Leveraging Sequence Classification by Taxonomy-based Multitask Learning

Christian Widmer,[1] Jose Leiva,[1,2] Yasemin Altun,[2] Gunnar Rätsch[1]

1 Friedrich Miescher Laboratory of the Max Planck Society, Tübingen, Germany
2 Max Planck Institute for Biological Cybernetics, Tübingen, Germany

RECOMB 2010

August 15th, 2010

Talk Outline

- Introduction of the setting
- Methods: 3 approaches to Hierarchical Mulitask Learning
- Experiments (Toy, Splicing)
- Outlook
- Summary

Why Multitask learning?



- Model quality limited by insufficient training data

  $\Rightarrow$ Idea: Use information from related domains/tasks

- Related Distributions: Different but similar

- In Computational Biology domains correspond to organisms
  - Organisms share evolutionary history
  - Many biological processes are conserved

Why Hierarchies?

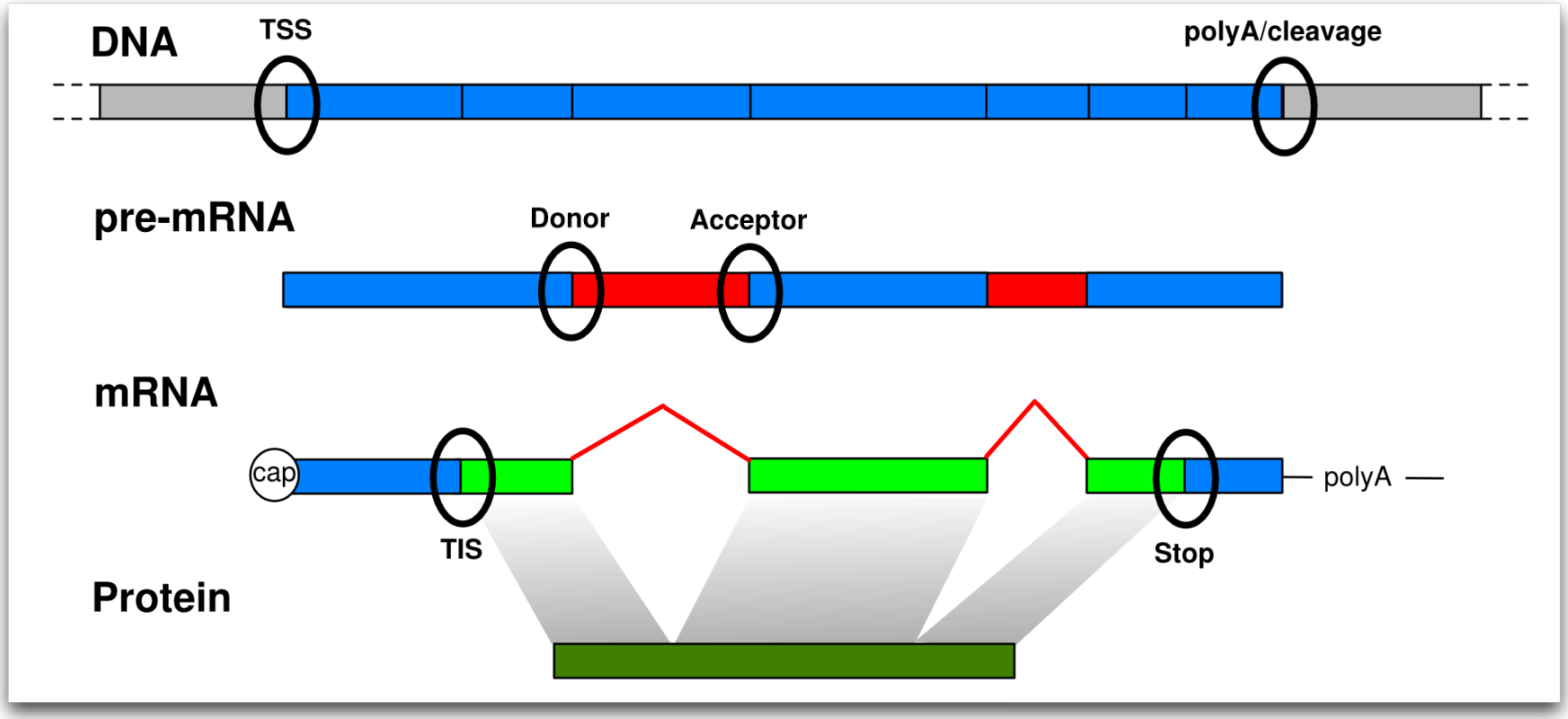
Hierarchical structure arises naturally from the Tree of Life



- Taxonomy used to define relationship between tasks
- Closer tasks benefit more from each other

What problems could benefit from this?

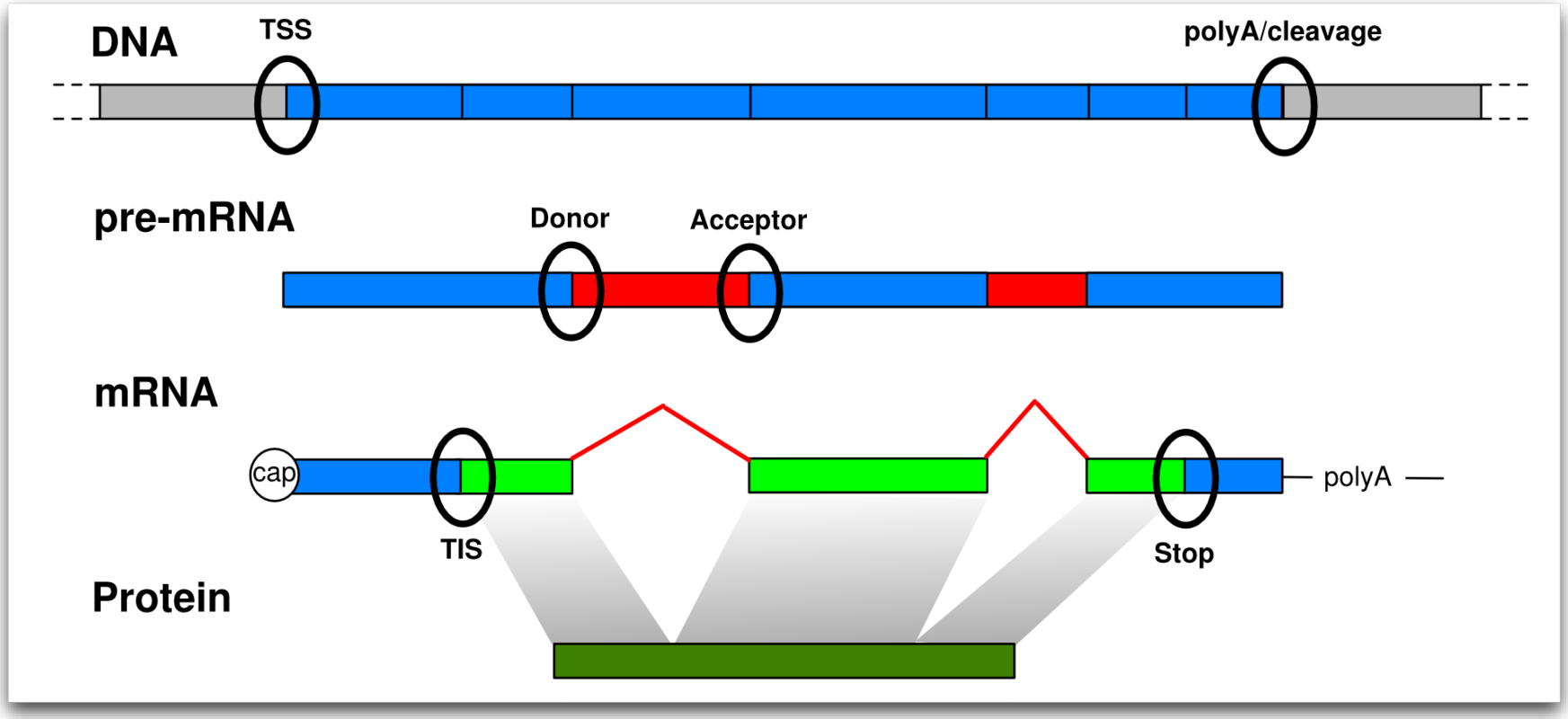
- Prediction of Sequence Signals in Gene Finding (mGene [4])



- Numerous potential applications for Multitask Learning
- Focus on splice-sites (Donor)

What problems could benefit from this?

- Prediction of Sequence Signals in Gene Finding (mGene [4])



- Numerous potential applications for Multitask Learning
- Focus on splice-sites (Donor)

Formal Problem Definition

Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition

Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition

Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition



Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition

Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Two ways of leveraging a given taxonomy $\mathcal{T}$

Taxonomy



Transformation

Task Similarity Matrix

$$\Gamma = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,M} \\ \vdots & \ddots & \vdots \\ \gamma_{M,1} & \cdots & \gamma_{M,M} \end{pmatrix}$$

Top-Down

Pairwise &

Multitask Kernel

Domain Adaptation by Regularization



Idea: Enforce model similarity via Regularization Term

- Regular SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right)$$

- DA-SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_{par}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$.

Domain Adaptation by Regularization



Idea: Enforce model similarity via Regularization Term

- Regular SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right)$$

- DA-SVM

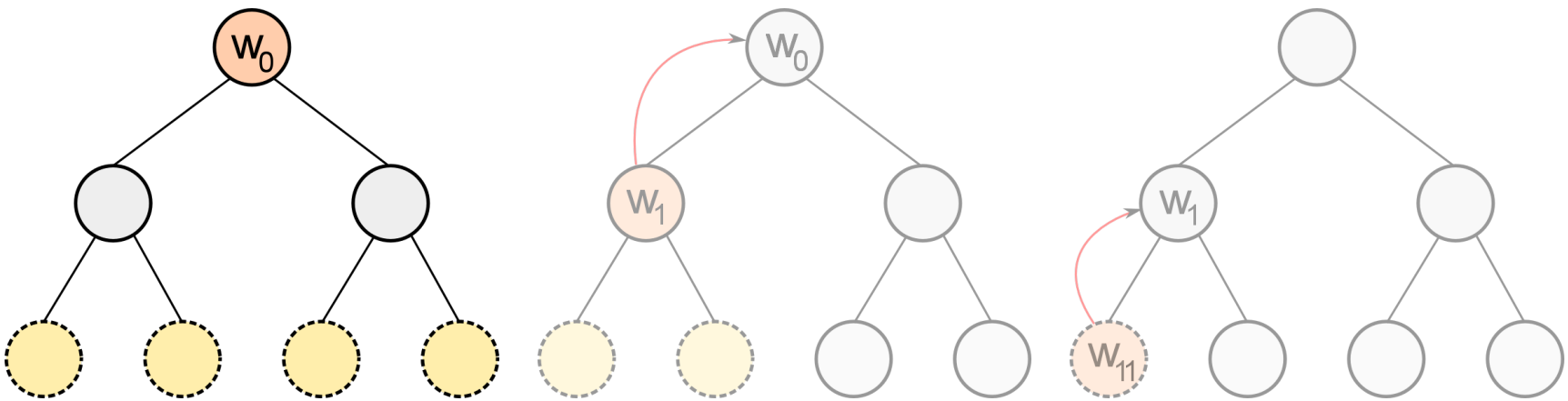
$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w} - \mathbf{w}_{par}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$.

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification

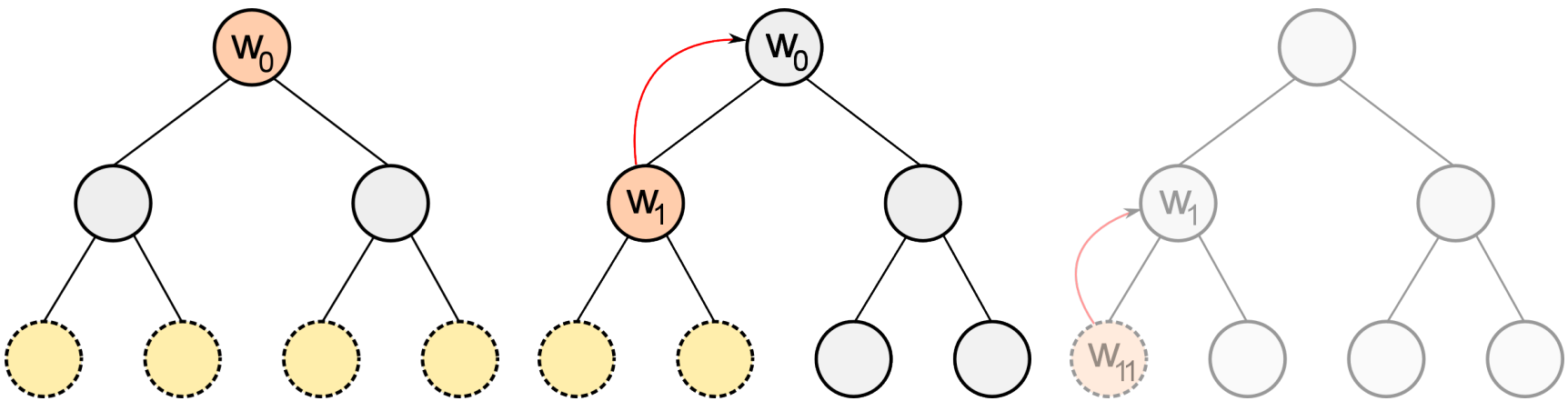


(a) Top-level training (b) Inner training (c) Taxon training

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification

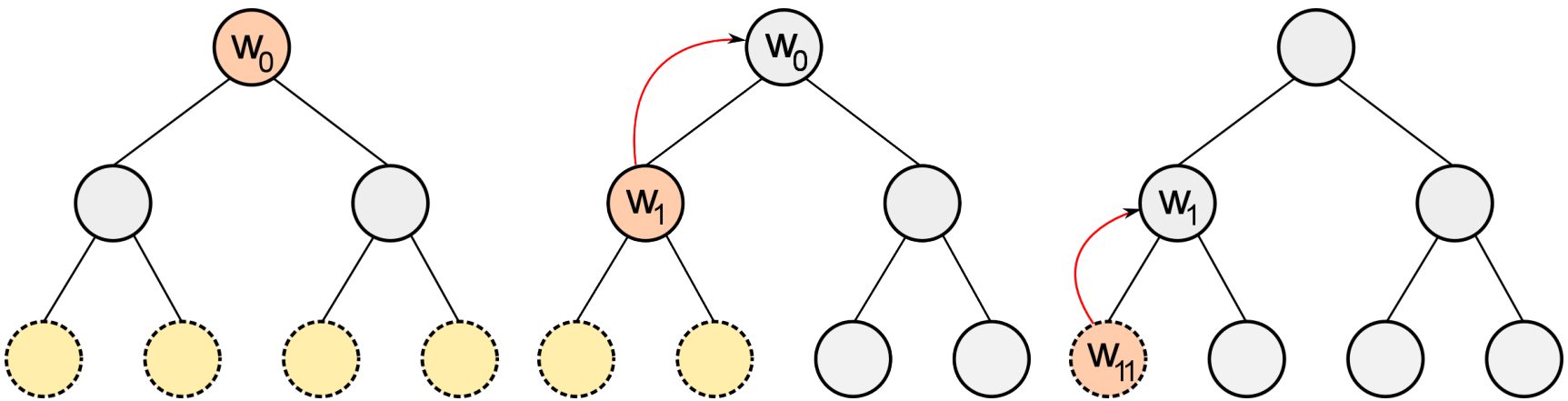


(a) Top-level training (b) Inner training (c) Taxon training

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification



(a) Top-level training (b) Inner training (c) Taxon training

Pairwise Approach

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_M} \quad \frac{1}{2}\sum_{t=1}^{M}\sum_{s=1}^{M}\gamma_{t,s}\|\mathbf{w}_t-\mathbf{w}_s\|^2+\sum_{t=1}^{M}C_t\sum_{(\mathbf{x},y)\in D_t}\ell\left(\langle\mathbf{x},\mathbf{w}_t\rangle,y\right).$$

where ℓ is the hinge loss, $\ell(z,y)=\max\{1-yz,0\}$.

- Train all classifiers $\mathbf{w}_i$ at the same time
- Loss is evaluated independently on datasets D_i
- Similarity is enforced via pairwise regularization term
- Task closeness controlled by $\gamma_{t,s}$

$\Rightarrow$ efficient solution possible via decomposition

Evgeniou et al. [2]

Multitask Kernel Approach

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \hat{k}(\mathbf{x}_i,\mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t. } 0 \leq \alpha_i \leq C \;\forall i \in [1,n]$$

$$\alpha^T \mathbf{y} = 0,$$

where

$$\hat{k}((x_i,s),(x_j,t)) = \underbrace{k_{\text{task}}(s,t)}_{\gamma_{t,s}} \cdot k(x_i,x_j)$$

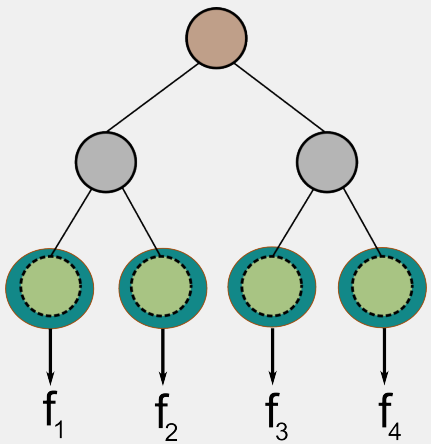
- Easily implemented by altering existing kernel functions (WDK)
- Reuse existing kernel algorithms (SVM)

Daume III [1], Vert et al. [3]

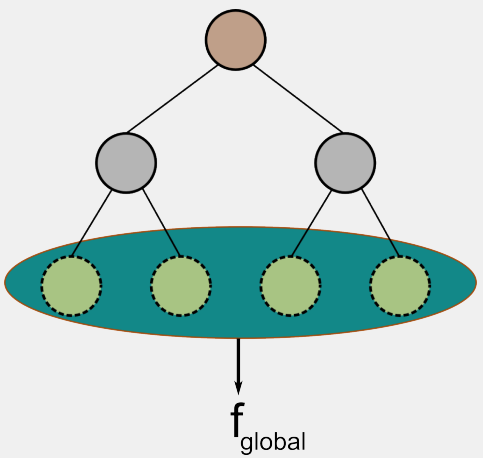
Method Summary

- Multitask Learning Methods
  - Top-Down
  - Pairwise Regularization
  - Multitask Kernel
- Additional Baselines

Plain

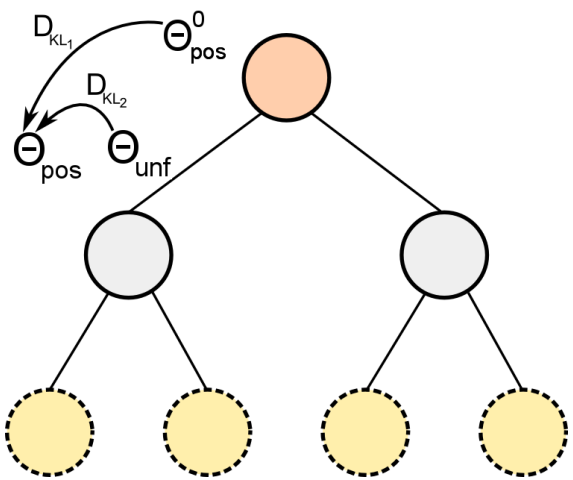


Union

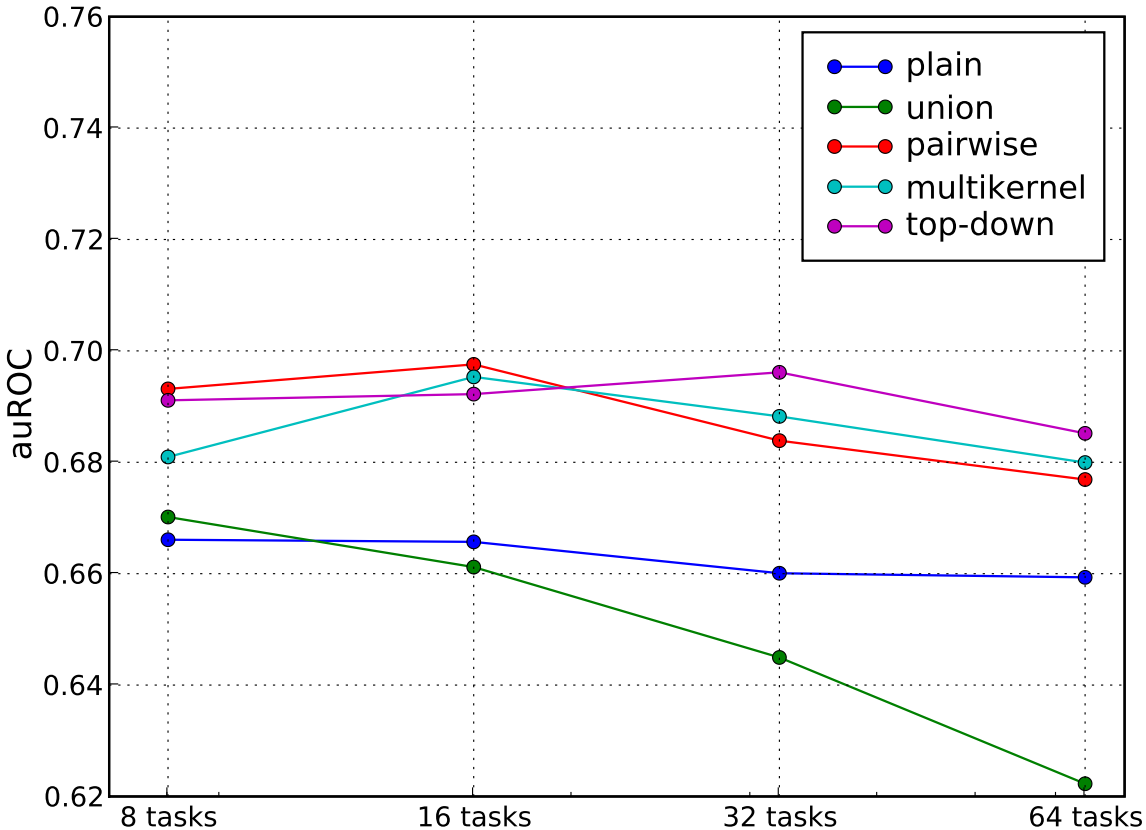


Toy Data Generation

- Sample sequences for M tasks
- Evolve generative model (PSSM) according to given structure
- We want to control
  - How hard the problems are
  - How different the problems are
  - $\rightarrow$ use relative entropy $D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$



Results Toy



- MTL methods outperform baselines
- High mutation rate leads to decreased performance in deep hierarchies

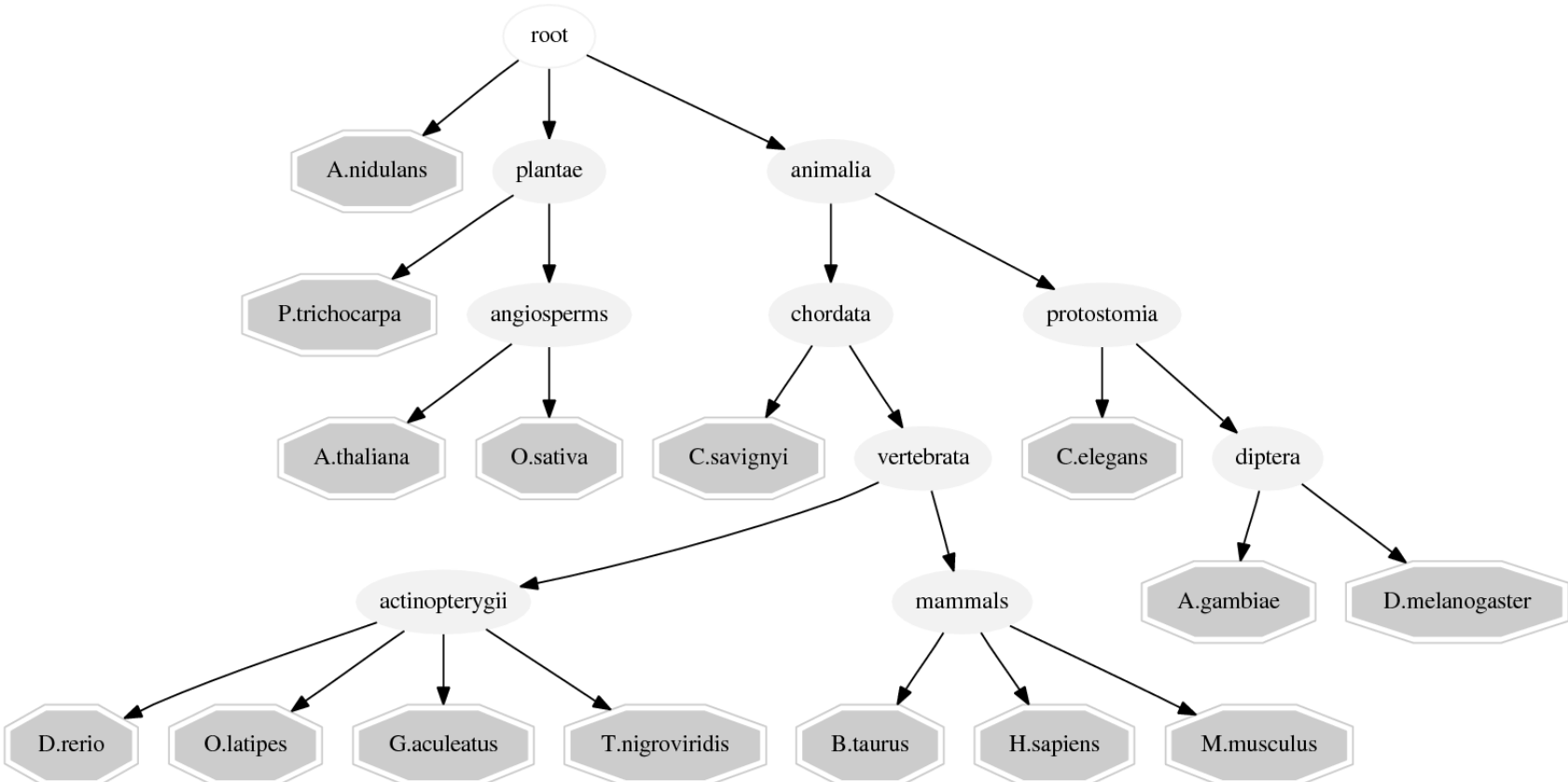
Application to splice-site recognition

- Formulate as binary classification problem

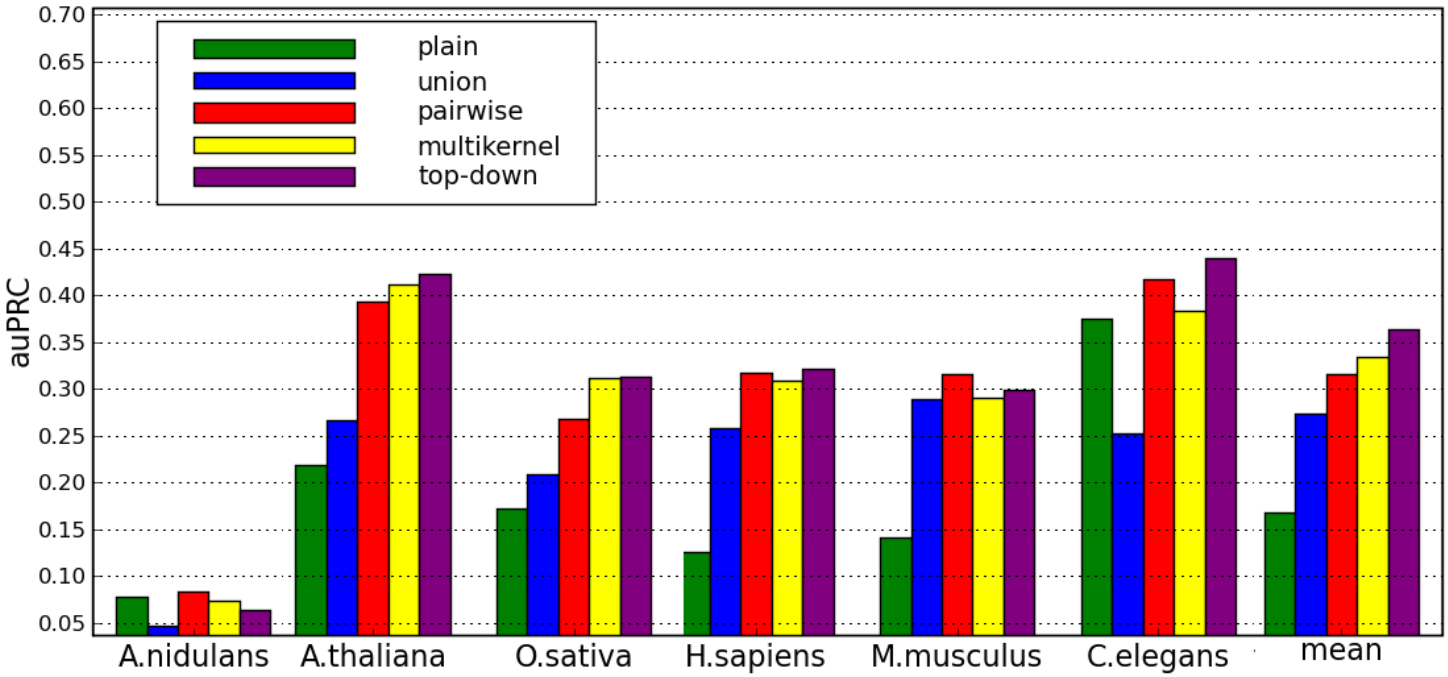
≈150 nucleotides window around dimer

```
CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA
```

- Utilize 15 organisms related by taxonomy



Results Splicing Data



- MTL methods outperform baselines
- Best performer is Top-Down
- Potential increase depends on the particular organism

Outlook

- Techniques for estimation of edge lengths
- Apply to Promoter prediction: ARTS [5] $\Rightarrow$ ARTS.MTL
- Extension to structured output learning

Summary

- Presented three MTL methods
  - Two extensions of existing MTL methods
  - One novel method
- Demonstrated performance on two datasets
  - Toy Data: Robustness
  - Splicing Data: Outperform baselines
- Implementations available in Shogun [6]
- Relevant to many application in Comp-Bio

Summary



- Presented three MTL methods
  - Two extensions of existing MTL methods
  - One novel method
- Demonstrated performance on two datasets
  - Toy Data: Robustness
  - Splicing Data: Outperform baselines
- Implementations available in Shogun [6]
- Relevant to many application in Comp-Bio

Summary



- Presented three MTL methods
  - Two extensions of existing MTL methods
  - One novel method
- Demonstrated performance on two datasets
  - Toy Data: Robustness
  - Splicing Data: Outperform baselines
- Implementations available in Shogun [6]
- Relevant to many application in Comp-Bio

Summary



- Presented three MTL methods
  - Two extensions of existing MTL methods
  - One novel method
- Demonstrated performance on two datasets
  - Toy Data: Robustness
  - Splicing Data: Outperform baselines
- Implementations available in Shogun [6]
- Relevant to many application in Comp-Bio

Acknowledgments

- Jose Leiva [1,2]
- Yasemin Altun [2]
- Sören Sonnenburg [2]
- Gabrielle Schweikert [1,2]
- Bernhard Schölkopf [2]
- Klaus Robert Müller [3]
- Gunnar Rätsch [1]

1 Friedrich Miescher Laboratory of the Max Planck Society
2 Max Planck Institute for Biological Cybernetics
3 Berlin Institute of Technology

Thank you for your attention.

Acknowledgments

- Jose Leiva [1,2]
- Yasemin Altun [2]
- Sören Sonnenburg [2]
- Gabrielle Schweikert [1,2]
- Bernhard Schölkopf [2]
- Klaus Robert Müller [3]
- Gunnar Rätsch [1]

1 Friedrich Miescher Laboratory of the Max Planck Society
2 Max Planck Institute for Biological Cybernetics
3 Berlin Institute of Technology

Thank you for your attention.

References I



H. Daumé.

Frustratingly easy domain adaptation.

In *ACL*. The Association for Computer Linguistics, 2007.



T. Evgeniou, C.A. Micchelli, and M. Pontil.

Learning multiple tasks with kernel methods.

Journal of Machine Learning Research, 6:615–637, 2005.



L. Jacob and J.P. Vert.

Efficient peptide-MHC-I binding prediction for alleles with few known binders.

Bioinformatics, 24(3):358, 2008.

References II



Gabriele Schweikert, Alexander Zien, Georg Zeller, Jonas Behr, Christoph Dieterich, Cheng Soon Ong, Petra Philips, Fabio De Bona, Lisa Hartmann, Anja Bohlen, Nina Krüger, Sören Sonnenburg, and Gunnar Rätsch.

mGene: accurate SVM-based gene finding with an application to nematode genomes.

Genome research, 19(11):2133–43, November 2009.



S. Sonnenburg, A. Zien, and G. Ratsch.

ARTS: accurate recognition of transcription starts in human.

Bioinformatics, 22(14):e472, 2006.

References III



Sören Sonnenburg, Gunnar Rätsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtech Franc.

The shogun machine learning toolbox.

Journal of Machine Learning Research, 2010.

accepted.