Transfer Learning and Applications in Computational Biology

Gunnar Rätsch[1], **Christian Widmer**[1,2], **Marius Kloft**[1,3,4]

Nico Görnitz, Gabriele Schweikert

[1] Memorial Sloan-Kettering Cancer Center, NY, USA

[2] Microsoft Research, Los Angeles, USA

[3] Courant Institute, NYU, New York, USA

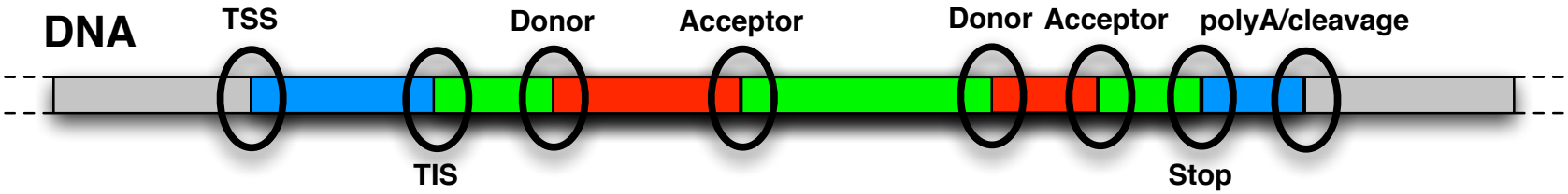
[4] Humbolt University, Berlin, Germany

Memorial Sloan-Kettering Cancer Center

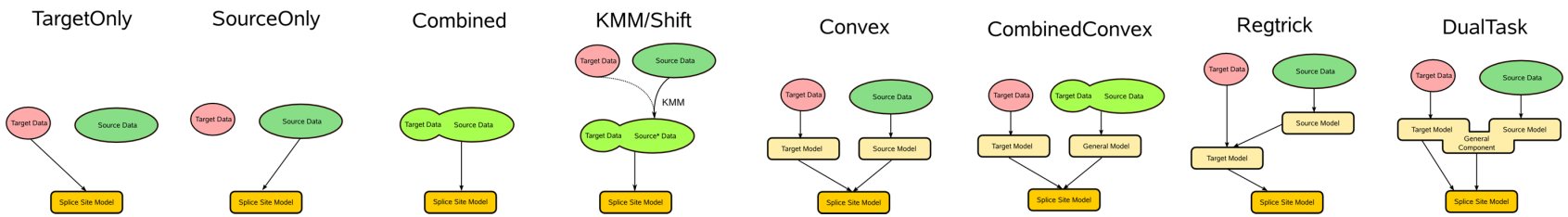
cBio@MSKCC

Roadmap

- Motivation from computational biology



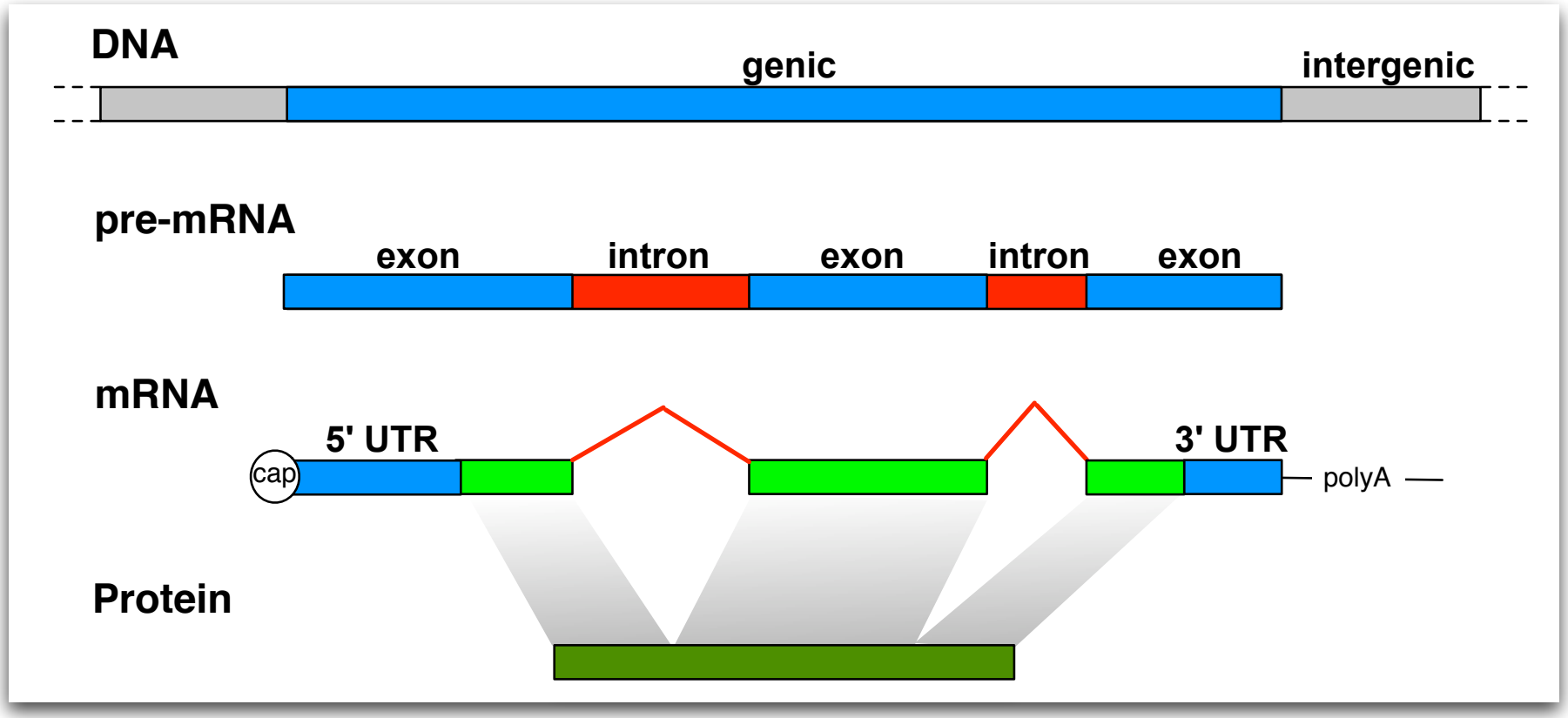
- Empirical comparison of domain adaptation algorithms



- Algorithms for hierarchical multi-task learning
- Algorithms for learning task relations
- Fast(er) Algorithms
- Discussion & Conclusion

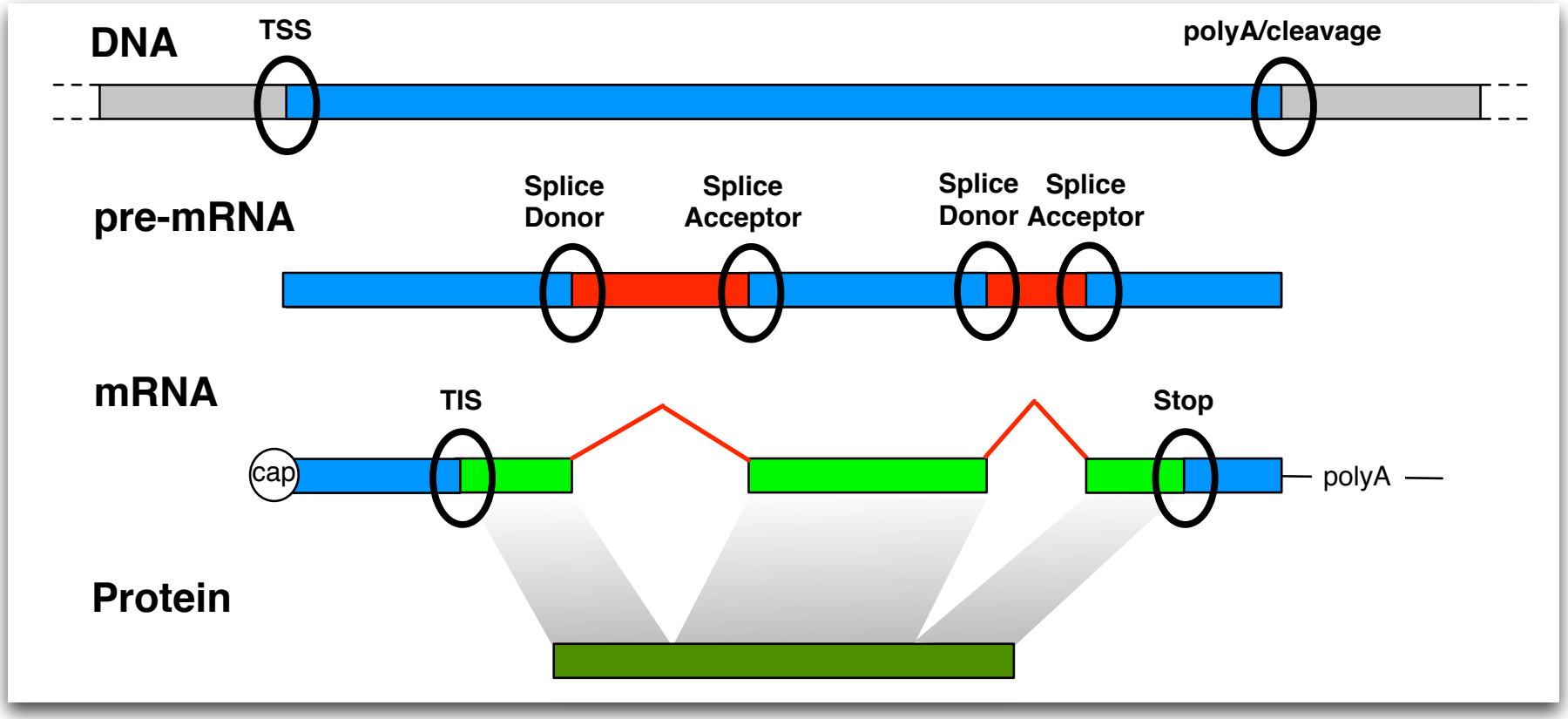


A Core CompBio Problem: Gene Finding



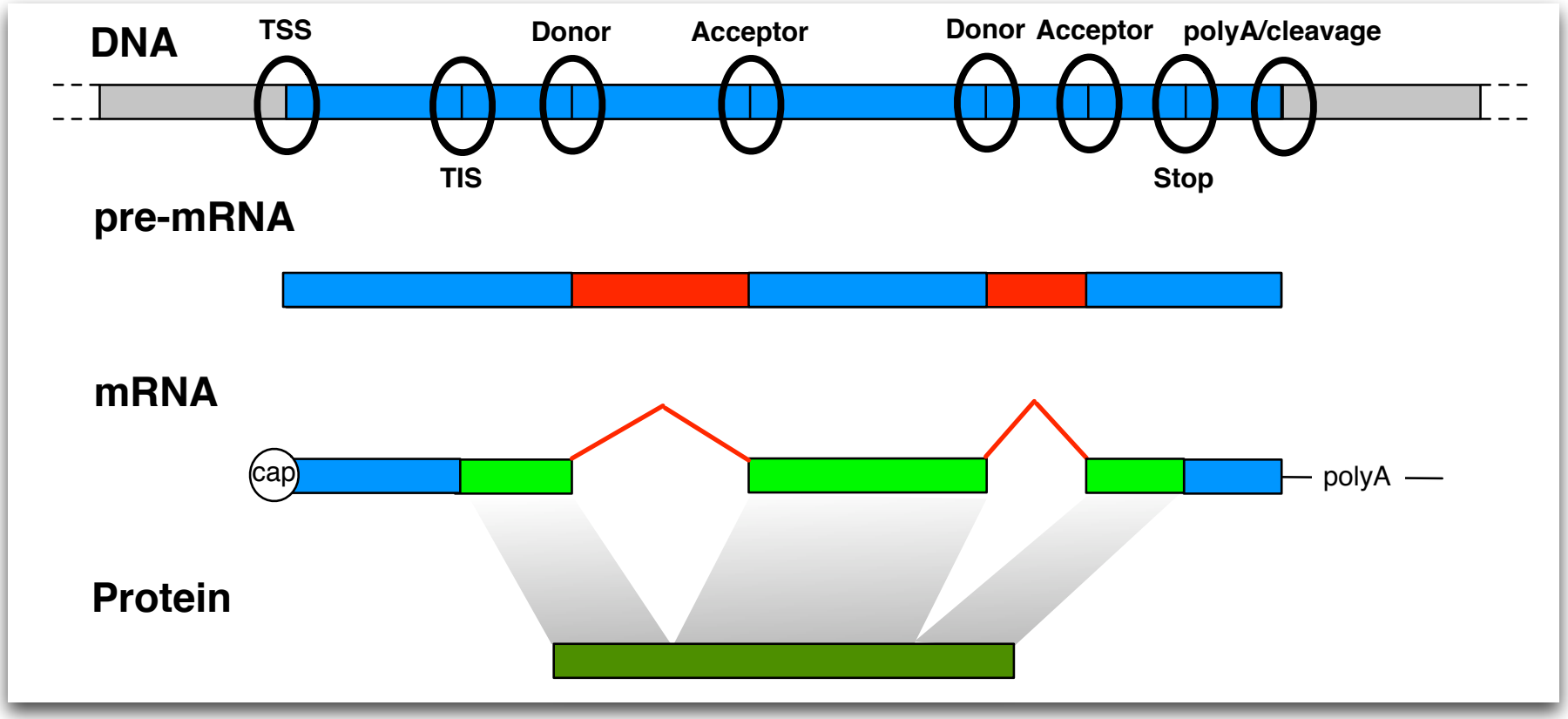
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

A Core CompBio Problem: Gene Finding



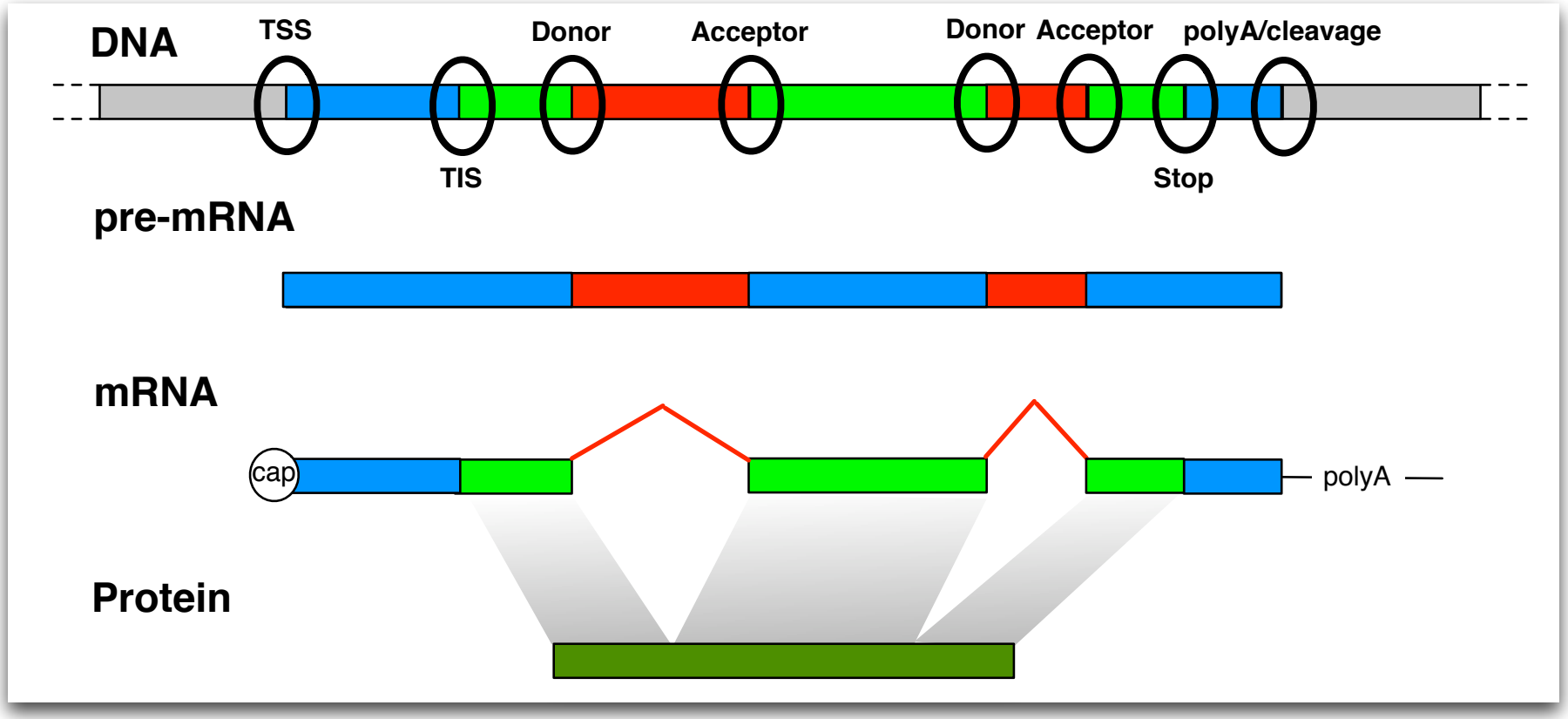
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

A Core CompBio Problem: Gene Finding



- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

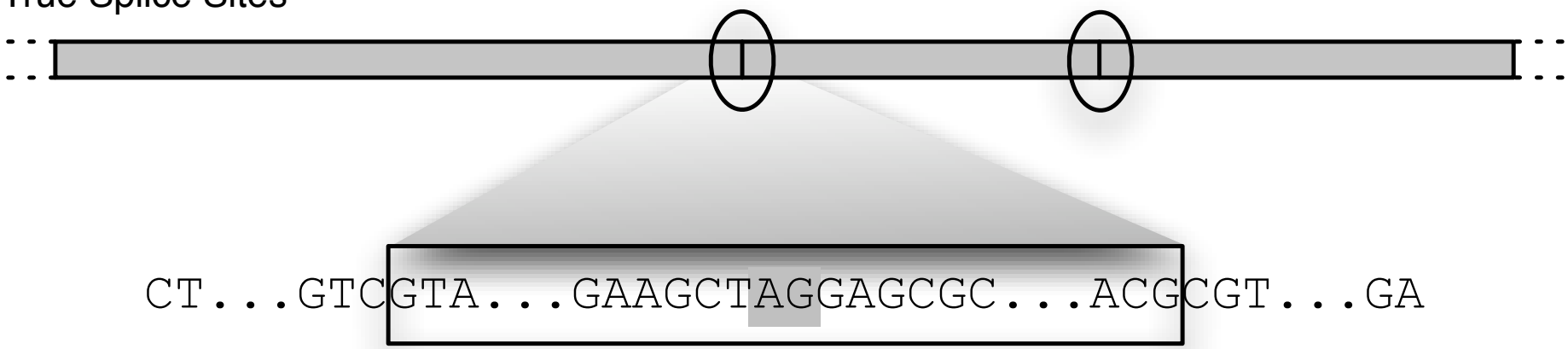
A Core CompBio Problem: Gene Finding



- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Example: Splice Site Recognition

True Splice Sites



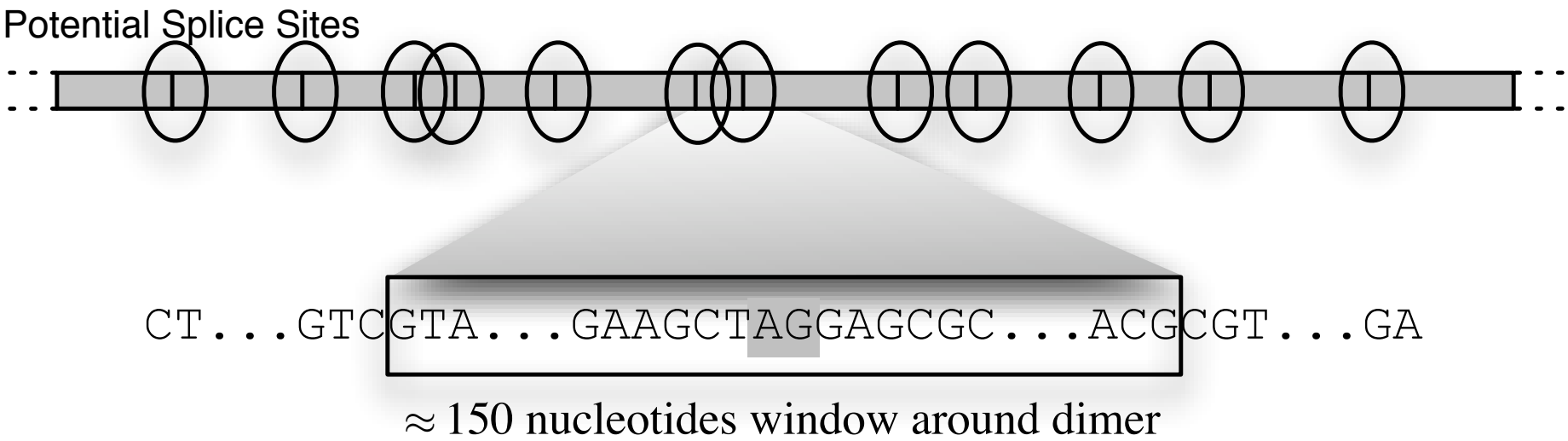
CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer



```
1  GCCAATATTTTTCTATTCAGGTGCAATCAATCACCCATCAT
1  ATTGAATGAACATATTCCAGGGTCTCCTTCCACCTCAACAA
1  AGCAACGAACTCCATTACAGCAAGGACATCGAAGTCGATCA
1  GCCAATTTTTGACCTTGCAGAATCAATCGTGCACGTTCGGA
```

Example: Splice Site Recognition



1	GCCAATATTTTTCTATTCAGGTGCAATCAATCACCCATCAT
1	ATTGAATGAACATATTCCAGGGTCTCCTTCCACCTCAACAA
1	AGCAACGAACTCCATTACAGCAAGGACATCGAAGTCGATCA
1	GCCAATTTTTGACCTTGCAGAATCAATCGTGCACGTTCGGA
−1	CATCTGAAATTTCCCCCAAGTATAGCGGAAATAGACCGACG
−1	GAAATTTCCCCCAAGTATAGCGGAAATAGACCGACGAAATC
−1	CCCAAGTATAGCGGAAATAGACCGACGAAATCGCTCTCTCC
−1	AATCGCTCTCTCCCTGGGAGCGATGCGAATGTCAAATTCGA
−1	ACCAAAAAATCAATTTTTAGATTTTTCGAATTAATTTTTCG
−1	TGCTTTGCATGTTTCTAAAGTTACAGCCGTTCAAAATTTAA
−1	GCATGTTTCTAAAGTTACAGCCGTTCAAAATTTAAAAACTC
−1	ACCAATACGCAATGACTGAGTCTGTAATTTCACATAGTAAT

Domain Adaptation for Genome Annotation

Motivation:

- Increasing number of sequenced genomes
- Often newly sequenced genomes are poorly annotated
- However often relatives with good annotation exist
- **Idea:** Transfer knowledge between organisms

Example: Splice site annotation in worm genomes (≈2010)

- Newly sequenced organism: *C. briggsae*
  - ≈ 100 confirmed genes (590 splice site pairs)
- Well annotated relative: *C. elegans*
  - ≈ 10.000 confirmed genens (36.782 splice site pairs)

Domain Adaptation for Genome Annotation

Motivation:

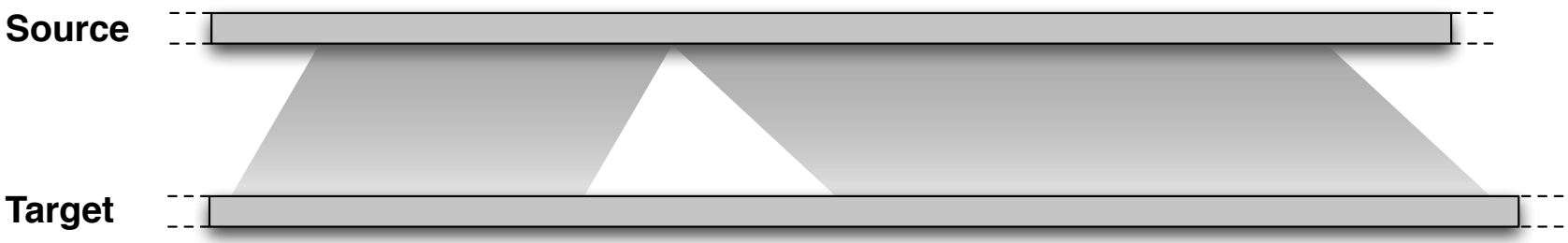
- Increasing number of sequenced genomes
- Often newly sequenced genomes are poorly annotated
- However often relatives with good annotation exist
- **Idea:** Transfer knowledge between organisms

Example: Splice site annotation in worm genomes (≈2010)

- Newly sequenced organism: *C. briggsae*
  - ≈ 100 confirmed genes (590 splice site pairs)
- Well annotated relative: *C. elegans*
  - ≈ 10.000 confirmed genens (36.782 splice site pairs)

The "Bioinformatics Way" of Transfer Learning

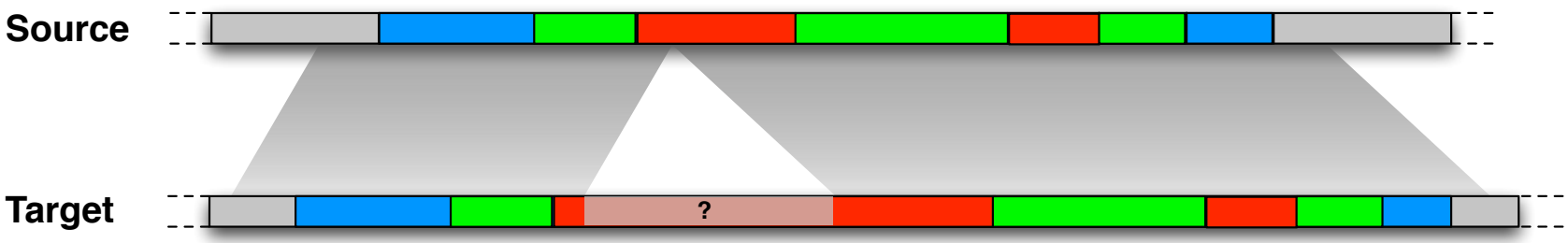
1. Homology-based annotation (a.k.a. "Comparative genomics")



Works for closely related species, does not require any labeled data from target organism.

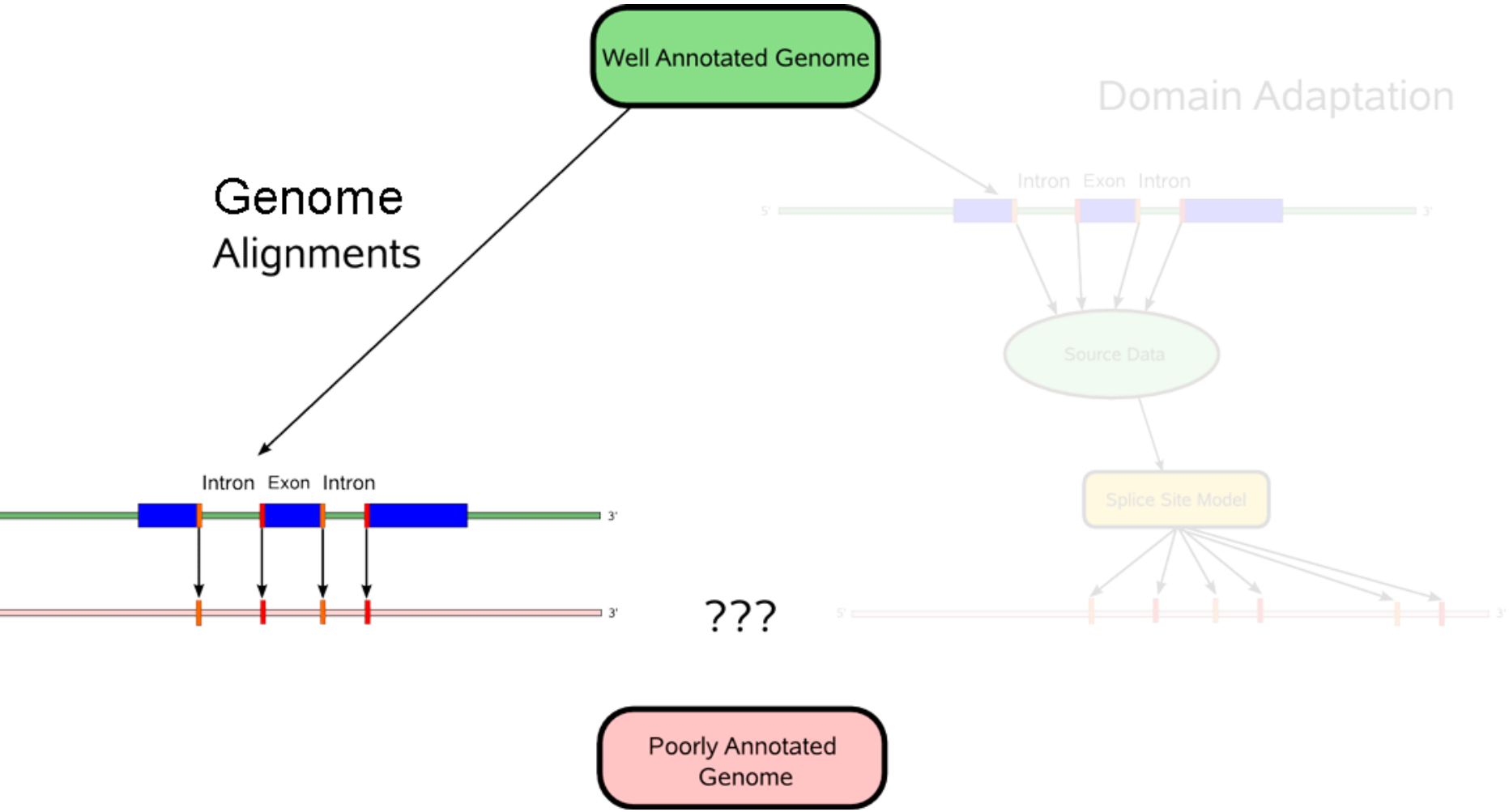
The "Bioinformatics Way" of Transfer Learning

1. Homology-based annotation (a.k.a. "Comparative genomics")



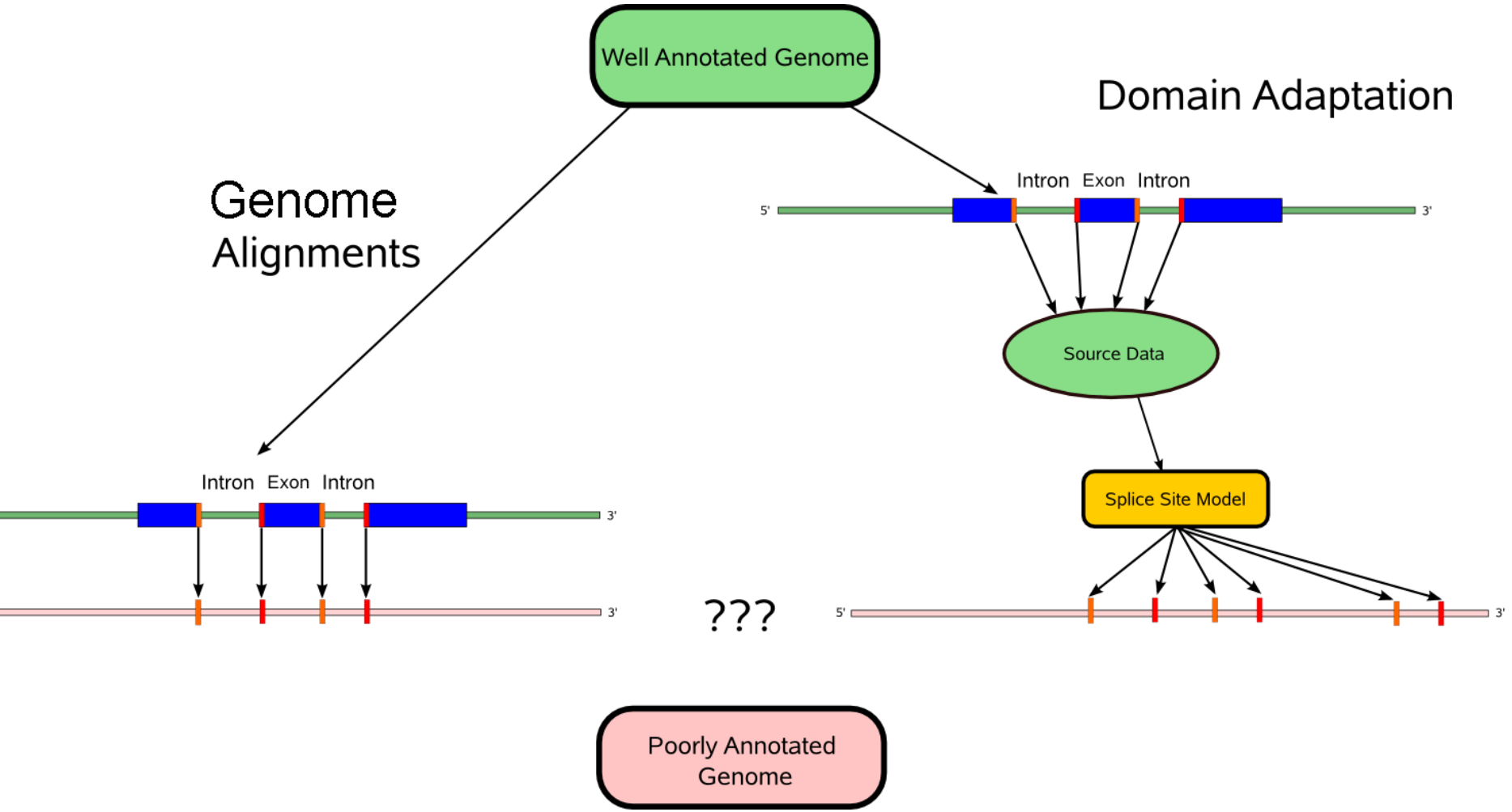
Works for closely related species, does not require any labeled data from target organism.

Domain Adaptation by Learning vs. Homology



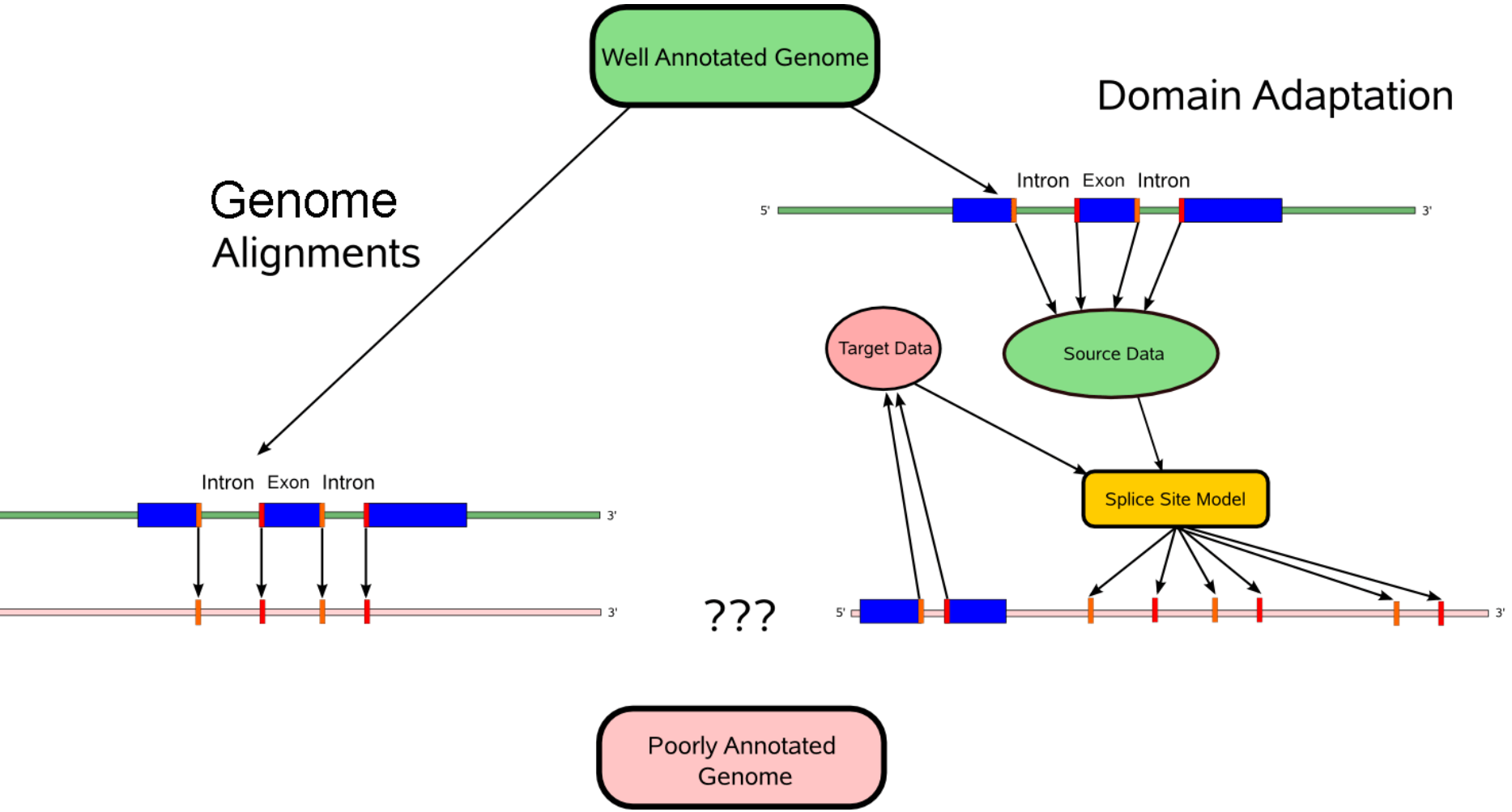
[Schweikert et al., 2008; Widmer et al., 2010b]

Domain Adaptation by Learning vs. Homology



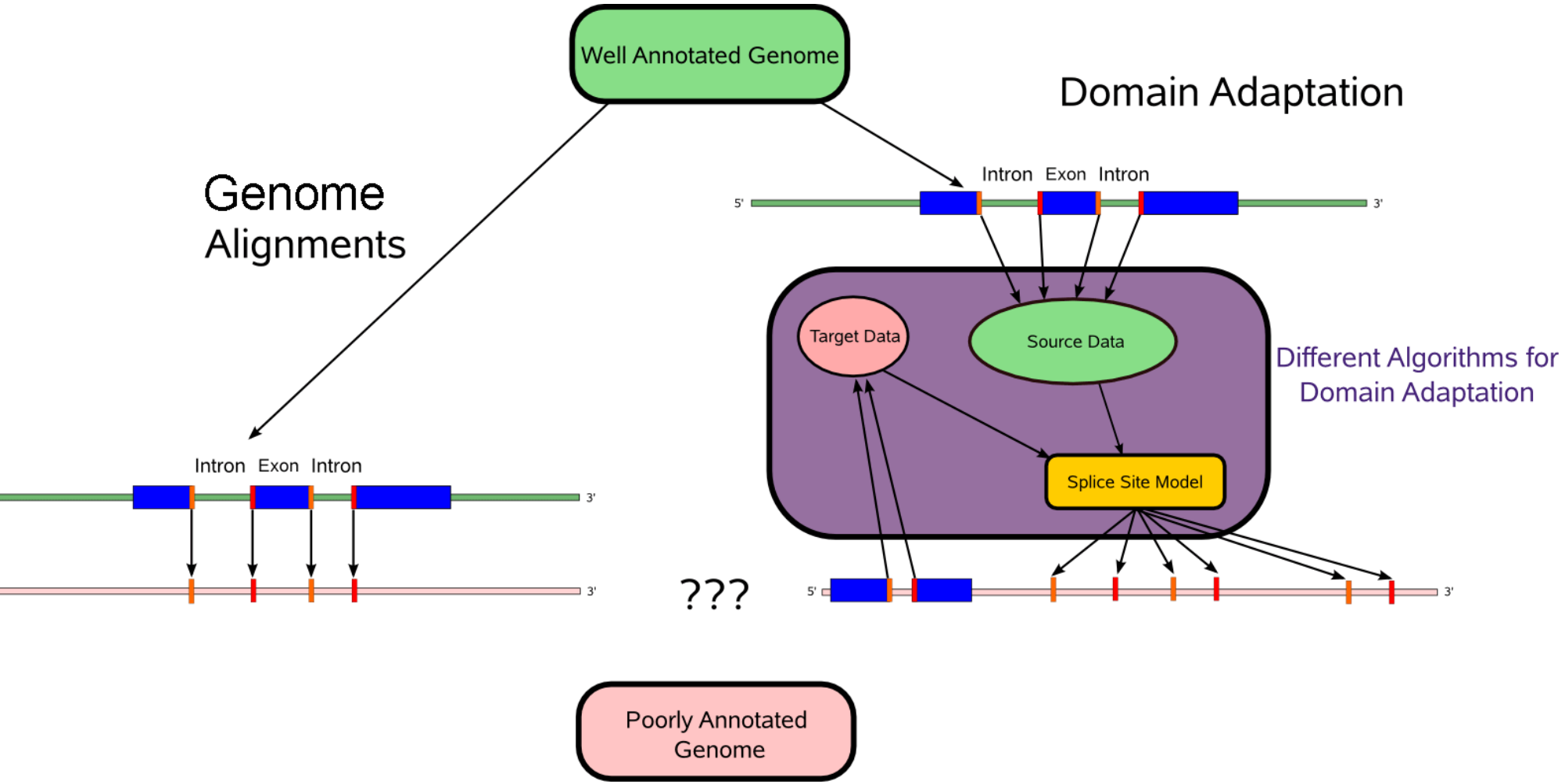
[Schweikert et al., 2008; Widmer et al., 2010b]

Domain Adaptation by Learning vs. Homology



[Schweikert et al., 2008; Widmer et al., 2010b]

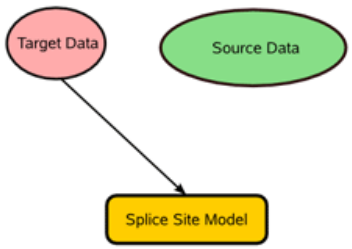
Domain Adaptation by Learning vs. Homology



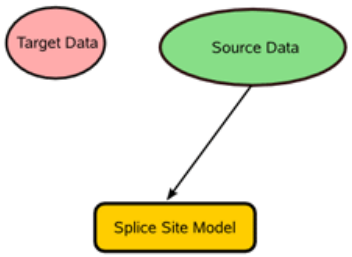
[Schweikert et al., 2008; Widmer et al., 2010b]

Domain Adaptation Algorithms Overview

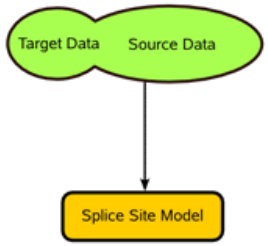
TargetOnly



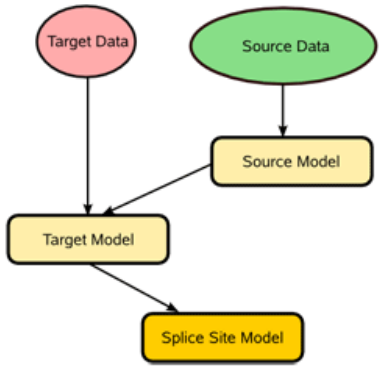
SourceOnly



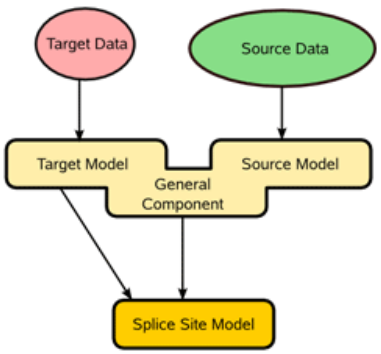
Combined



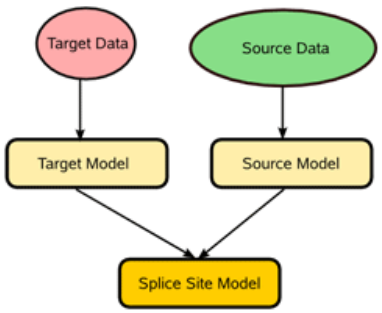
Regtrick



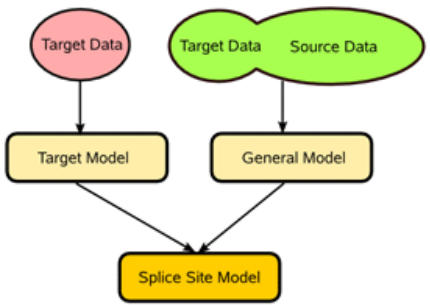
DualTask



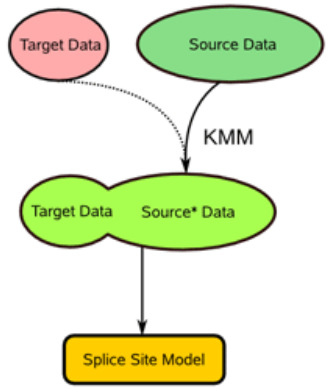
Convex



CombinedConvex

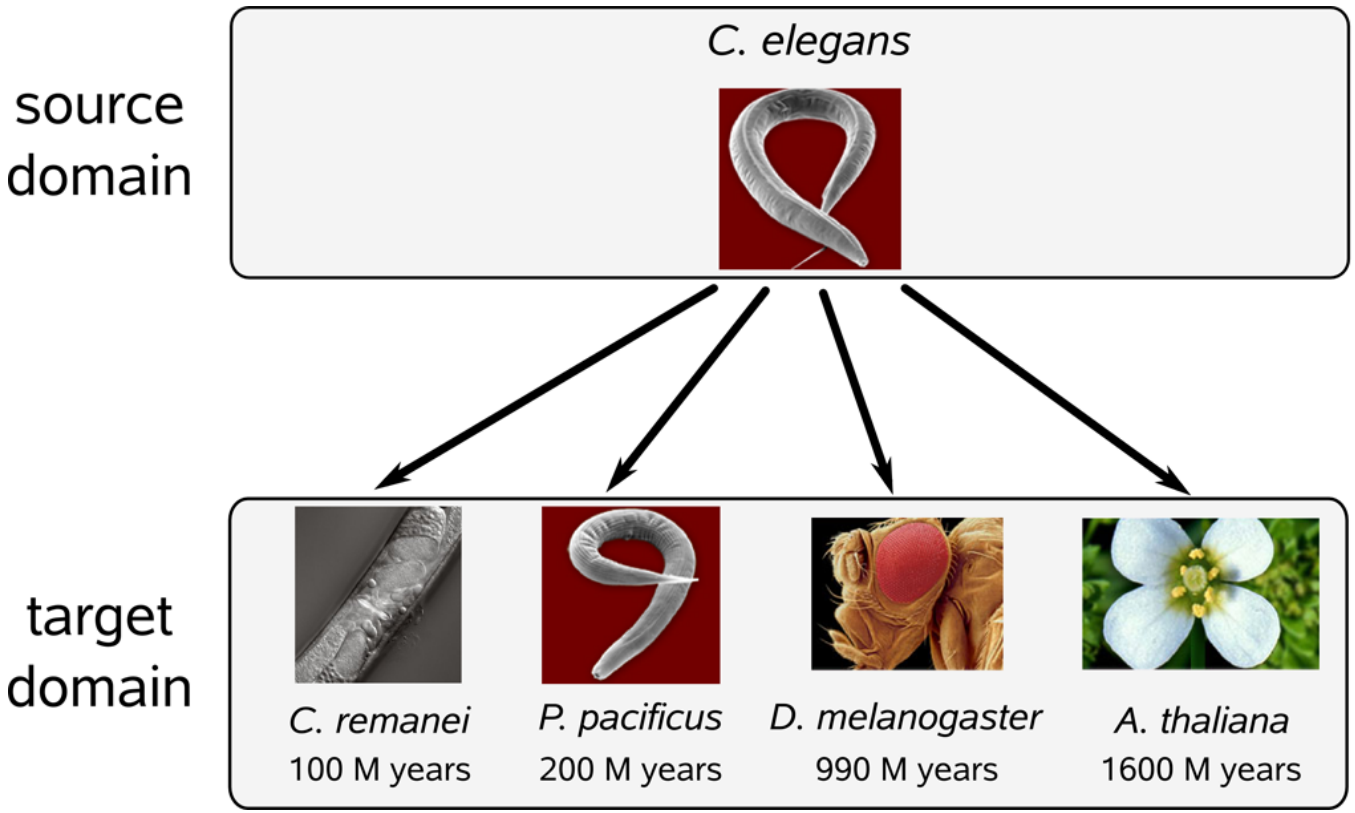


KMM/Shift



[Schweikert et al., 2008]

Large-Scale Empirical Comparison



- Varying distances
- Different data set sizes

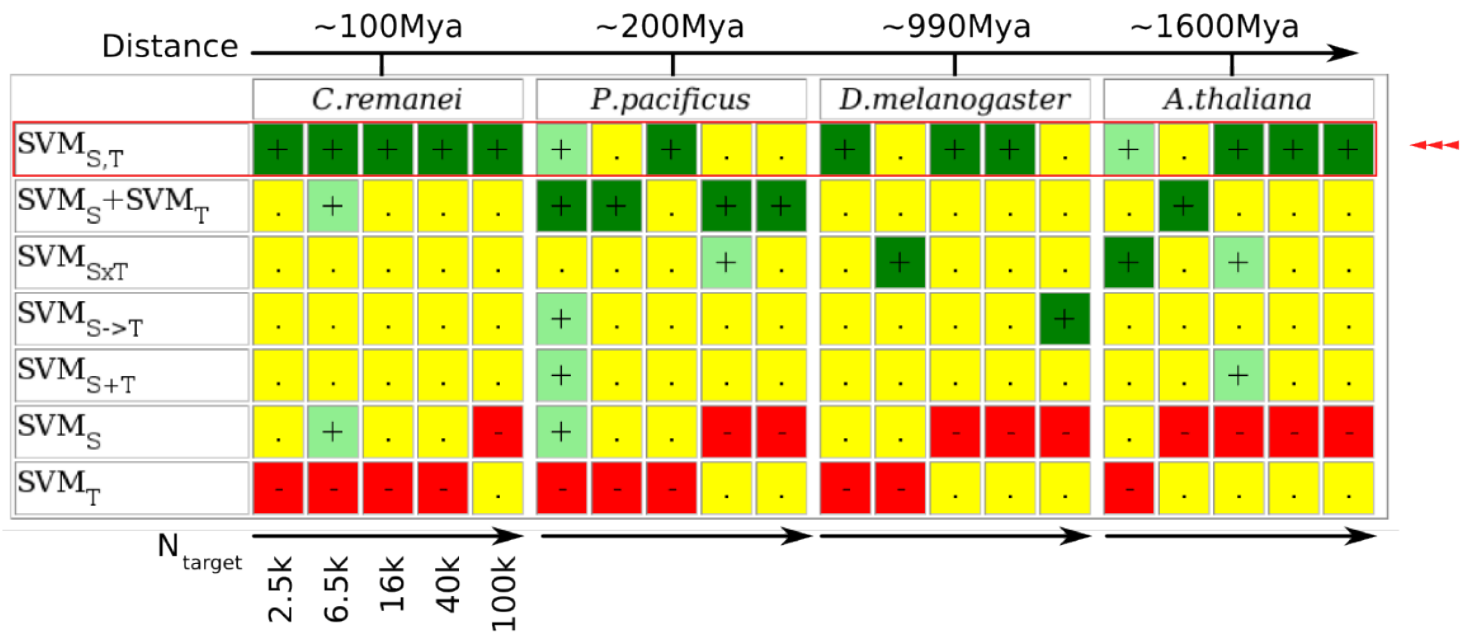
[MPI Developmental Biology and UCSC Genome Browser]

Experimental Setup

- Source dataset size: always 100k examples
- Target dataset sizes: $\{2500, 6500, 16000, 64000, 100000\}$
- Simple kernel (WDK of degree 1 $\Rightarrow$ under-fitting)
- Extensive model selection for each method
- Area under Precision/Recall curve for evaluation

Domain Adaptation Results Summary

- Considerable improvements possible
- Sophisticated domain adaptation methods needed on distantly related organisms
- Best overall performance has **DualTask**
- Most cost effective **Convex/AdvancedConvex**

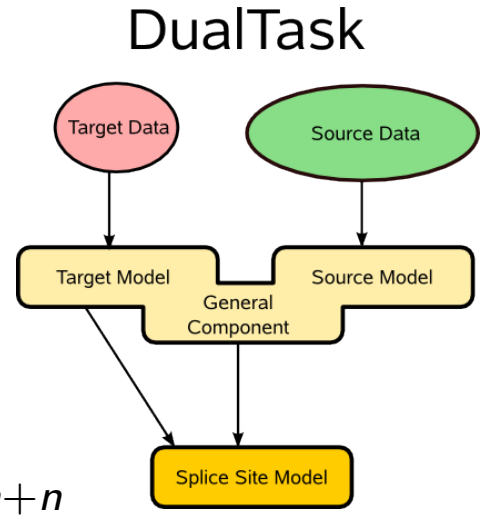


[Schweikert et al., 2008]

Domain Adaptation Methods

Idea: [e.g., Caruana, 1997]

- Simultaneous optimization of both models
- Similarity between solution enforced



Approach:

$$\min_{\boldsymbol{w}_S, \boldsymbol{w}_T, \xi} \quad \frac{1}{2}\|\boldsymbol{w}_S\|^2 + \frac{1}{2}\|\boldsymbol{w}_T\|^2 - B\boldsymbol{w}_T^T\boldsymbol{w}_S + C\sum_{i=1}^{m+n}\xi_i$$

$$\text{s.t.} \quad y_i(\langle \boldsymbol{w}_S, \Phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, m$$

$$y_i(\langle \boldsymbol{w}_T, \Phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \xi_i \qquad i = m+1, \ldots, m+n$$

Equivalent to multi-task kernel learning: [Daume III, 2007]

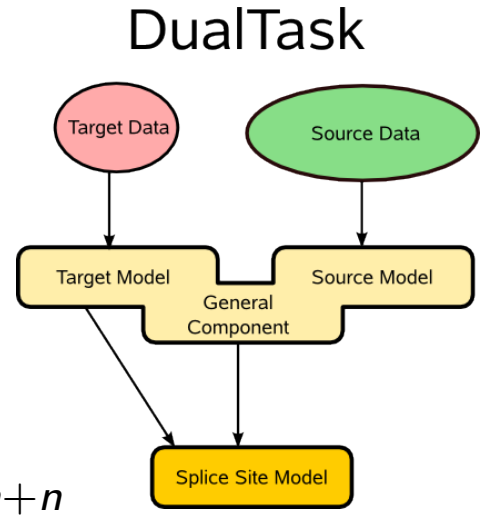
$$K_{MTK}((\boldsymbol{x}, t), (\boldsymbol{x}', t')) = \gamma_{t,t'} K(\boldsymbol{x}, \boldsymbol{x}')$$

for a suitably chosen Γ (p.s.d.).

Domain Adaptation Methods

Idea: [e.g., Caruana, 1997]

- Simultaneous optimization of both models
- Similarity between solution enforced



Approach:

$$\min_{\boldsymbol{w}_S, \boldsymbol{w}_T, \xi} \quad \frac{1}{2}\|\boldsymbol{w}_S\|^2 + \frac{1}{2}\|\boldsymbol{w}_T\|^2 - B\boldsymbol{w}_T^T\boldsymbol{w}_S + C\sum_{i=1}^{m+n}\xi_i$$

$$\text{s.t.} \quad y_i(\langle \boldsymbol{w}_S, \Phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, m$$

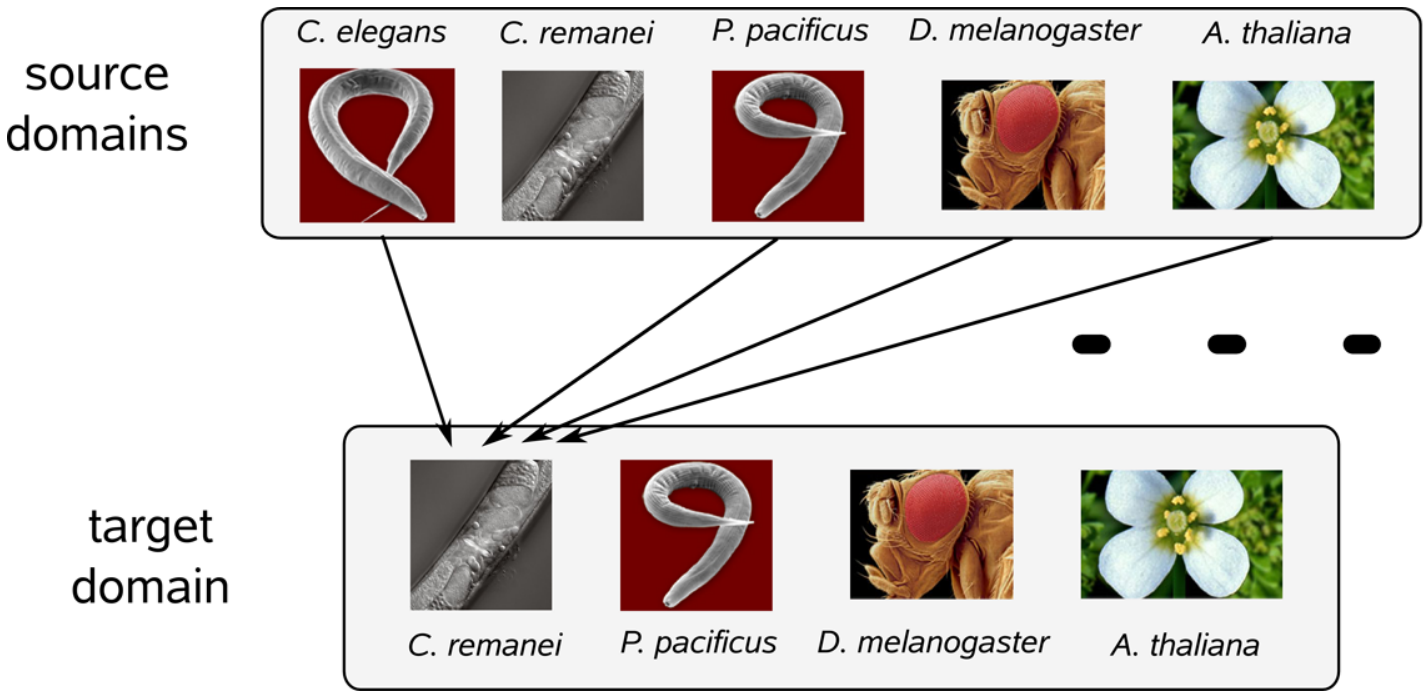
$$y_i(\langle \boldsymbol{w}_T, \Phi(\boldsymbol{x}_i)\rangle + b) \geq 1 - \xi_i \qquad i = m+1, \ldots, m+n$$

Equivalent to multi-task kernel learning: [Daume III, 2007]

$$K_{MTK}((\boldsymbol{x}, t), (\boldsymbol{x}', t')) = \gamma_{t,t'} K(\boldsymbol{x}, \boldsymbol{x}')$$

for a suitably chosen Γ (p.s.d.).

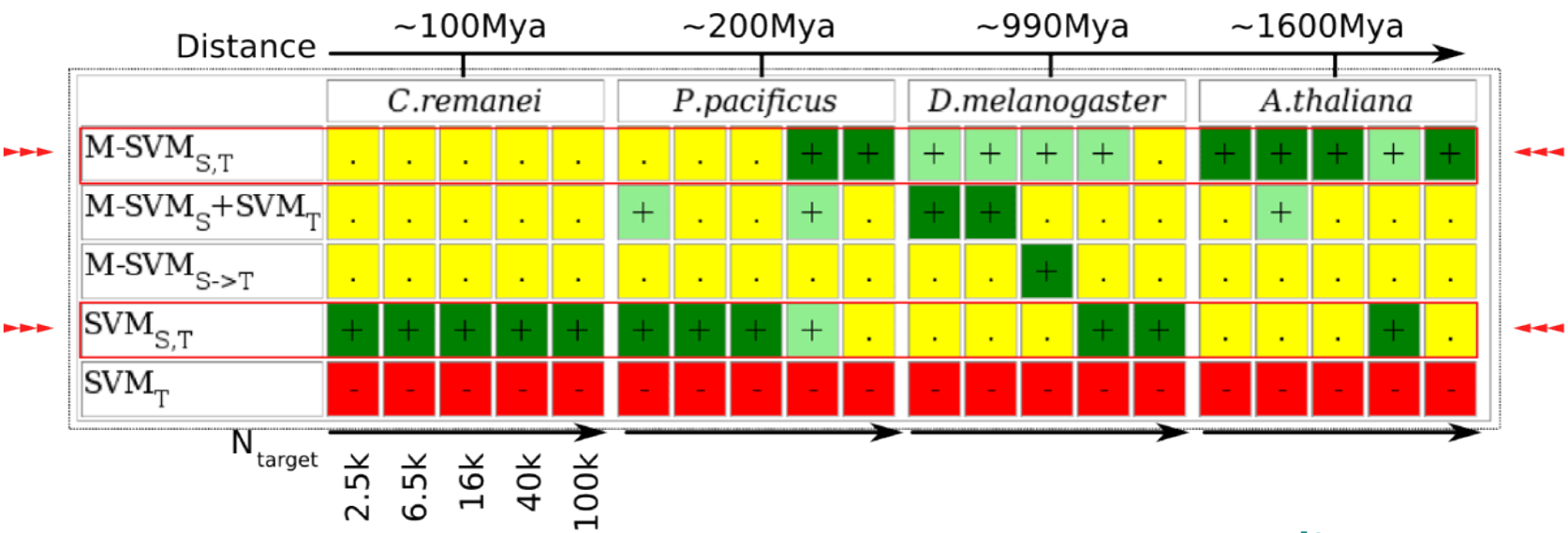
Multiple Source Domains



- Combine information from several sources (treated equally)
- Methods: Multi-task learning, Convex combination, Shifting

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest

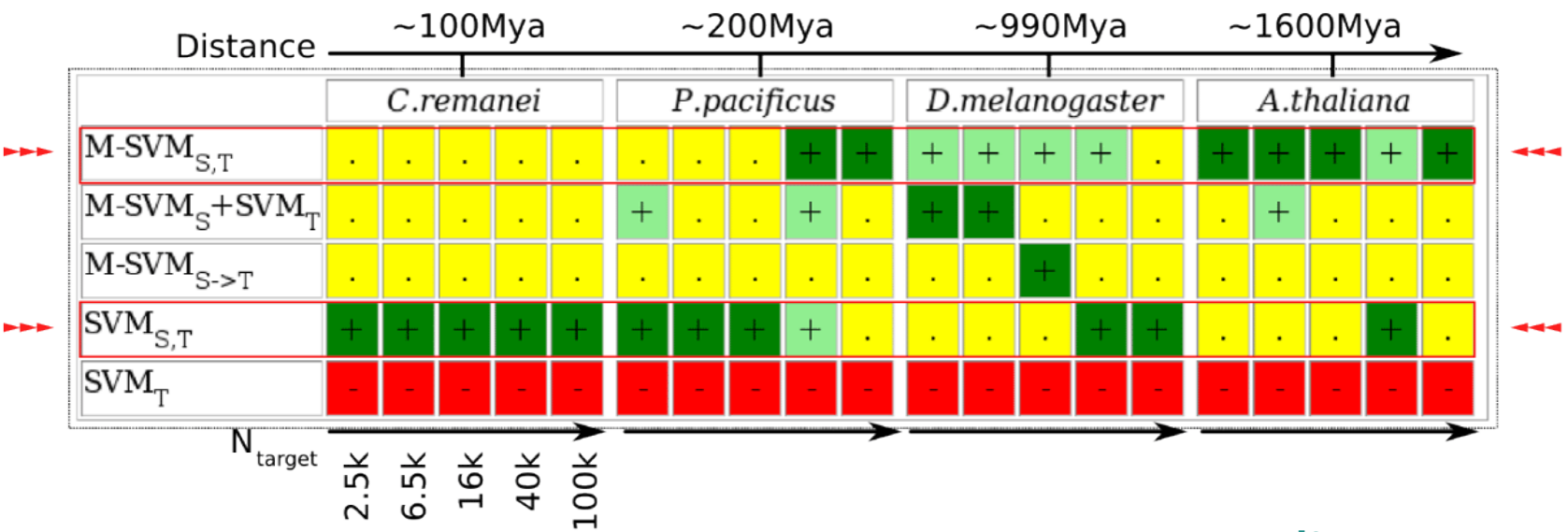


[Schweikert et al., 2008]

Multiple sources can be worse than single source.
How to use information on relatedness in learning?

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest



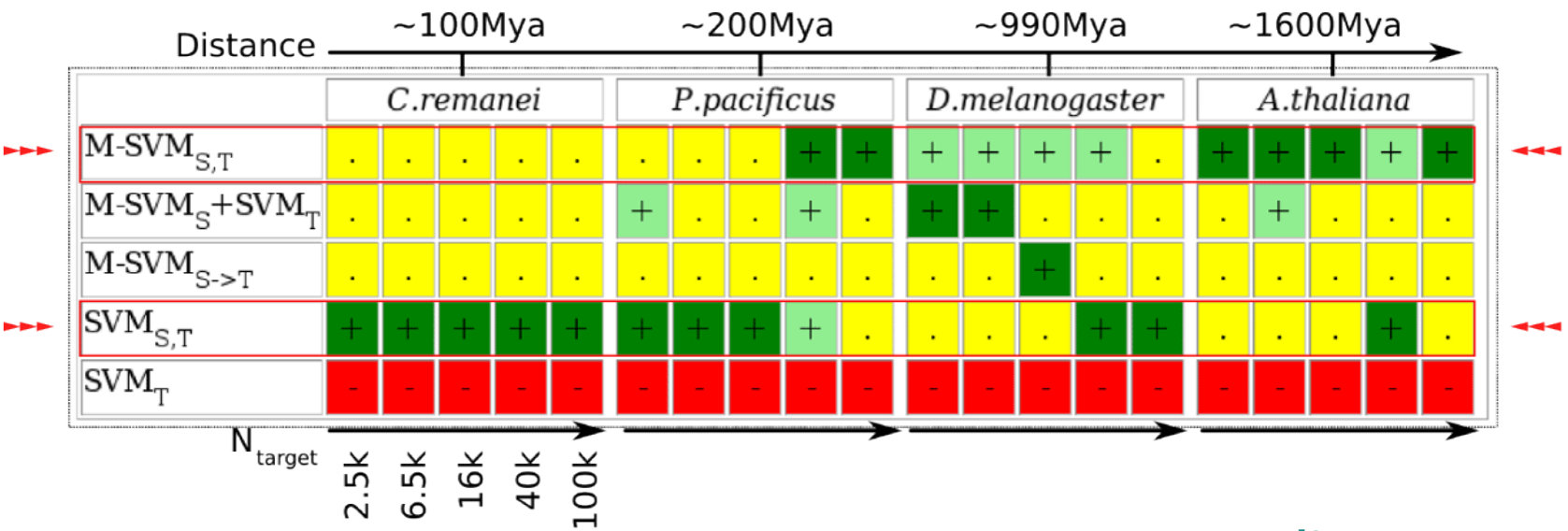
[Schweikert et al., 2008]

Multiple sources can be worse than single source.

How to use information on relatedness in learning?

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest



[Schweikert et al., 2008]

Multiple sources can be worse than single source.
How to use information on relatedness in learning?

Multitask learning

Hierarchical structure arises naturally from the Tree of Life



- Taxonomy defines relationship between tasks
- Closer tasks benefit more from each other

[Widmer et al., 2010a]

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

Taxonomy



Transformation

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

Task Similarity Matrix

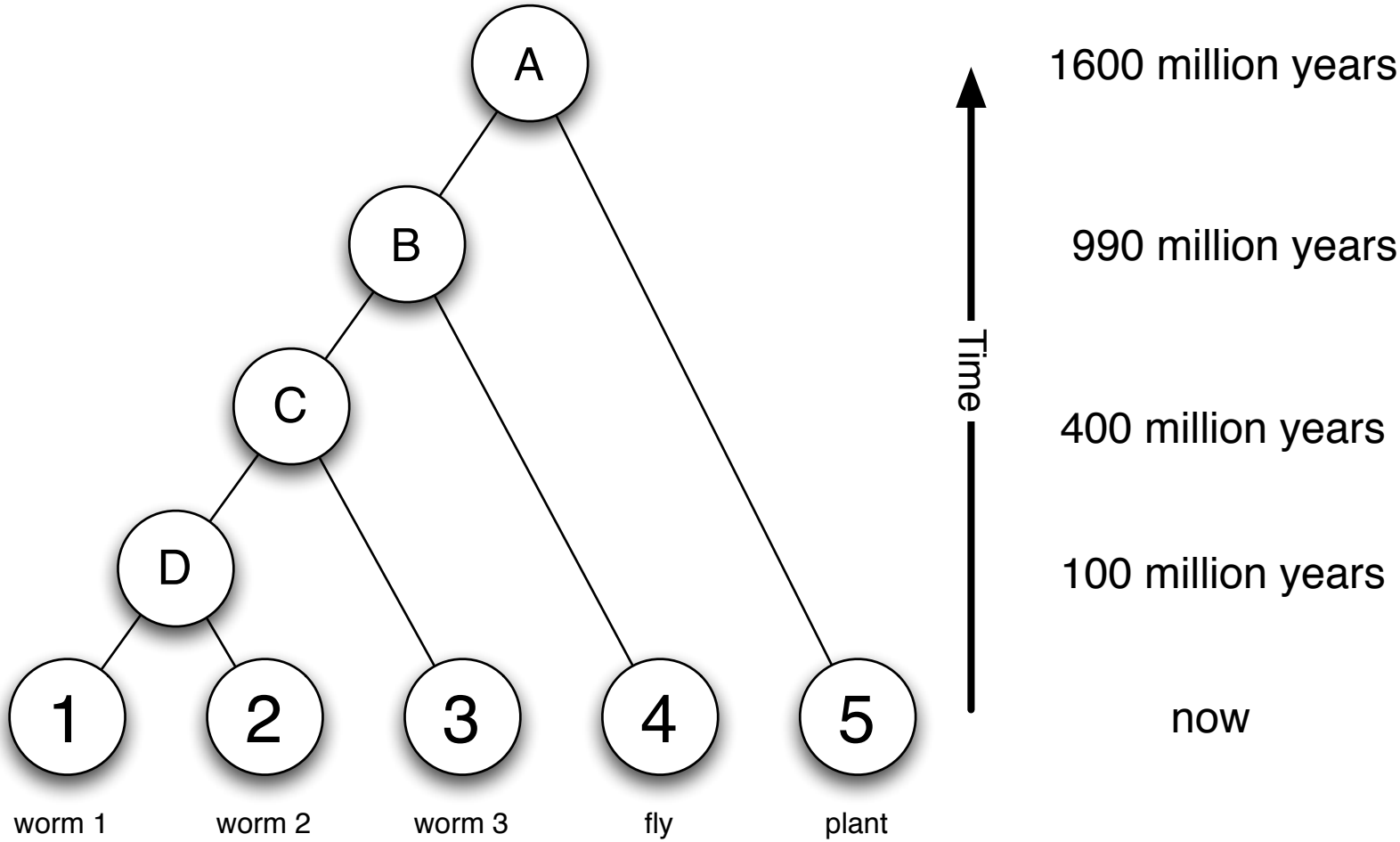
Top-Down

Multitask Kernel

$$K_{MTL}((x,t),(x',t')) = \gamma_{t,t'} K(x,x')$$

[Widmer et al., 2010a]

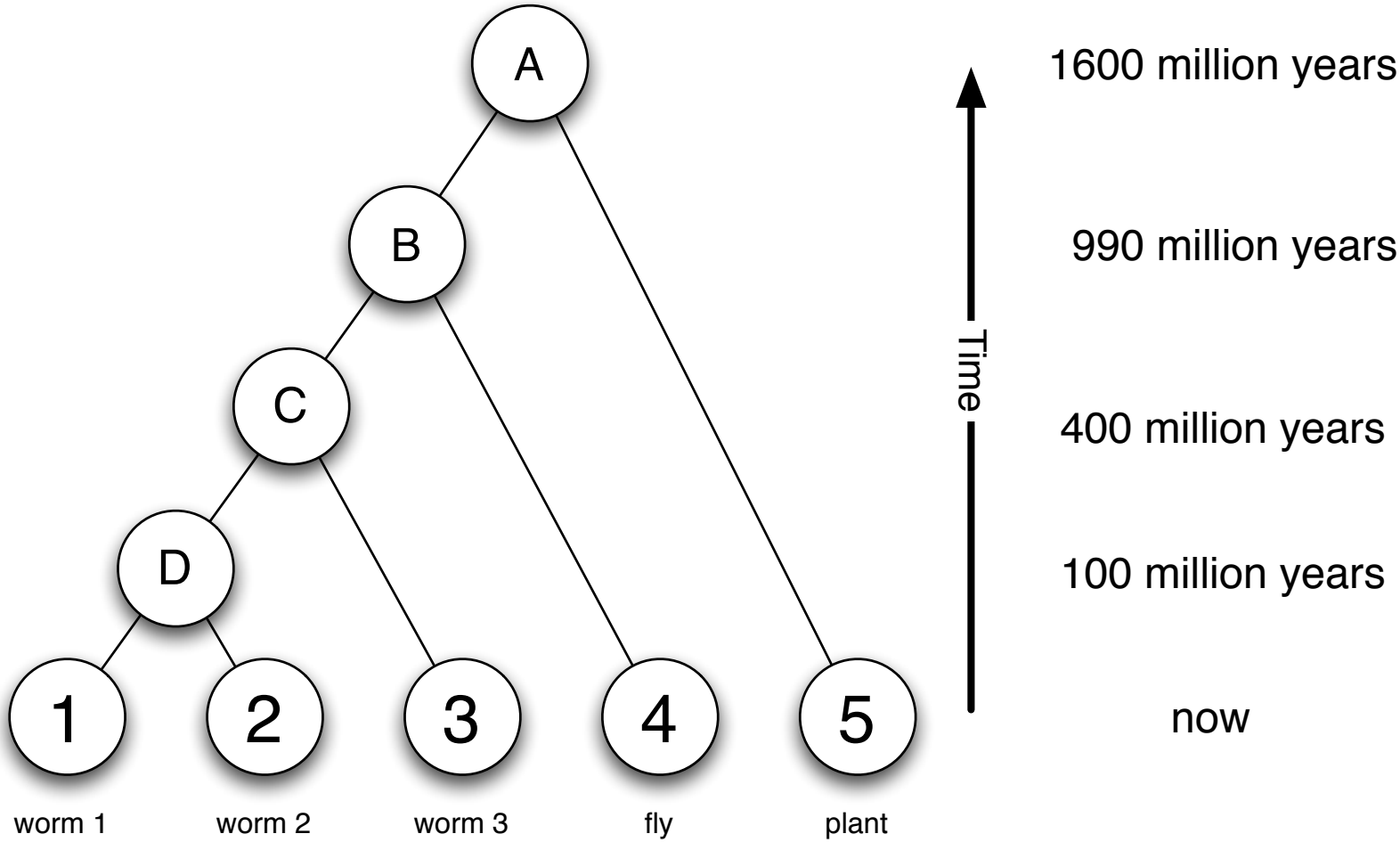
From Taxonomy to Γ



Idea: $\gamma_{i,j}$ should be inversely related to time to last common ancestor

Strategies: 1/years, **Hop-distance**, ...

From Taxonomy to Γ



Idea: $\gamma_{i,j}$ should be inversely related to time to last common ancestor

Strategies: 1/years, **Hop-distance**, ...

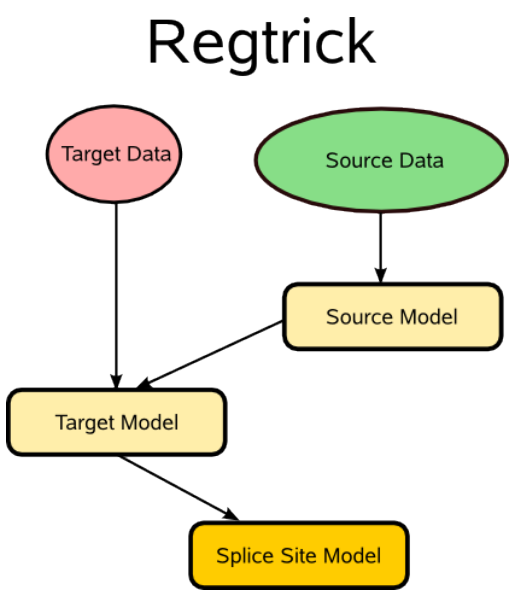
Hierarchical Top-Down Approach

Idea: Exploit taxonomy G in a top-down fashion

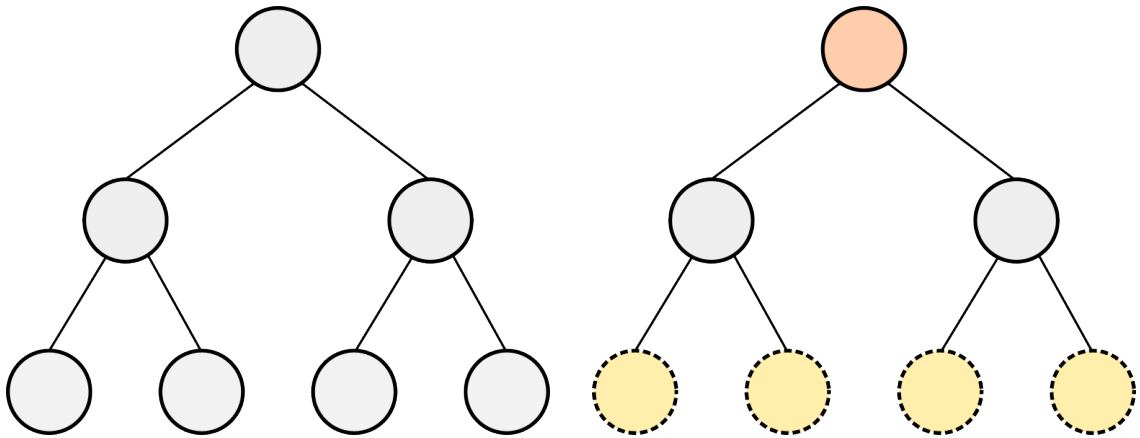
- Use taxonomy $\mathcal{T}$ in a top-down procedure
- Initialization: w_0 trained on union of all task datasets
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\boldsymbol{w}_i$ against parent predictor $\boldsymbol{w}_{parent}$:

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

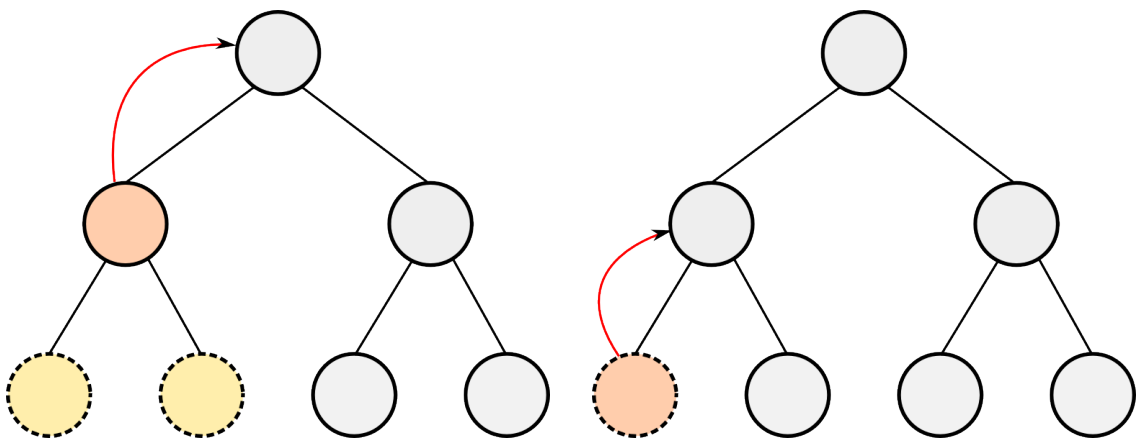
- Use leaf predictors for classification



Hierarchical Top-Down Approach: Illustration



(a) Given taxonomy (b) Top-level training



(c) Intermediate training (d) Taxon training

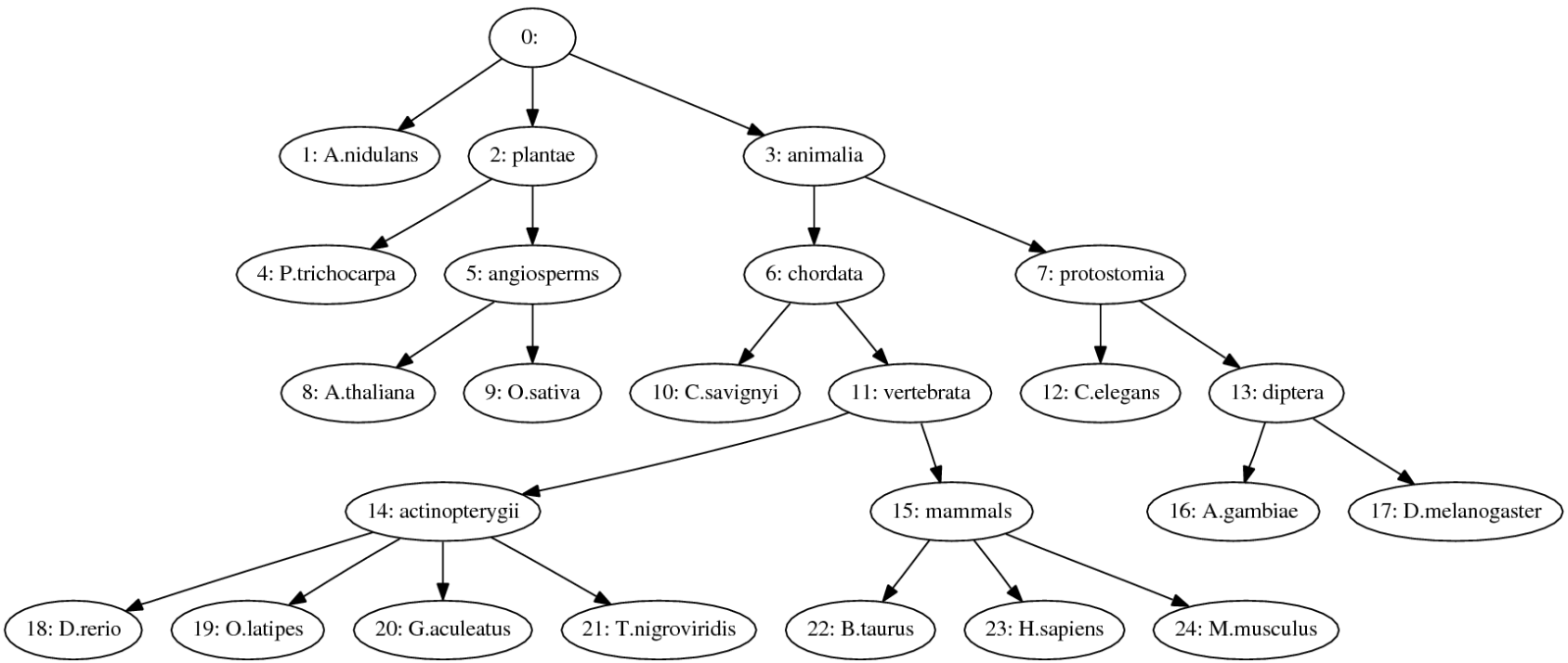
Application to Splicing Data

- Formulation as binary classification problem

≈ 150 nucleotides window around dimer

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

- Utilize 15 organisms related by taxonomy



- Restricted to at most 10.000 examples per organism

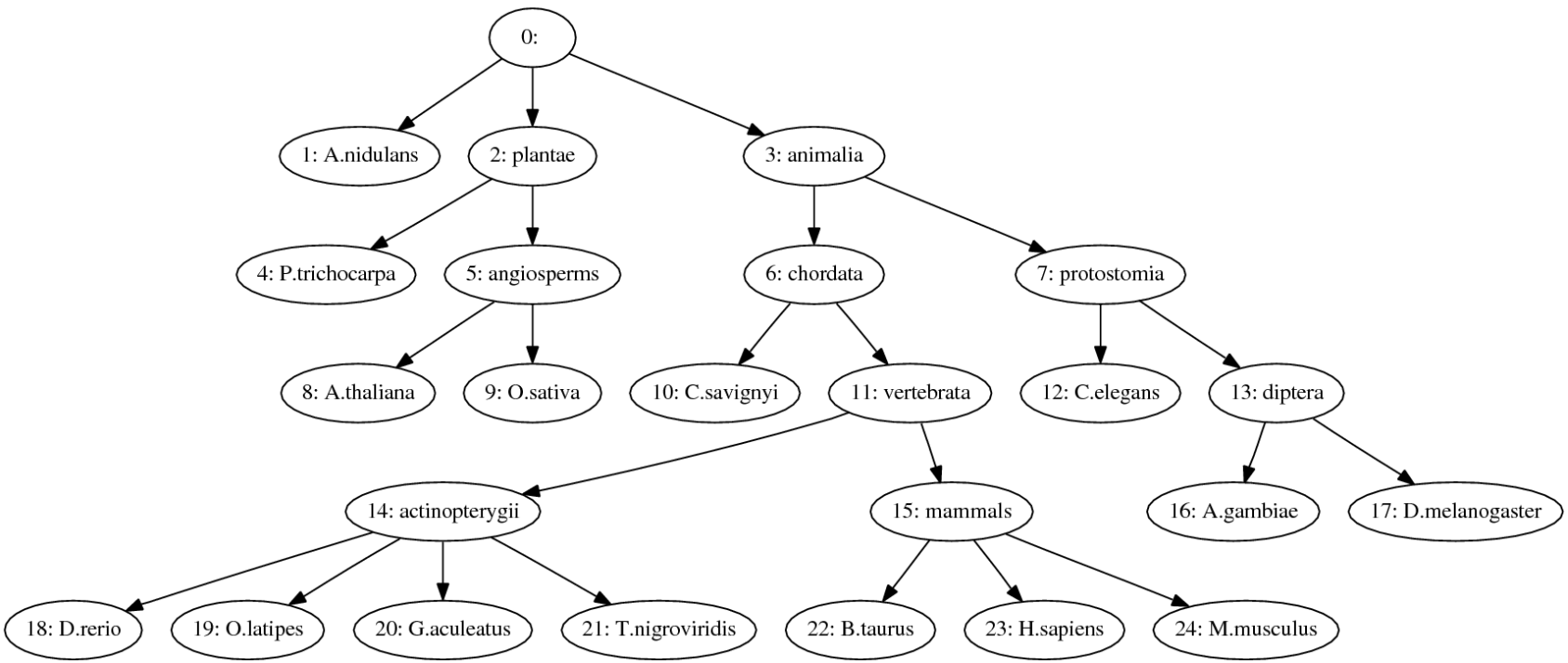
Application to Splicing Data

- Formulation as binary classification problem

≈ 150 nucleotides window around dimer

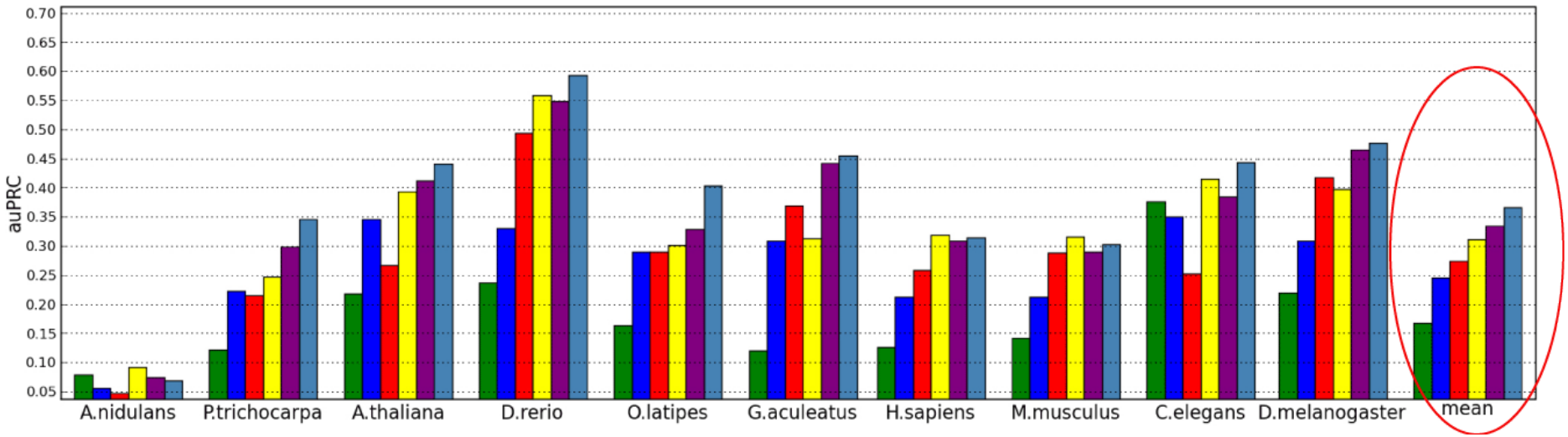
CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

- Utilize 15 organisms related by taxonomy



- Restricted to at most 10.000 examples per organism

Results: Splicing Data



Observations:

- Union > Plain → conservation
- Often: Union > Nearest
- MTL methods outperform baselines
- Best performer: Top-Down (& MT-Kernel)

Digestion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

$$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

$\Rightarrow$ Multiple-Kernel Multi-Task Learning!

Digestion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

$$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

$\Rightarrow$ **Multiple-Kernel Multi-Task Learning!**

Digestion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

$$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

$\Rightarrow$ **Multiple-Kernel Multi-Task Learning!**

Digestion

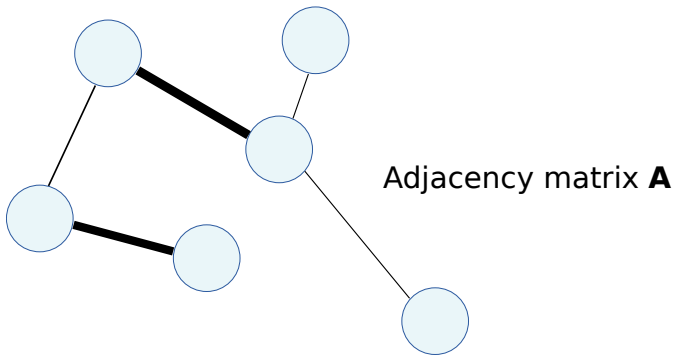
- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

    Adaptive Multi-Task approach?

    $\Rightarrow$ **Multiple-Kernel Multi-Task Learning!**

MTL by Graph Regularization



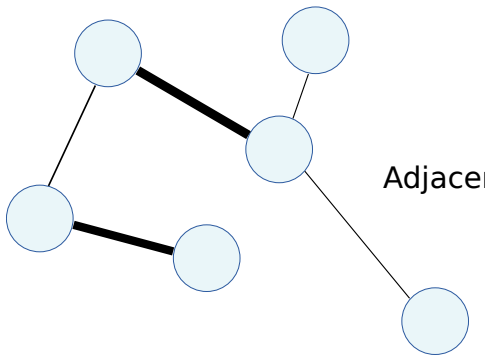
[Evgeniou et al. 2005]

- Regularizer expressed using **Graph Laplacian L**

$$\sum_{s=1}^{T}\sum_{t=1}^{T} \mathbf{A}_{st} \left|\left| w_s - w_t \right|\right|^2 = \sum_{s=1}^{T}\sum_{t=1}^{T} \mathbf{L}_{st} w_s^T w_t = Tr(W\mathbf{L}W^\top)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and $D_{ii} = \sum_j \mathbf{A}_{ij}$.

MTL by Graph Regularization



Adjacency matrix **A**

$$\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{A}_{st}\|\mathbf{w}_s-\mathbf{w}_t\|^2$$

mtl regularizer

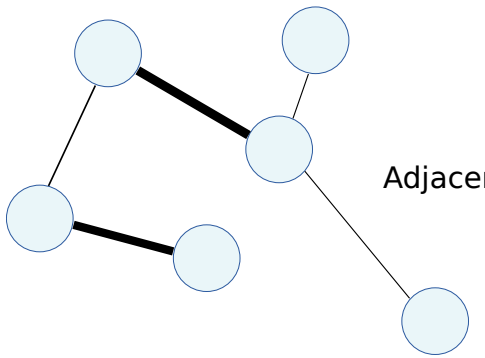
[Evgeniou et al. 2005]

- Regularizer expressed using **Graph Laplacian L**

$$\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{A}_{st}\,||w_s-w_t||^2=\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{L}_{st}w_s^Tw_t=Tr(W\mathbf{L}W^\top)$$

where $\mathbf{L}=\mathbf{D}-\mathbf{A}$ and $D_{ii}=\sum_j\mathbf{A}_{ij}$.

MTL by Graph Regularization



Adjacency matrix **A**

$$\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{A}_{st}\|\mathbf{w}_s-\mathbf{w}_t\|^2$$

mtl regularizer

[Evgeniou et al. 2005]

- Regularizer expressed using **Graph Laplacian L**

$$\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{A}_{st}\,||w_s-w_t||^2=\sum_{s=1}^{T}\sum_{t=1}^{T}\mathbf{L}_{st}w_s^Tw_t=Tr(W\mathbf{L}W^\top)$$

where $\mathbf{L}=\mathbf{D}-\mathbf{A}$ and $D_{ii}=\sum_j\mathbf{A}_{ij}$.

Beyond given task similarities

Question: How can we learn these similarities?

Multitask Multiple Kernel Learning

- **General formulation**:

$$\inf_{\substack{||\theta||_p \leq 1 \\ \boldsymbol{W} \in H}} \quad \frac{1}{2} \sum_{m=1}^{M} \theta_m{}^{-1} Tr(\boldsymbol{W}_m Q_m \boldsymbol{W}_m^T) \; + \; C \sum_{i=1}^{N} l\left(y_i \sum_{m=1}^{M} \left\langle \boldsymbol{w}_{m\tau(i)}, \phi_m(x_i) \right\rangle \right)$$

- **Key properties**
  - We learn a weighted sum of M regularizers
  - Allows to parameterize the task similarity measure
  - Learning the weights is **convex**
- **Key contributions**
  - Subsume existing methods and derive new formulations
  - Estiablish primal dual relationship using Fenchel duality
  - Derive dual-coordinate descend solver

[Widmer et al. 2010, 2013, 2014 i.P.]

Multitask Multiple Kernel Learning

- **General formulation**:

$$\inf_{\substack{||\theta||_p \leq 1 \\ \boldsymbol{W} \in H}} \quad \frac{1}{2} \sum_{m=1}^{M} \theta_m{}^{-1} Tr(\boldsymbol{W}_m Q_m \boldsymbol{W}_m^T) \; + \; C \sum_{i=1}^{N} l\left(y_i \sum_{m=1}^{M} \left\langle \boldsymbol{w}_{m\tau(i)}, \phi_m(x_i) \right\rangle \right)$$

- **Key properties**
  - We learn a weighted sum of M regularizers
  - Allows to parameterize the task similarity measure
  - Learning the weights is **convex**
- **Key contributions**
  - Subsume existing methods and derive new formulations
  - Estiablish primal dual relationship using Fenchel duality
  - Derive dual-coordinate descend solver

[Widmer et al. 2010, 2013, 2014 i.P.]

Recover existing formulations

$$\inf_{\substack{\mathbf{W}\in H \\ ||\theta||_p \leq 1}} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1}\ Tr(\mathbf{W}_m \mathbf{Q}_m \mathbf{W}_m^T) \ + \ C \sum_{i=1}^{N} l(y_i \sum_{m=1}^{M} \left\langle \mathbf{w}_{m\tau(i)}, \phi_m(\mathbf{x}_i) \right\rangle)$$

Existing methods are recovered by choosing Q, M, T, and ϕ.

- Multiple Kernel Learning: $T = 1$, different ϕ_m [Kloft et al., 2011]
- Standard single-task: $M = 1, T = 1$ [Boser et al., 1992]
- Frustratingly easy: $M = 1, T = 2$, special Q [Daume III, 2007]
- Graph-regularized MTL: $M = 1, Q = L$ [Evgeniou et al., 2005]

Recover existing formulations

$$\inf_{\substack{\mathbf{w}\in H \\ ||\theta||_p \leq 1}} \quad \frac{1}{2}\sum_{m=1}^{M} \theta_m^{-1}\ Tr(\ \mathbf{w}_m\ I_m\ \mathbf{w}_m^T)\ +\ C\sum_{i=1}^{N} l(y_i \sum_{m=1}^{M} \left\langle \mathbf{w}_{m\,\tau(i)}, \phi_m(\mathbf{x}_i)\right\rangle)$$

Existing methods are recovered by choosing Q, M, T, and ϕ.

- Multiple Kernel Learning: $T = 1$, different ϕ_m [Kloft et al., 2011]
- Standard single-task: $M = 1, T = 1$ [Boser et al., 1992]
- Frustratingly easy: $M = 1, T = 2$, special Q [Daume III, 2007]
- Graph-regularized MTL: $M = 1, Q = L$ [Evgeniou et al., 2005]

Recover existing formulations

$$\inf_{\substack{\mathbf{w}\in H \\ ||\theta||^p \leq 1}} \quad \frac{1}{2} \sum_{m=1}^{M} \theta_m^{-1} \, Tr(\, \mathbf{w}_m \, I_m \, \mathbf{w}_m^T \,) + C \sum_{i=1}^{N} l(y_i \sum_{m=1}^{M} \left\langle \mathbf{w}_{m\tau(i)}, \phi_m(\mathbf{x}_i) \right\rangle)$$

Existing methods are recovered by choosing Q, M, T, and ϕ.

- Multiple Kernel Learning: $T = 1$, different ϕ_m [Kloft et al., 2011]
- Standard single-task: $M = 1, T = 1$ [Boser et al., 1992]
- Frustratingly easy: $M = 1, T = 2$, special Q [Daume III, 2007]
- Graph-regularized MTL: $M = 1, Q = L$ [Evgeniou et al., 2005]

Recover existing formulations

$$\inf_{\substack{\mathbf{W}\in H \\ ||\theta||^p \leq 1}} \frac{1}{2}\sum_{m=1}^{M}\theta_m^{-1}\, Tr(\mathbf{W}\begin{pmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{pmatrix}\mathbf{W}^T) + C\sum_{i=1}^{N} l(y_i, \sum_{m=1}^{M}\left\langle \mathbf{w}_{m\tau(i)}, \phi_m(\mathbf{x}_i)\right\rangle)$$

Existing methods are recovered by choosing Q, M, T, and ϕ.

- Multiple Kernel Learning: $T = 1$, different ϕ_m [Kloft et al., 2011]
- Standard single-task: $M = 1, T = 1$ [Boser et al., 1992]
- Frustratingly easy: $M = 1, T = 2$, special Q [Daume III, 2007]
- Graph-regularized MTL: $M = 1, Q = L$ [Evgeniou et al., 2005]

Recover existing formulations

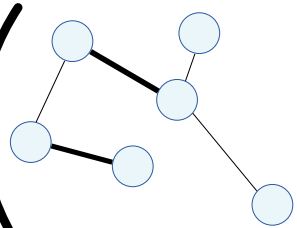
$$\inf_{\substack{\mathbf{W}\in H \\ ||\theta||^p\leq 1}} \quad \frac{1}{2}\sum_{m=1}^{M}\theta_m^{-1}\, Tr(\mathbf{W}_m \mathbf{L} \mathbf{W}_m^T) + C\sum_{i=1}^{N} l(y_i, \sum_{m=1}^{M}\langle \mathbf{w}_{m\tau(i)}, \phi_m(\mathbf{x}_i)\rangle)$$

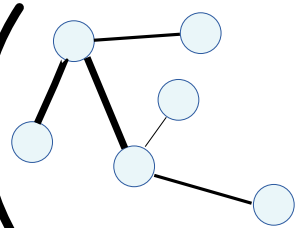
Existing methods are recovered by choosing Q, M, T, and ϕ.

- Multiple Kernel Learning: $T = 1$, different ϕ_m [Kloft et al., 2011]
- Standard single-task: $M = 1, T = 1$ [Boser et al., 1992]
- Frustratingly easy: $M = 1, T = 2$, special Q [Daume III, 2007]
- Graph-regularized MTL: $M = 1, Q = L$ [Evgeniou et al., 2005]

Novel formulations

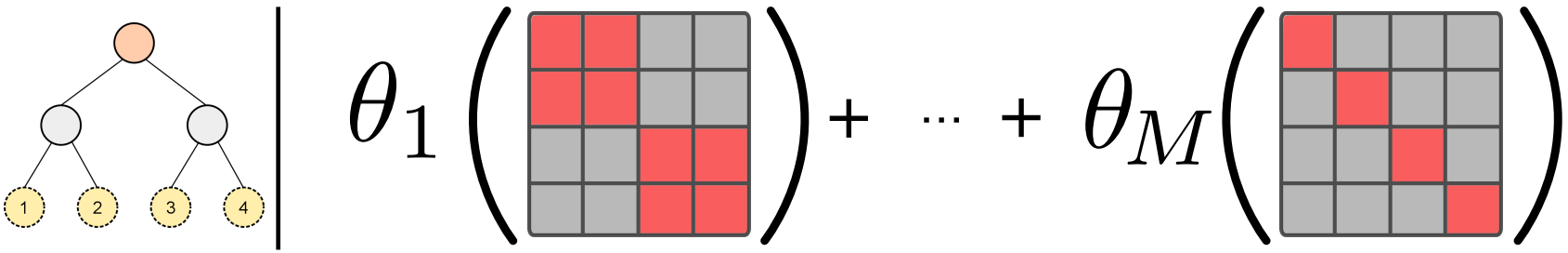
$$\begin{matrix} G_1 \\ \dots \\ G_M \end{matrix} \Bigg| \quad \theta_1 \left(\quad \right) + \quad \cdots \quad + \quad \theta_M \left(\quad \right)$$





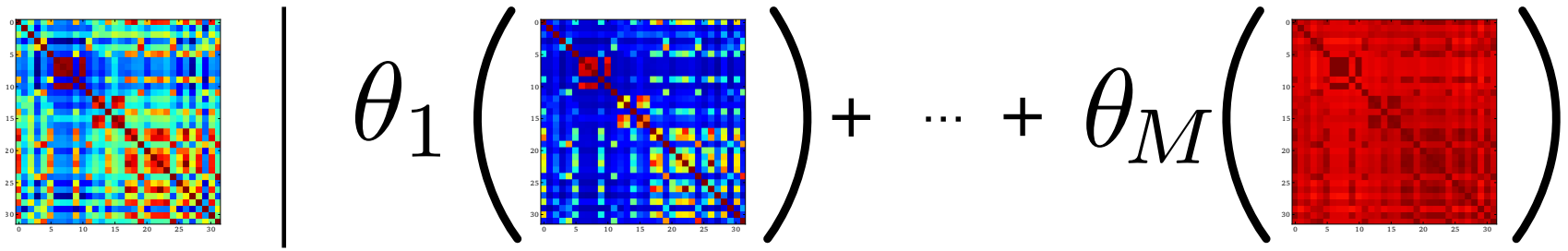
- Graphs from different sources
- Hierarchical decomposition
- Transformations at different length scales, e.g. $\mathbf{Q}_{st}^{(m)} = exp(A_{st}/\sigma_m)$
- Powerset of tasks

Novel formulations



- Graphs from different sources
- Hierarchical decomposition
- Transformations at different length scales, e.g. $\mathbf{Q}_{st}^{(m)} = exp(A_{st}/\sigma_m)$
- Powerset of tasks

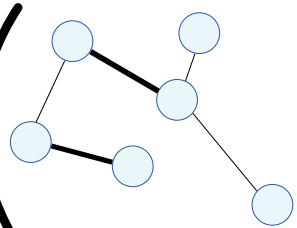
Novel formulations

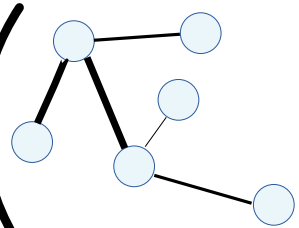


- Graphs from different sources
- Hierarchical decomposition
- Transformations at different length scales, e.g. $\mathbf{Q}_{st}^{(m)} = exp(A_{st}/\sigma_m)$
- Powerset of tasks

Novel formulations

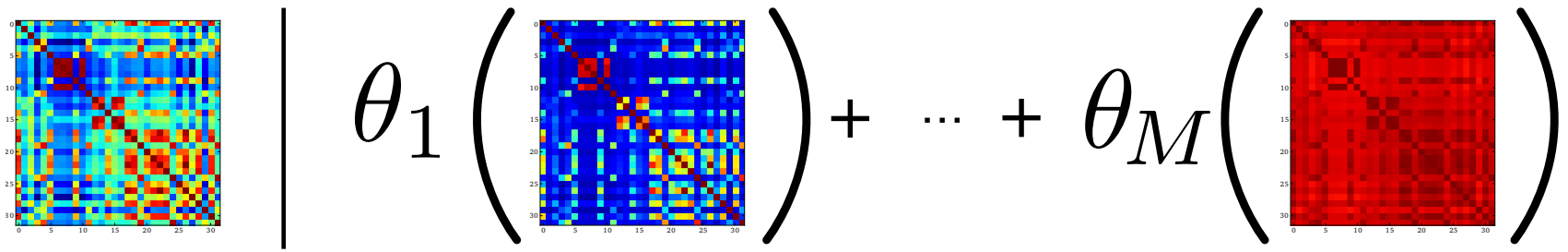
$$\begin{matrix} G_1 \\ \dots \\ G_M \end{matrix} \Bigg| \; \theta_1 \Big(\quad \Big) + \; \cdots \; + \; \theta_M \Big(\quad \Big)$$





- Graphs from different sources
- Hierarchical decomposition
- Transformations at different length scales, e.g. $\mathbf{Q}_{st}^{(m)} = exp(A_{st}/\sigma_m)$
- Powerset of tasks

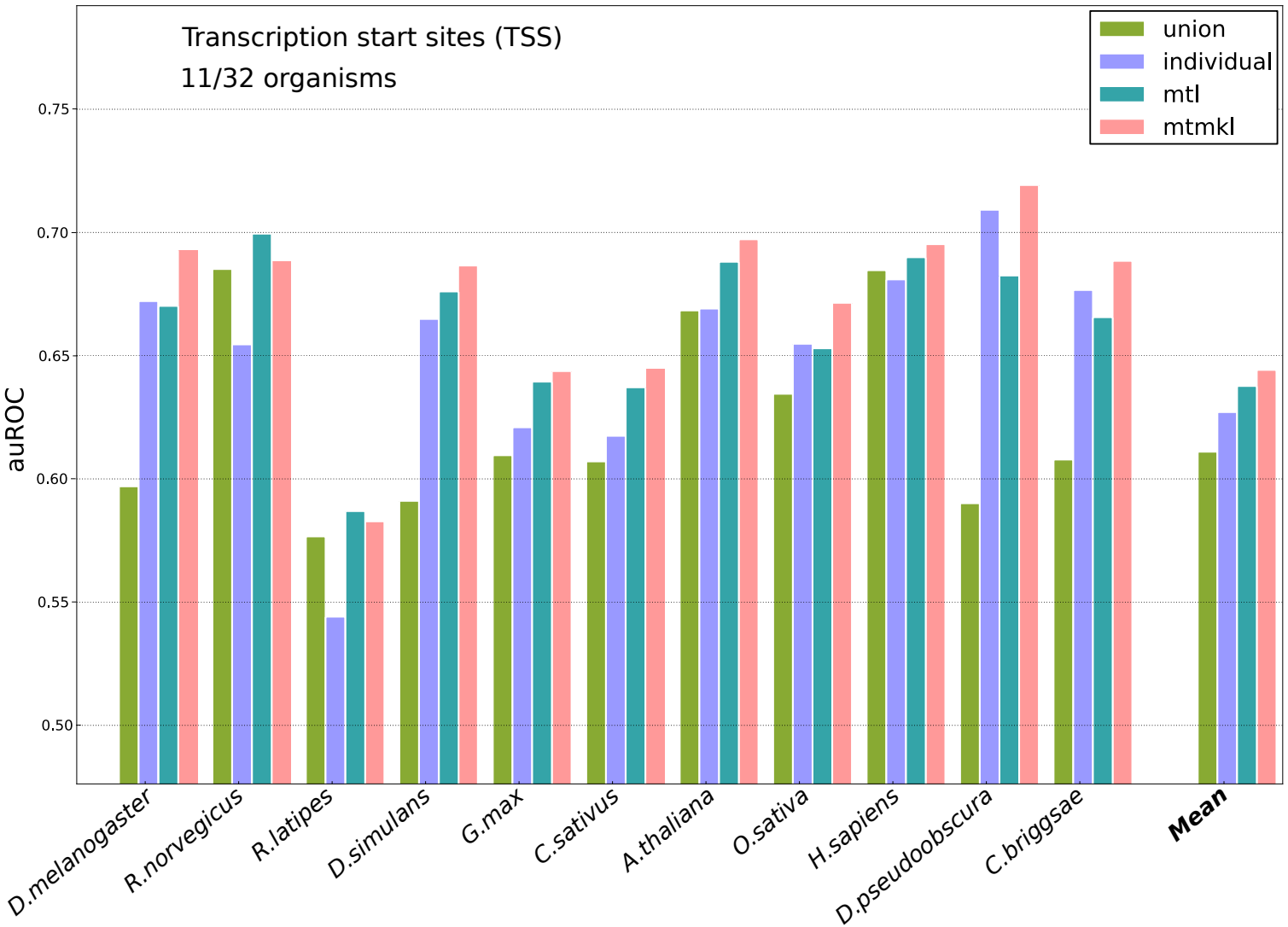
Illustration on Detecting Gene Starts



- Initial task-similarity **A** from conserved sequences
- Different length scales to adapt trade-off:
  $\mathbf{Q}_{st}^{(m)} = exp(A_{st}/\sigma_m)$

[Widmer et al. 2014 i.P.]

Illustration on Detecting Gene Starts



An efficient solver

How can we solve this efficiently?

Dual-coordinate descend

- We bring the state-of-the-art from SVMs to MTL: LibLinear
- Update one example weight α_i at a time
- Keep primal variable $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$ in memory
- Use this to speed up computation:

$$\nabla_i f(\alpha) = (\bar{Q}\alpha)_i - 1 = \underbrace{\sum_{j=1}^{N} y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j - 1}_{O(N \cdot D)} = \underbrace{y_i \mathbf{w}^T \mathbf{x}_i - 1}_{O(D)} .$$

Dual-coordinate descend

- Establish primal-dual relationship using Fenchel duality

$$\inf_{\theta\in\Theta_p} \sup_{0\preceq\alpha\preceq C} \; -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \sum_{m=1}^{M}\theta_m \underbrace{Q^{(-1)}_{m\tau(i)\tau(j)}k_m(x_i,x_j)}_{\tilde{k}_m} \; + \; \sum_{i=1}^{N}\alpha_i,$$

- Representer theorem:

$$\boldsymbol{w}_{mt} = \theta_m \sum_{i=1}^{N} Q^{(-1)}_{m\tau(i)t}\alpha_i y_i \phi_m(x_i)\,.$$

- Derive update rule (set gradient to zero, solve for d, project):

$$d^\star = \max\left(-\alpha_i\,,\;\; \min\left(C-\alpha_i\,,\; \frac{1-y_i\sum_{m=1}^{M}\theta_m\left\langle \boldsymbol{w}_{m\tau(i)}, \phi_m(x_i)\right\rangle}{\sum_{m=1}^{M}\theta_m\tilde{k}_m(x_i,x_i)}\right)\right)$$

- Caveat: We need to update all $\boldsymbol{w}^{\text{new}}_{mt} := \boldsymbol{w}^{\text{old}}_{mt} + d\theta_m Q^{(-1)}_{m\tau(i)t} y_i \phi_m(x_i)$

[Widmer et al., 2012]

Computational Experiments

	dim	examples	tasks	time(s) mtk	time(s) dcd
Gauss2D	2	$1 \cdot 10^5$	2	$2 \cdot 10^3$	$4 \cdot 10^0$
Breast Cancer	44	474	3	$1 \cdot 10^1$	$2 \cdot 10^0$
MNIST-MTL	784	$9.0 \cdot 10^3$	3	$3 \cdot 10^2$	$1 \cdot 10^2$
Land Mine	9	$1.5 \cdot 10^4$	29	$8 \cdot 10^1$	$2 \cdot 10^2$
Splicing	$6 \cdot 10^6$	$6.4 \cdot 10^6$	4	$> 1 \cdot 10^6$	$9 \cdot 10^4$

- Different data set properties
- Compare to SVMLight-based solver on $\tilde{k}$
- Fast convergence for moderate number of tasks

[Widmer et al., 2012]

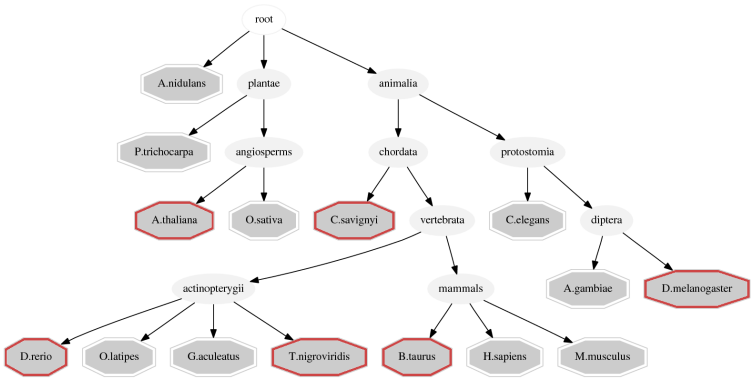
Computational Experiments

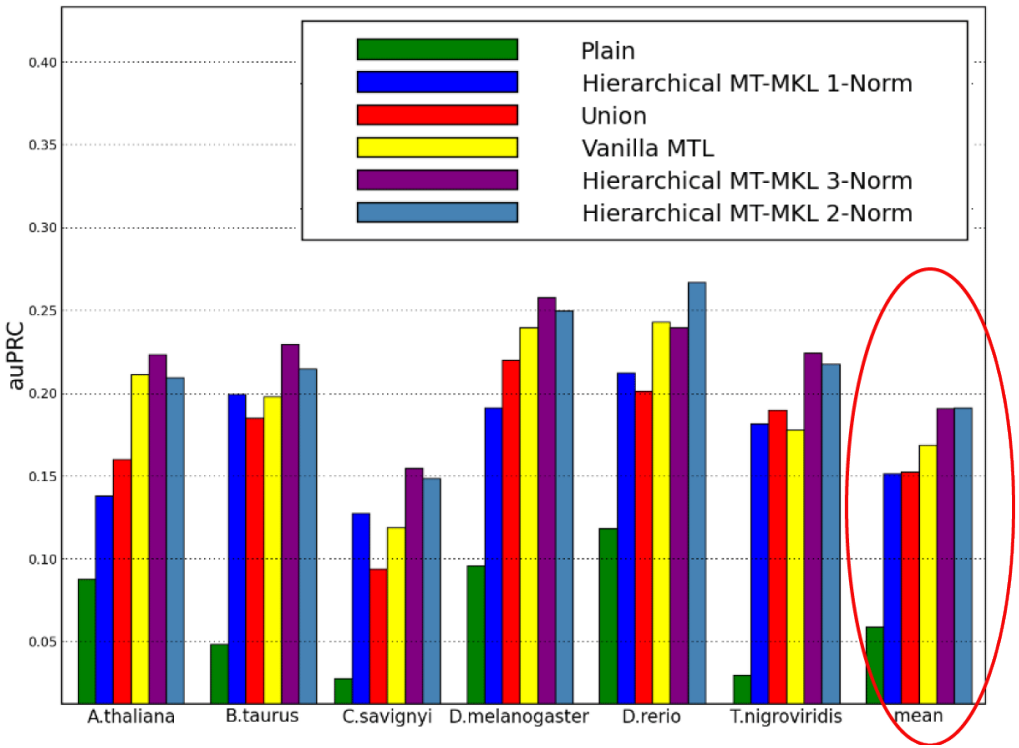
	dim	examples	tasks	time(s) mtk	time(s) dcd
Gauss2D	2	$1 \cdot 10^5$	2	$2 \cdot 10^3$	$4 \cdot 10^0$
Breast Cancer	44	474	3	$1 \cdot 10^1$	$2 \cdot 10^0$
MNIST-MTL	784	$9.0 \cdot 10^3$	3	$3 \cdot 10^2$	$1 \cdot 10^2$
Land Mine	9	$1.5 \cdot 10^4$	29	$8 \cdot 10^1$	$2 \cdot 10^2$
Splicing	$6 \cdot 10^6$	$6.4 \cdot 10^6$	4	$> 1 \cdot 10^6$	$9 \cdot 10^4$

- Different data set properties
- Compare to SVMLight-based solver on $\tilde{k}$
- Fast convergence for moderate number of tasks

[Widmer et al., 2012]

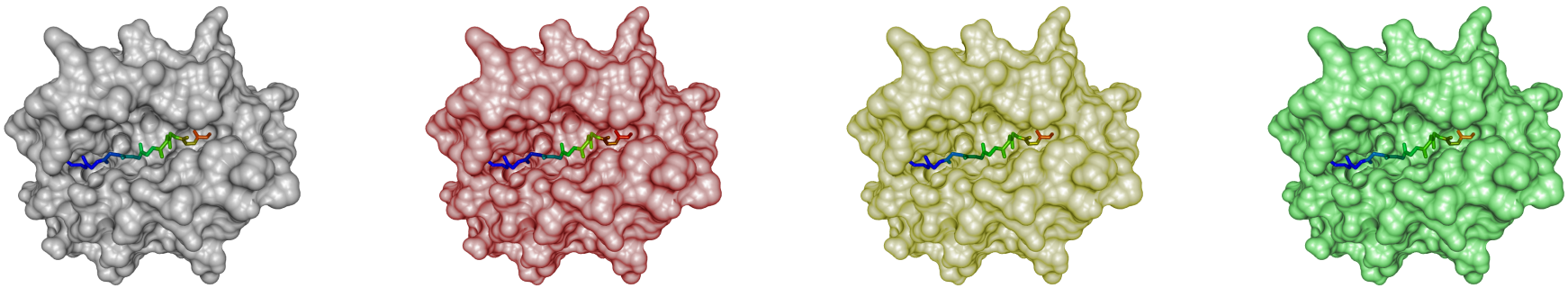
Experiments (a): Splice-site recognition





- Taxonomy is used to define collection of meta-tasks $\mathcal{I}$
- Baselines: Plain, Union, Vanilla MTL
- Best performance for norm $q = 2, 3$

Experiments (b): MHC-I binding prediction

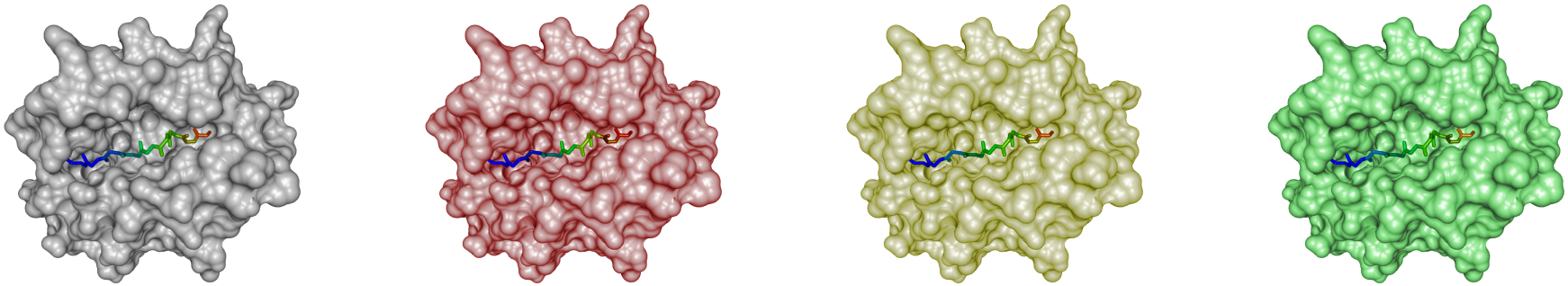


Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	**69.9%**

- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

[Widmer et al., 2010b]

Experiments (b): MHC-I binding prediction



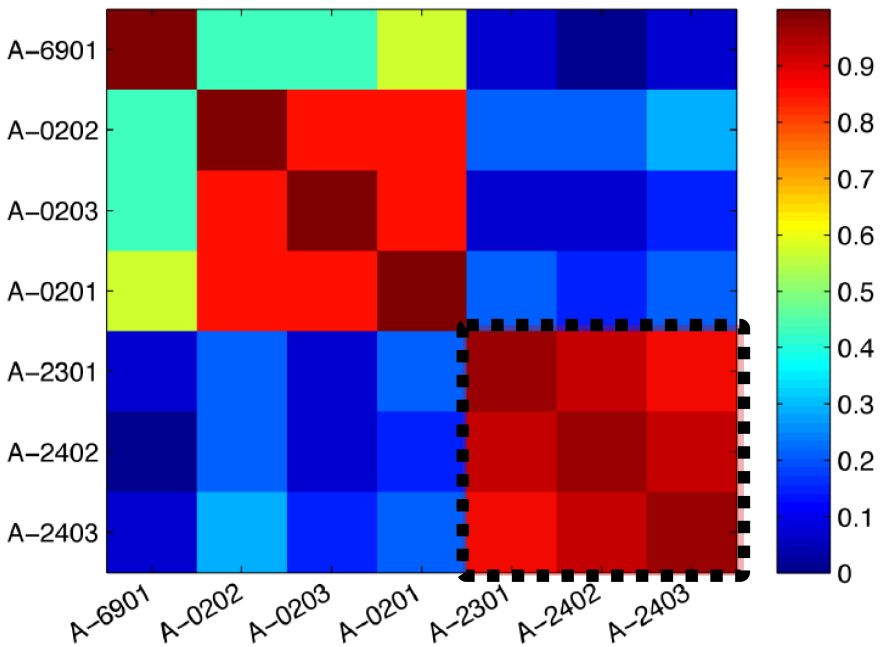
Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	**69.9%**

- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

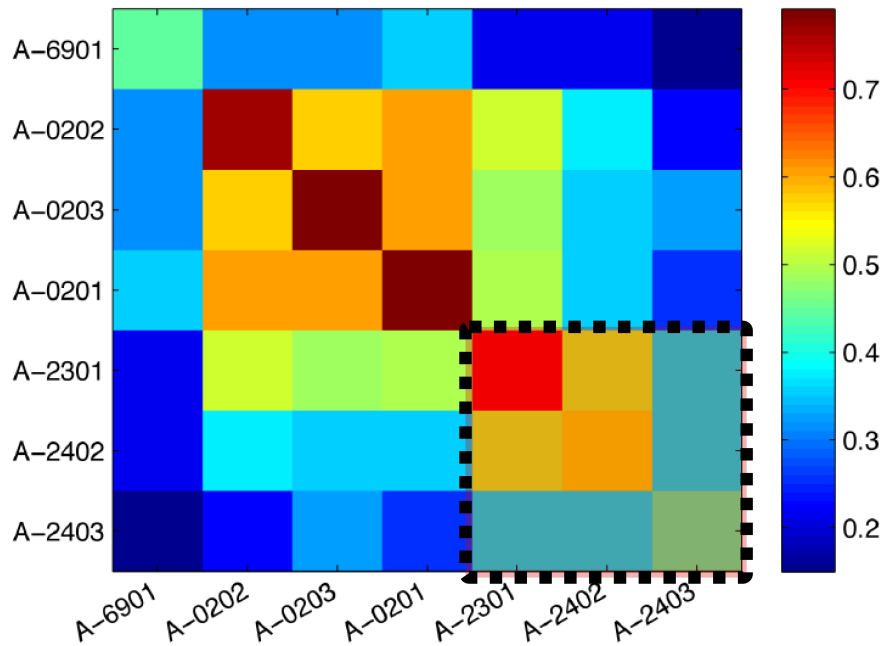
[Widmer et al., 2010b]

Experiments (b): MHC-I binding prediction

Learned weights can also be used for interpretation purposes:



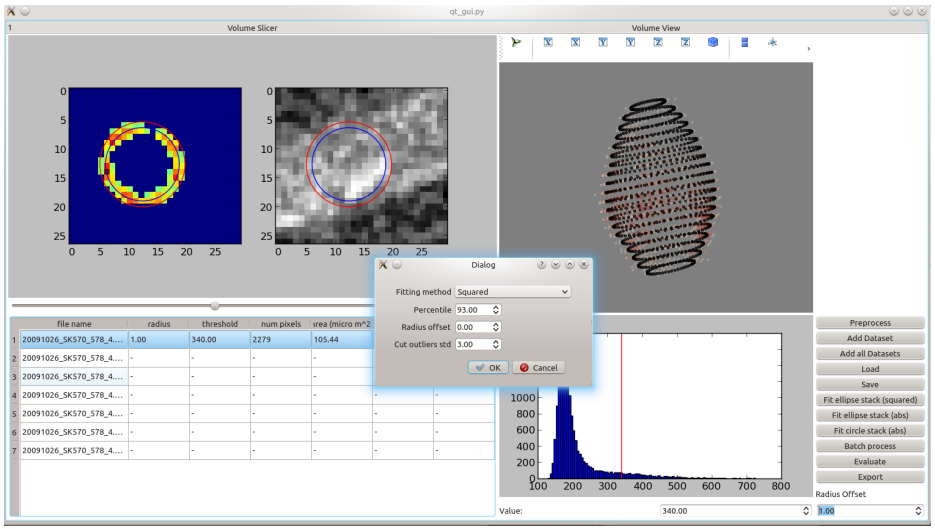
(a) Sequence similarity between alleles



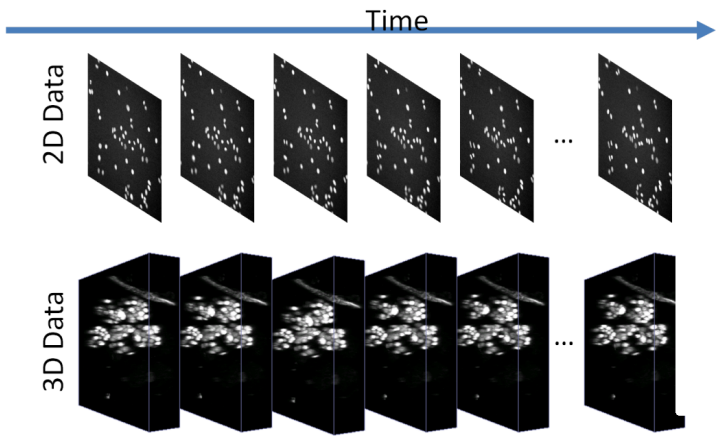
(b) Task similarity computed from weights

- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

Application to Imaging



Nucleus Fitting



2D to 3D

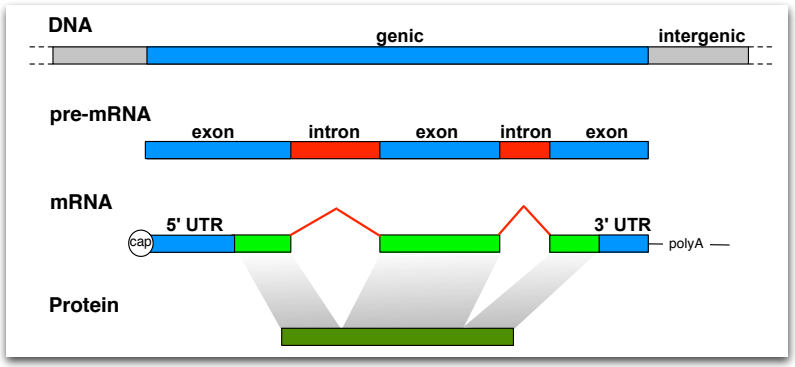
- Nucleus fitting: Couple layers [Widmer et al., 2014]
- 2D helps 3D: Couple 2D and 3D [Lou et al., 2012]

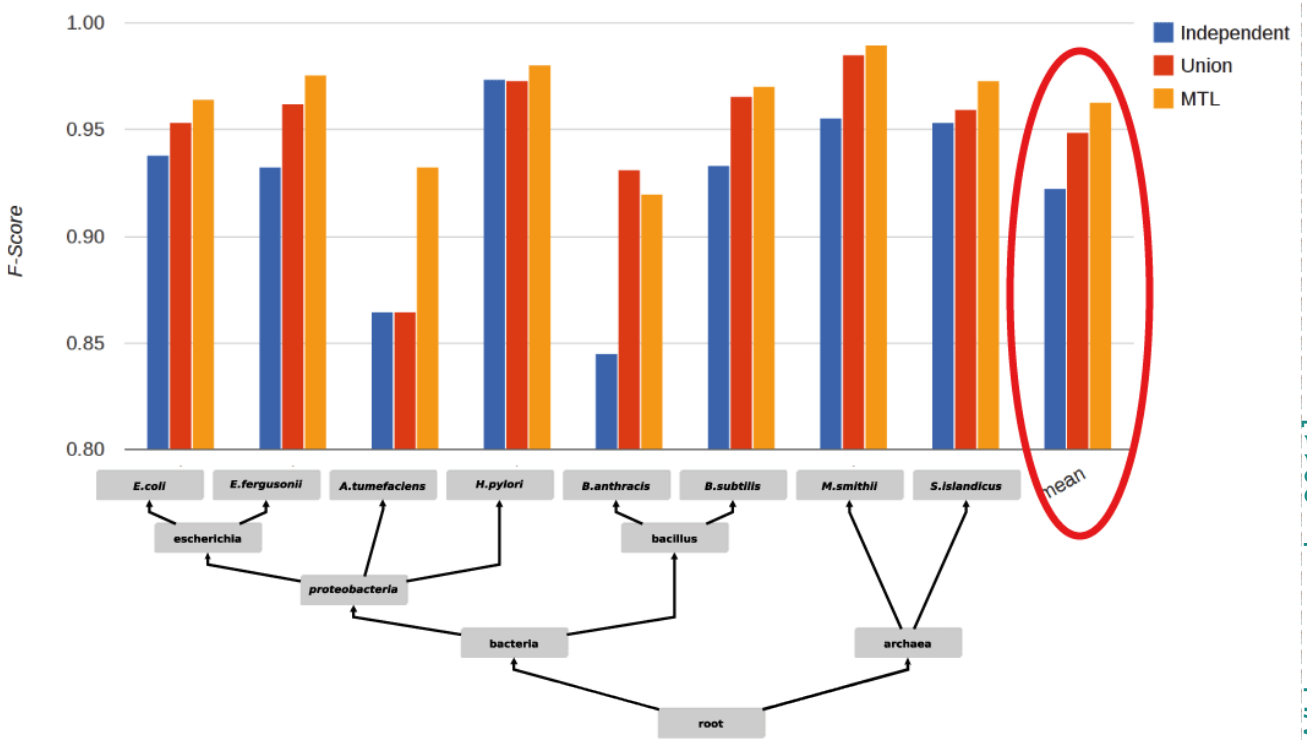
Extension to Structured Output Learning

Strategies only work on regularizer

Also work for structured output learning

- Hidden Markov SVMs
- Gene prediction (prokaryotes)





[Widmer et al., 2011]

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: www.raetschlab.org/suppl/transfer-learning

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: www.raetschlab.org/suppl/transfer-learning

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: www.raetschlab.org/suppl/transfer-learning

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: www.raetschlab.org/suppl/transfer-learning

Acknowledgements



Christian Widmer

(MSKCC & TU Berlin, Microsoft)



Marius Kloft

(MSKCC, NYU, Humboldt U)



Vipin Sreedharan

(MSKCC)

Involved earlier

- Gabriele Schweikert
- Nico Görnitz
- Nora Toussaint
- Jose Leiva
- Yasemin Altun
- Bernhard Schölkopf

Funding by German Research Foundation, Max Planck Society & MSKCC

Thank you for your attention!

Acknowledgements



Christian Widmer

(MSKCC & TU Berlin, Microsoft)



Marius Kloft

(MSKCC, NYU, Humboldt U)



Vipin Sreedharan

(MSKCC)

Involved earlier

- Gabriele Schweikert
- Nico Görnitz
- Nora Toussaint
- Jose Leiva
- Yasemin Altun
- Bernhard Schölkopf

Funding by German Research Foundation, Max Planck Society & MSKCC

Thank you for your attention!



Many algorithms implemented in Shogun toolbox (GPL, $\geq$ 1000 users)

Soon...

Journal of Machine Learning Research 15 (2014) XXX-XXX Submitted ??/14; Published ??/14

Framework for Multi-task Multiple Kernel Learning and Applications in Computational Biology

Christian Widmer* CWIDMER@CBIO.MSKCC.ORG
Computational Biology Center
Memorial Sloan-Kettering Cancer Center
415 E 68th street
New York, NY 10065, USA

Marius Kloft† MKLOFT@CS.NYU.EDU
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012, USA

Vipin T. Sreedharan VIPIN@CBIO.MSKCC.ORG
Computational Biology Center
Memorial Sloan-Kettering Cancer Center
415 E 68th street
New York, NY 10065, USA

Gunnar Rätsch RATSCHG@MSKCC.ORG
Computational Biology Center
Memorial Sloan-Kettering Cancer Center
415 E 68th street
New York, NY 10065, USA

References

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Haussler, D., editor, Proc. Annual Conf. Computational Learning Theory, pages 144–152, Pittsburgh, PA. ACM Press.

Caruana, R. (1997). Multitask learning. Machine Learning, 28(1):41–75.

Daume III, H. (2007). Frustratingly easy domain adaptation. In Conference of the Association for Computational Linguistics (ACL), Prague, Czech Republic.

Evgeniou, T., Micchelli, C., and Pontil, M. (2005). Learning multiple tasks with kernel methods. Journal of Machine Learning Research, 6:615–637.

Lou, X., Widmer, C., Hadjantonakis, K., and Rätsch, G. (2012). Structured domain adaptation across imaging modality how 2d data helps 3d inference. In Proc. Workshop on Machine Learning in Computational Biology.

M. Kloft, U. Brefeld, S. S. and Zien, A. (2011). Lp-norm multiple kernel learning. Journal of Machine Learning Research, (12):953–997.

Schweikert, G., Widmer, C., Schölkopf, B., and Rätsch, G. (2008). An empirical analysis of domain adaptation algorithms. In Advances in Neural Information Processing System, NIPS, volume 22, Vancouver, B.C.

Widmer, C. (2014). Regularization-based Multitask Learning. PhD thesis, Berlin Institute of Technology.

Widmer, C., Görnitz, N., Zeller, G., and Rätsch, G. (2011). Hierarchical multitask structured output learning for large-scale sequence segmentation. submitted.

Widmer, C., Heinrich, S., Drewe, P., Lou, X., Umrania, S., and Rätsch, G. (2014). Graph-regularized 3d shape reconstruction from highly anisotropic and noisy images. SIViP.

Widmer, C., Kloft, M., Görnitz, N., and Rätsch, G. (2012). Efficient training of graph-regularized multitask svms. In Proc. ECML.

Widmer, C., Leiva, J., Altun, Y., and Rtsch, G. (2010a). Leveraging sequence classification by taxonomy-based multitask learning. In Proc. RECOMB'10. accepted.

Widmer, C. and Rätsch, G. (2011). Transfer learning in computational biology. In Proc. ICML.

Widmer, C., Toussaint, N., Altun, Y., and Rätsch, G. (2010b). Inferring latent task structure for multi-task learning by multiple kernel learning. BMC Bioinformatics, 11(Suppl. 8):S5.