setminus \mathbf{y_i}} \{F_{\mathbf{w}}(\mathbf{x_i}, \mathbf{y})\}$$

- hard formulation: $\gamma_i \geq 1$
- soft formulation: $\gamma_i \geq 1 - \xi_i$
- margin rescaling formulation: $\gamma_i \geq \Delta(\mathbf{y_i}, \mathbf{y_{wrong}}) - \xi_i$

Linear constraints

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Linear constraints

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$
- ... and for any possible wrong output $\boldsymbol{y}$

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$
- ... and for any possible wrong output $\boldsymbol{y}$
- ... have the score for the correct output

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$
- ... and for any possible wrong output $\boldsymbol{y}$
- ... have the score for the correct output
- ... greater than the score for the incorrect output

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$
- ... and for any possible wrong output $\boldsymbol{y}$
- ... have the score for the correct output
- ... greater than the score for the incorrect output
- ... by at least the loss between the correct and incorrect output

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Linear constraints

- For all training examples $(\boldsymbol{x_i}, \boldsymbol{y_i})$
- ... and for any possible wrong output $\boldsymbol{y}$
- ... have the score for the correct output
- ... greater than the score for the incorrect output
- ... by at least the loss between the correct and incorrect output
- ... eventually subtracted by slack variable

$$\forall i : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - \max_{\boldsymbol{y} \in Y \setminus \boldsymbol{y_i}} \{F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y})\} \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y_{wrong}}) - \xi_i$$

Cancel maximum function:

$$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$$

Quadratic program formulation

$$\min_{\boldsymbol{w},\xi} \frac{||\boldsymbol{w}||^2}{2} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

s.t. $\forall i : \xi_i \geq 0$

$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$

Quadratic program formulation

$$\min_{\boldsymbol{w},\xi} \frac{||\boldsymbol{w}||^2}{2} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

s.t. $\forall i : \xi_i \geq 0$
$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$

- possibly very large number of constraints, even infinite

Quadratic program formulation

$$\min_{\boldsymbol{w},\xi} \frac{||\boldsymbol{w}||^2}{2} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

s.t. $\forall i : \xi_i \geq 0$

$\forall i, \forall \boldsymbol{y} \in \boldsymbol{Y} : F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y_i}) - F_{\boldsymbol{w}}(\boldsymbol{x_i}, \boldsymbol{y}) \geq \Delta(\boldsymbol{y_i}, \boldsymbol{y}) - \xi_i$

- possibly very large number of constraints, even infinite
- SVMstruct[1] framework (using cutting plane method for optimization)

[1] T. Joachims et al., Machine Learning 2009

Structure

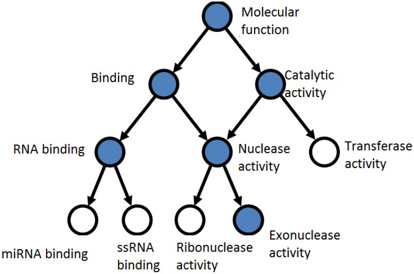
1. Structural classification
   - Support vector machine (SVM)
   - Structural support vector machine (SSVM)
2. Protein function prediction problem
   - Protein function representation
   - Structural classification problem
3. Implementation
   - Parameter adjusting
   - Loss function
   - Optimization algorithm
4. Performance

Why is it important?

- Knowing protein function informs us on its role in the organism
- Functional mutations may be the cause of different human diseases
- Growing number of newly discovered proteins
- Slow and expensive experimental methods vs. fast and cheaper computational methods

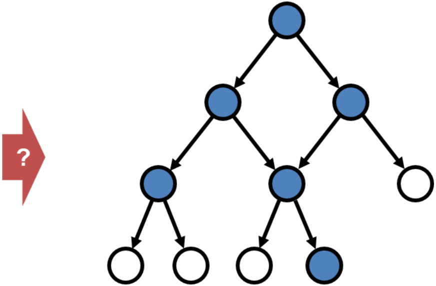
Protein function representation

- GO project
- Three different ontologies:
    - molecular function
    - biological processes
    - cellular component
- Each node describes more specific function than its ancestors.
- Consistency requirement.



Structural classification problem

```
>sp|P04637|P53_HUMAN
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIE
QWFTEDPGPDEAPRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKT
YQGSYGFRLGFLHSGTAKSVTCTYSPALNKMFCQLAKTCPVQLWVDSTPPP
GTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLAPPQHLIRVEGNLRVE
YLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPILTII
TLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKR
ALPNNTSSSPQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGK
EPGGSRAHSSHLKSKKGQSTSRHKKLMFKTEGPDSD
```



Structure

1. Structural classification
   - Support vector machine (SVM)
   - Structural support vector machine (SSVM)
2. Protein function prediction problem
   - Protein function representation
   - Structural classification problem
3. Implementation
   - Parameter adjusting
   - Loss function
   - Optimization algorithm
4. Performance

Adjusting SVMstruct to solving protein function prediction problem

- input vector $\boldsymbol{x}$ - primary sequence information coded as histogram of tetragrams

Adjusting SVMstruct to solving protein function prediction problem

- input vector $\boldsymbol{x}$ - primary sequence information coded as histogram of tetragrams
- output vector $\boldsymbol{y}$ - 0-1 vector, each element corresponds to one node in the GO ontology

Adjusting SVMstruct to solving protein function prediction problem

- input vector $\boldsymbol{x}$ - primary sequence information coded as histogram of tetragrams
- output vector $\boldsymbol{y}$ - 0-1 vector, each element corresponds to one node in the GO ontology
- joint representation of input and output vector $\Psi(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} \otimes \boldsymbol{y}$

Adjusting SVMstruct to solving protein function prediction problem

- input vector $\boldsymbol{x}$ - primary sequence information coded as histogram of tetragrams
- output vector $\boldsymbol{y}$ - 0-1 vector, each element corresponds to one node in the GO ontology
- joint representation of input and output vector $\Psi(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} \otimes \boldsymbol{y}$
- loss function $\Delta(\boldsymbol{y}, \boldsymbol{y'})$: Jaccard's distance, $1 - F_1$, semantic distance[2]

[2] Clark and Radivojac, Bioinformatics 2013

Adjusting SVMstruct to solving protein function prediction problem

- input vector $\boldsymbol{x}$ - primary sequence information coded as histogram of tetragrams
- output vector $\boldsymbol{y}$ - 0-1 vector, each element corresponds to one node in the GO ontology
- joint representation of input and output vector $\Psi(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} \otimes \boldsymbol{y}$
- loss function $\Delta(\boldsymbol{y}, \boldsymbol{y'})$: Jaccard's distance, $1 - F_1$, semantic distance[2]
- solution for inference and augmented inference problem: proposed algorithm for solving the following optimization problems that appear in the training and testing phase:

$$\arg\max_{\boldsymbol{y'} \in Y} (F(\boldsymbol{x}, \boldsymbol{y'}) + \Delta(\boldsymbol{y}, \boldsymbol{y'}))$$

$$\arg\max_{\boldsymbol{y'} \in Y} (F(\boldsymbol{x}, \boldsymbol{y'}))$$

[2] Clark and Radivojac, Bioinformatics 2013

Information content of a graph

- $i(T)$ - information content of graph T

$$\begin{aligned} i(T) &= \log \frac{1}{Pr(T)} \\ &= \log \frac{1}{\prod_{v \in T} Pr(v|\mathcal{P}(v))} \\ &= \sum_{v \in T} \log \frac{1}{Pr(v|\mathcal{P}(v))} \\ &= \sum_{v \in T} ia(v) \end{aligned}$$

- $ia(v)$ - information assertion of node v

Semantic distance

- T - true DAG
- P - predicted DAG
- remaining uncertainty

$$ru(T,P) = \sum_{v \in T \setminus P} ia(v)$$

- misinformation

$$mi(T,P) = \sum_{v \in P \setminus T} ia(v)$$

- semantic distance

$$s_k(T,P) = (ru(T,P)^k + mi(T,P)^k)^{\frac{1}{k}}$$

- normalized semantic distance

$$normalized\_s_k(T,P) = \frac{s_k(T,P)}{\sum_{v \in T \cup P} ia(v)}$$

```
Input: training instance $(\boldsymbol{x_i}, \boldsymbol{y_i})$
Output: $\boldsymbol{y_{best}}$ that maximizes $H(\boldsymbol{x_i}, \boldsymbol{y})$ over $\boldsymbol{y} \in Y$
Initialization: $L = \{\boldsymbol{y_{root}}\}, \boldsymbol{y_{best}} = \emptyset, H_{best} = -\infty$
repeat
    $\boldsymbol{y_{head}}$ := first element from $L$
    $Y_{ext}$ := all extensions of $\boldsymbol{y_{head}}$ by one node
    for each $\boldsymbol{y_{ext}} \in Y_{ext}$ do
        if $i(\boldsymbol{y_{ext}}) \geq imax$ then
            continue
        end if
        insert $\boldsymbol{y_{ext}}$ in sorted linked list $L$
        if $H(\boldsymbol{y_{ext}}) > H_{best}$ then
            update $\boldsymbol{y_{best}}, H_{best}$
        end if
    end for
    remove $\boldsymbol{y_{head}}$ from $L$, increment $step$
until $step > smax$ or $L$ is empty
```

Structure

1. Structural classification
   - Support vector machine (SVM)
   - Structural support vector machine (SSVM)
2. Protein function prediction problem
   - Protein function representation
   - Structural classification problem
3. Implementation
   - Parameter adjusting
   - Loss function
   - Optimization algorithm
4. Performance

Performance

- comparison to current methods on CAFA (around 130)
- very simple input, basic method setting, promising results

organism	F_1	best F_1	CAFA rang
ARATH	0.69	0.74	4
ECOLI	0.36	0.6	75
HUMAN	0.47	0.62	45
MOUSE	0.54	0.62	16
RAT	0.63	0.78	17

Thank you for your attention!