Transfer Learning in Computational Biology

Gunnar Rätsch



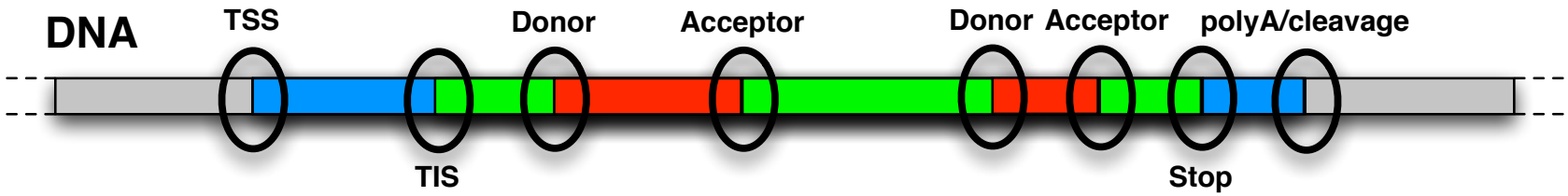
Tübingen, Germany

July 2, 2011

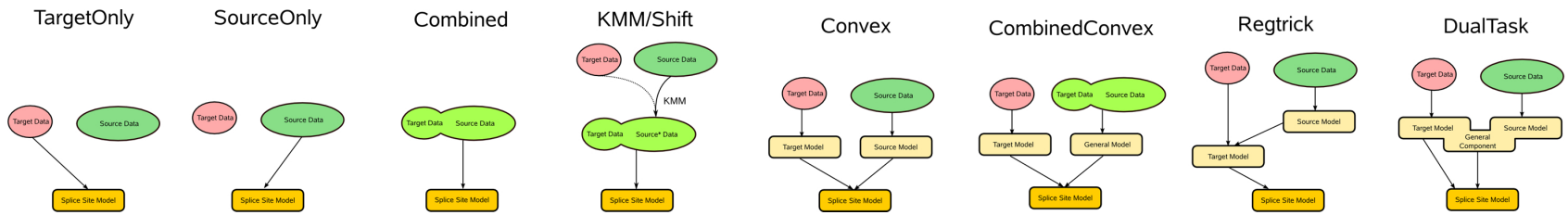
ICML Workshop on Unsupervised and Transfer Learning
Bellevue, WA

Roadmap

- Motivation from computational biology



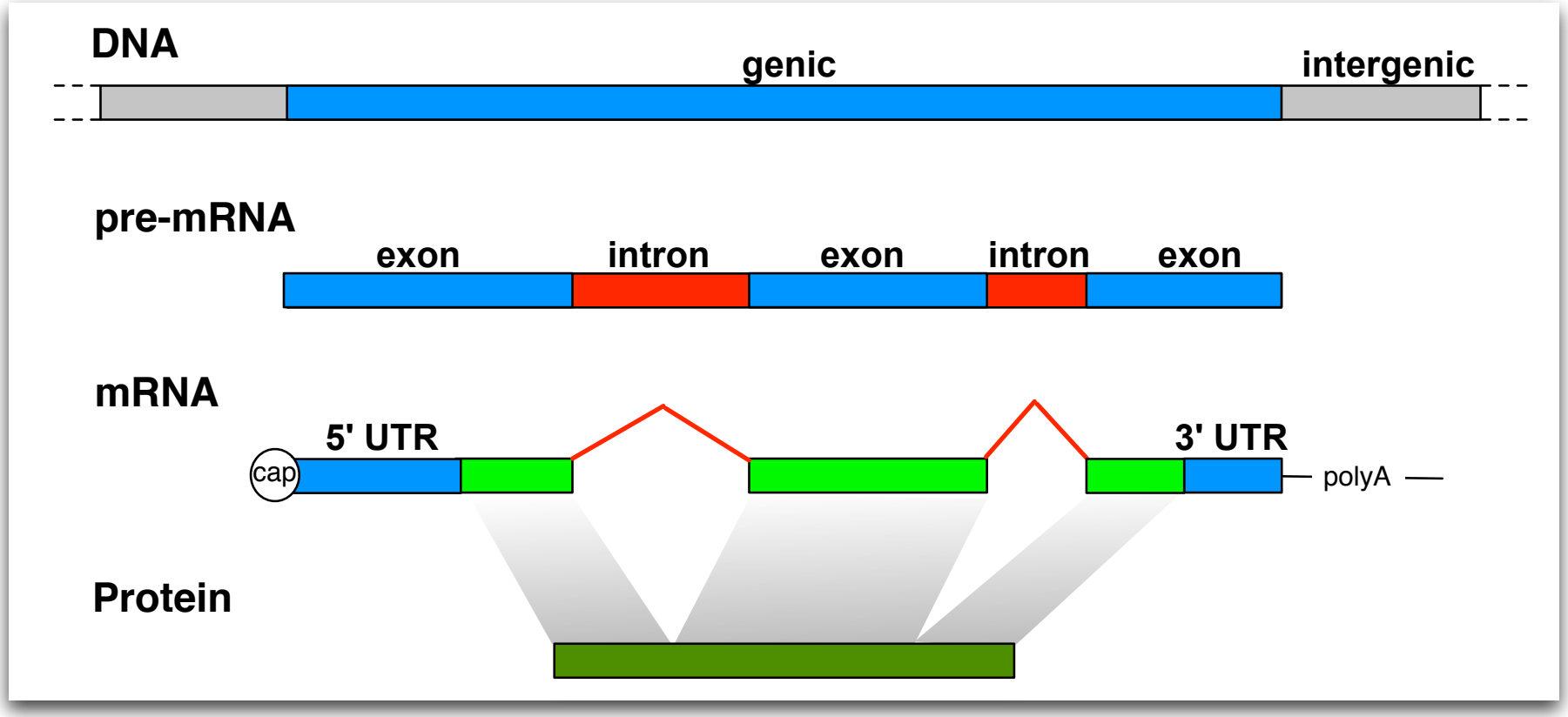
- Empirical comparison of domain adaptation algorithms



- Algorithms for hierarchical multi-task learning
- Algorithms for learning task relations
- Discussion & Conclusion

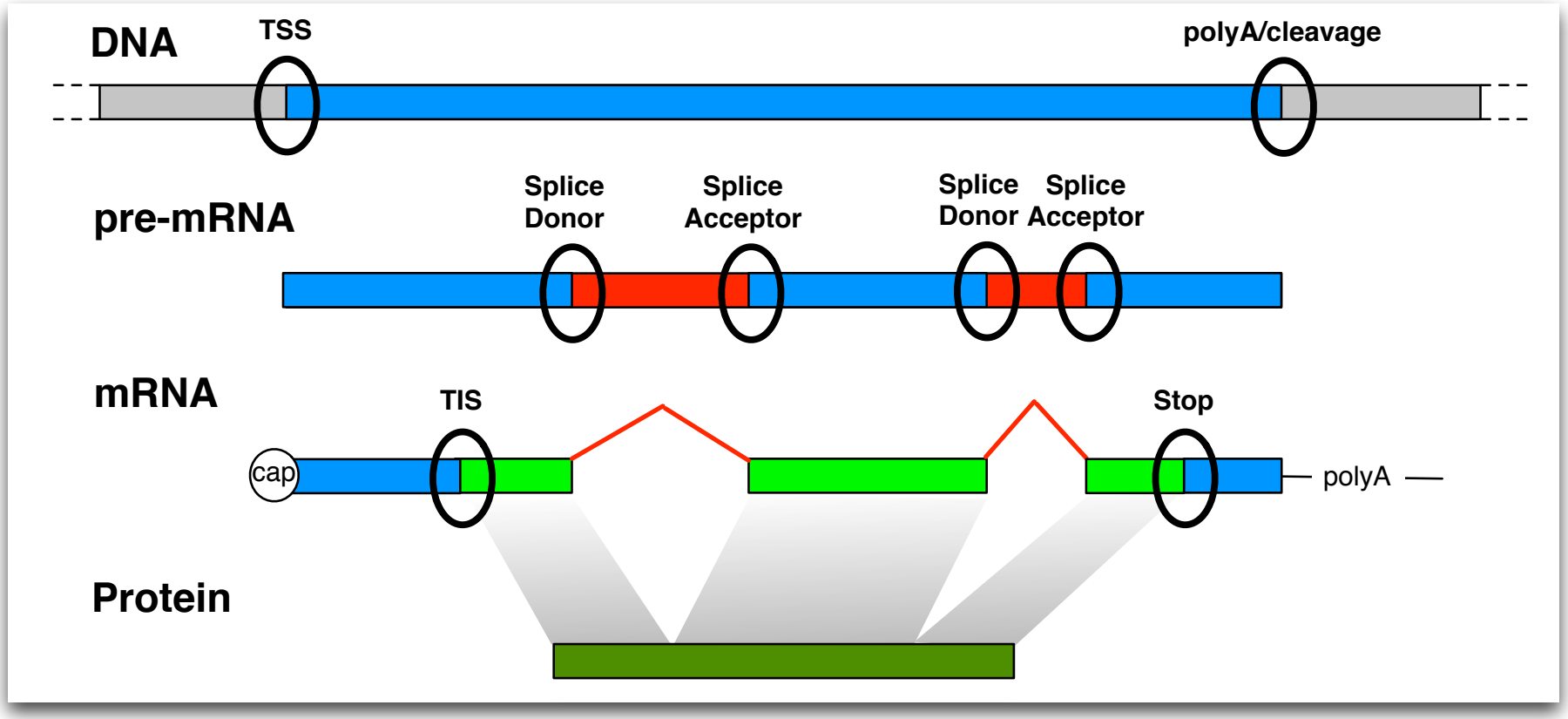


Computational Gene Finding



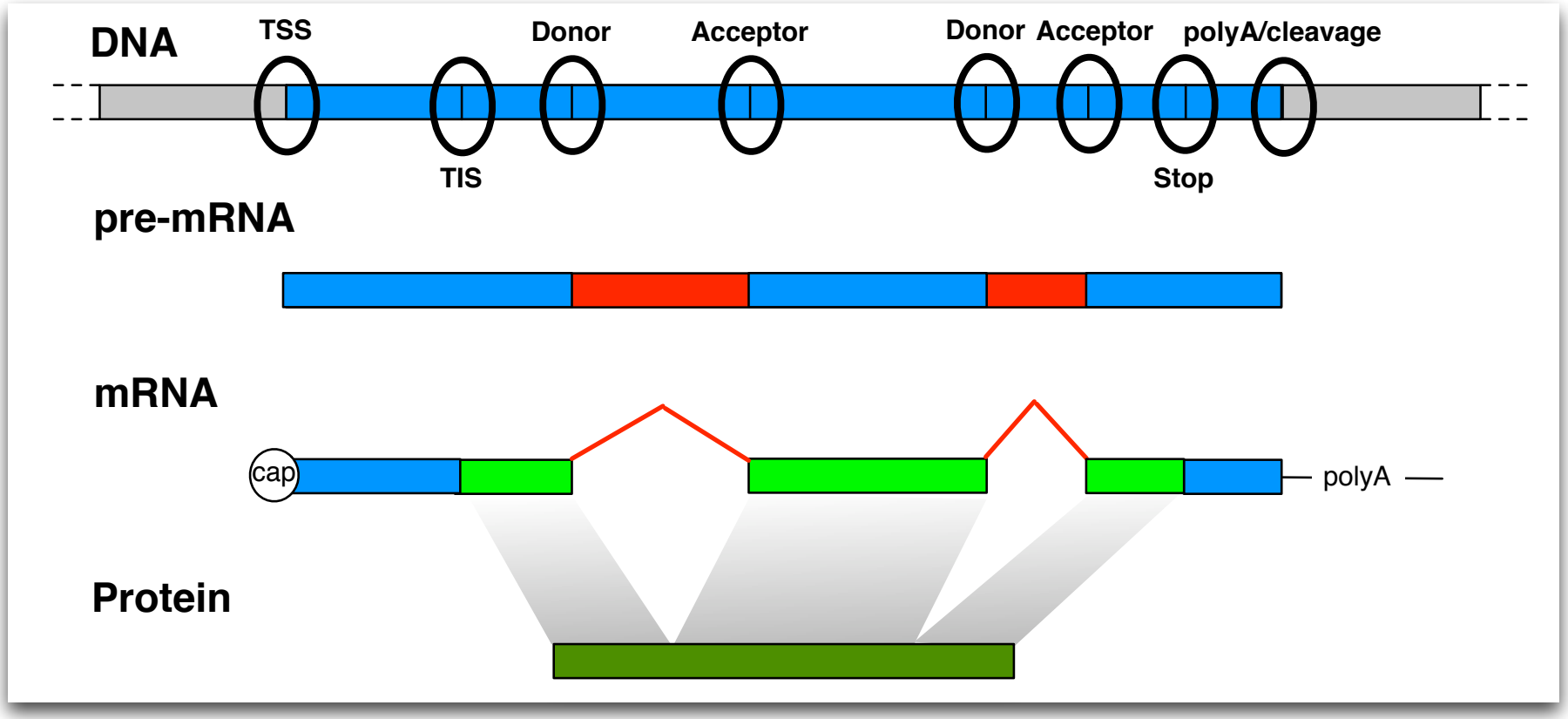
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Computational Gene Finding



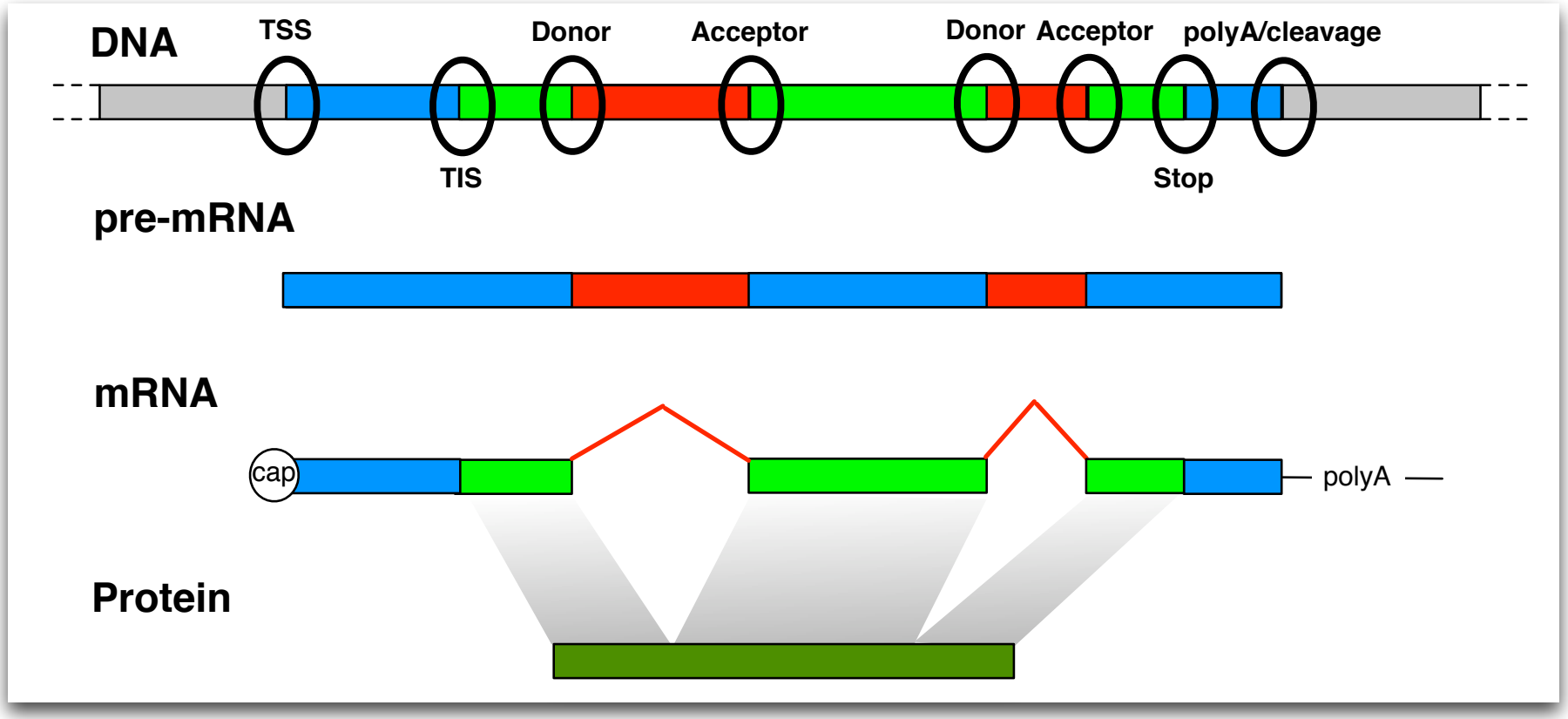
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Computational Gene Finding



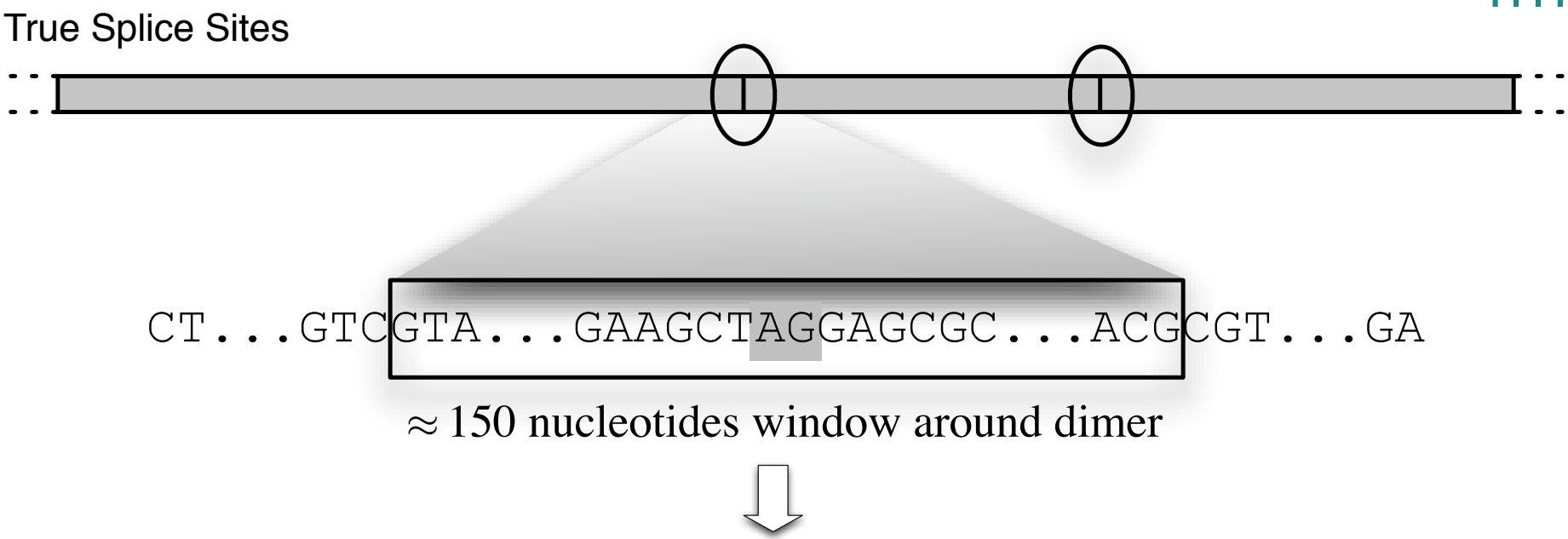
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Computational Gene Finding

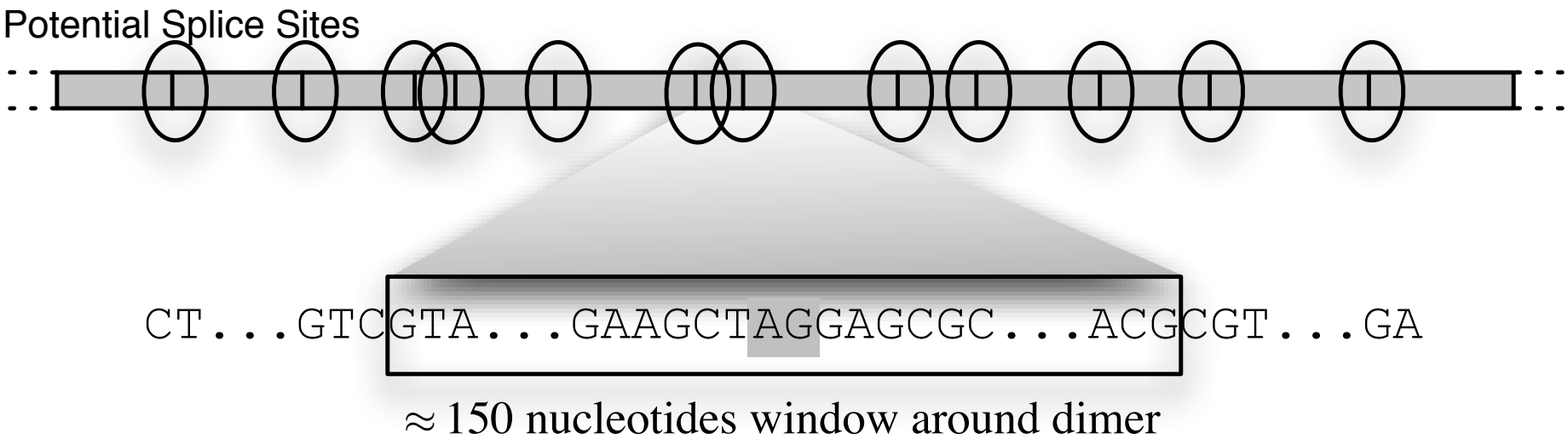


- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Example: Splice Site Recognition



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

- Newly sequenced organism: *C. briggsae*
  - $\approx$ 100 confirmed genes (590 splice site pairs)
- Well annotated relative: *C. elegans*
  - $\approx$ 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

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

The Bioinformatics Way of Transfer Learning

1. Use trained model of related organism, e.g. train on *C. elegans* and predict on other nematodes (Schweikert et al., 2009)

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

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

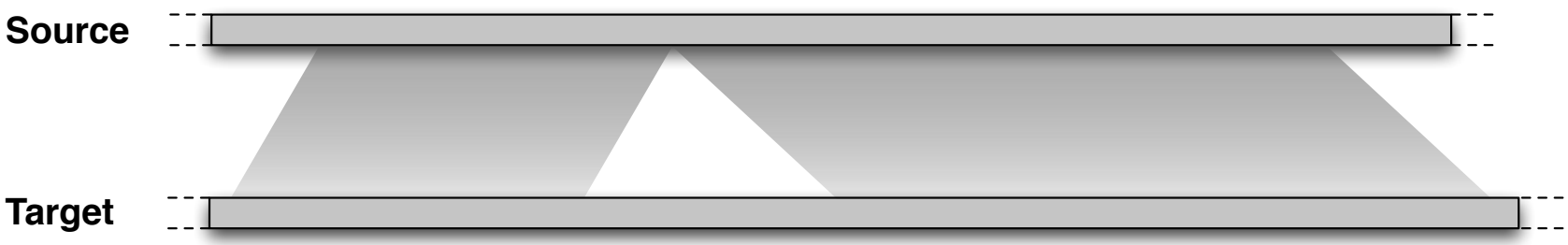
Works for closely related species, does not require any labeled data from target organism.

The Bioinformatics Way of Transfer Learning

1. Use trained model of related organism, e.g. train on *C. elegans* and predict on other nematodes (Schweikert et al., 2009)

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

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



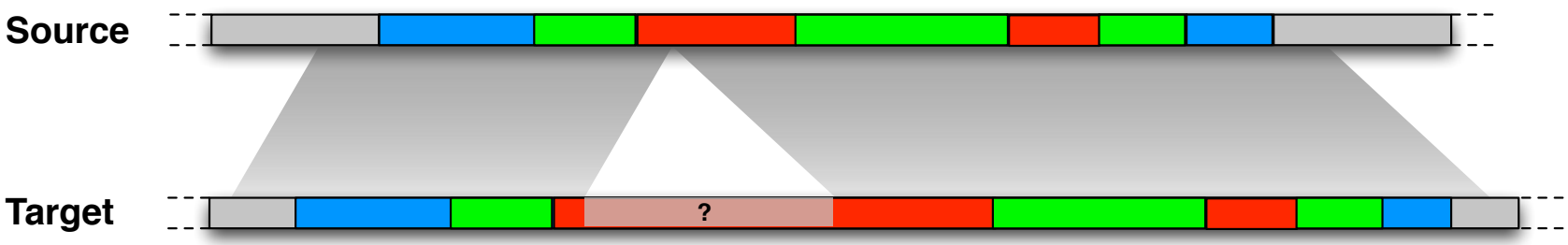
Works for closely related species, does not require any labeled data from target organism.

The Bioinformatics Way of Transfer Learning

1. Use trained model of related organism, e.g. train on *C. elegans* and predict on other nematodes (Schweikert et al., 2009)

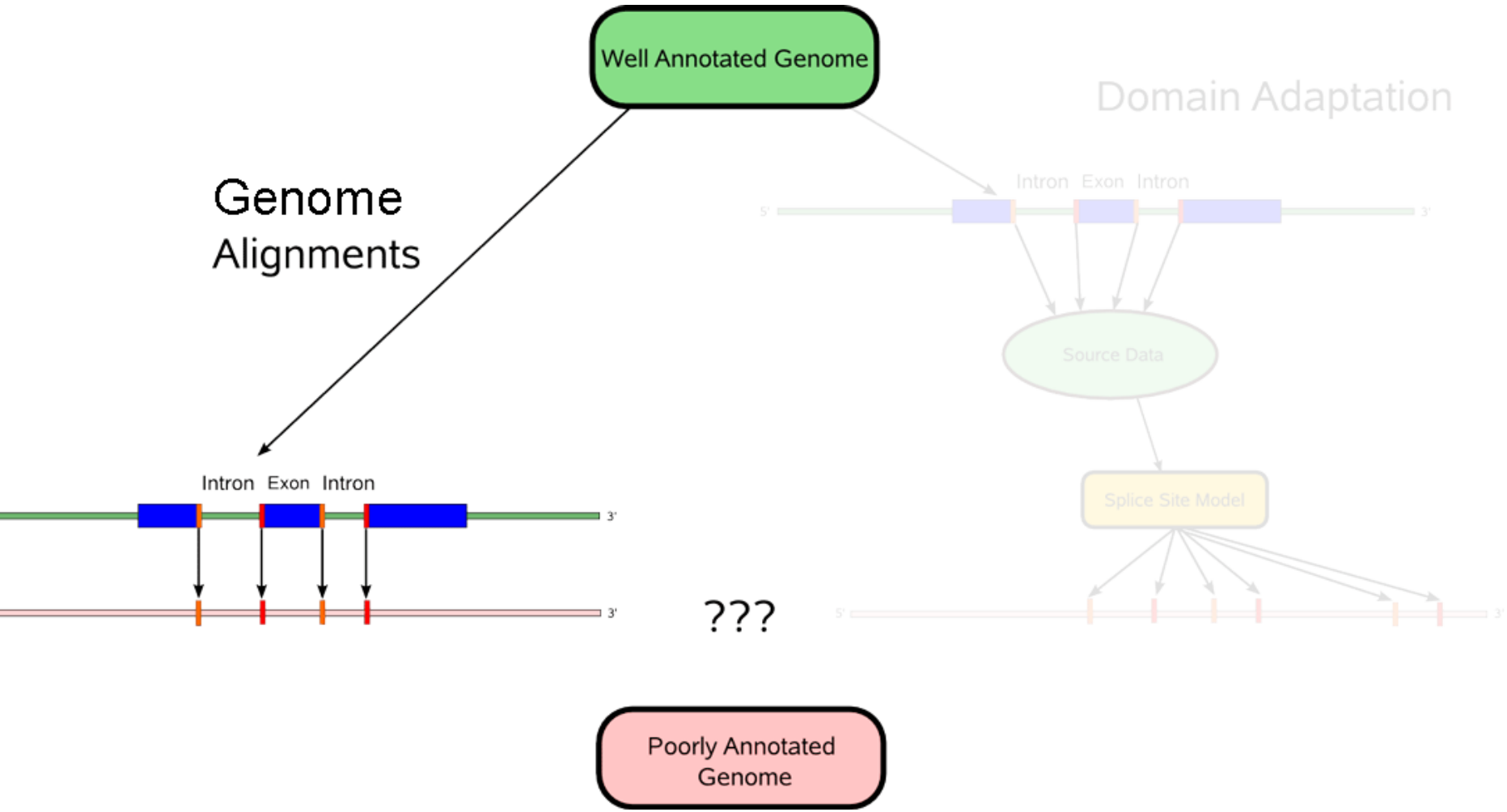
Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

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



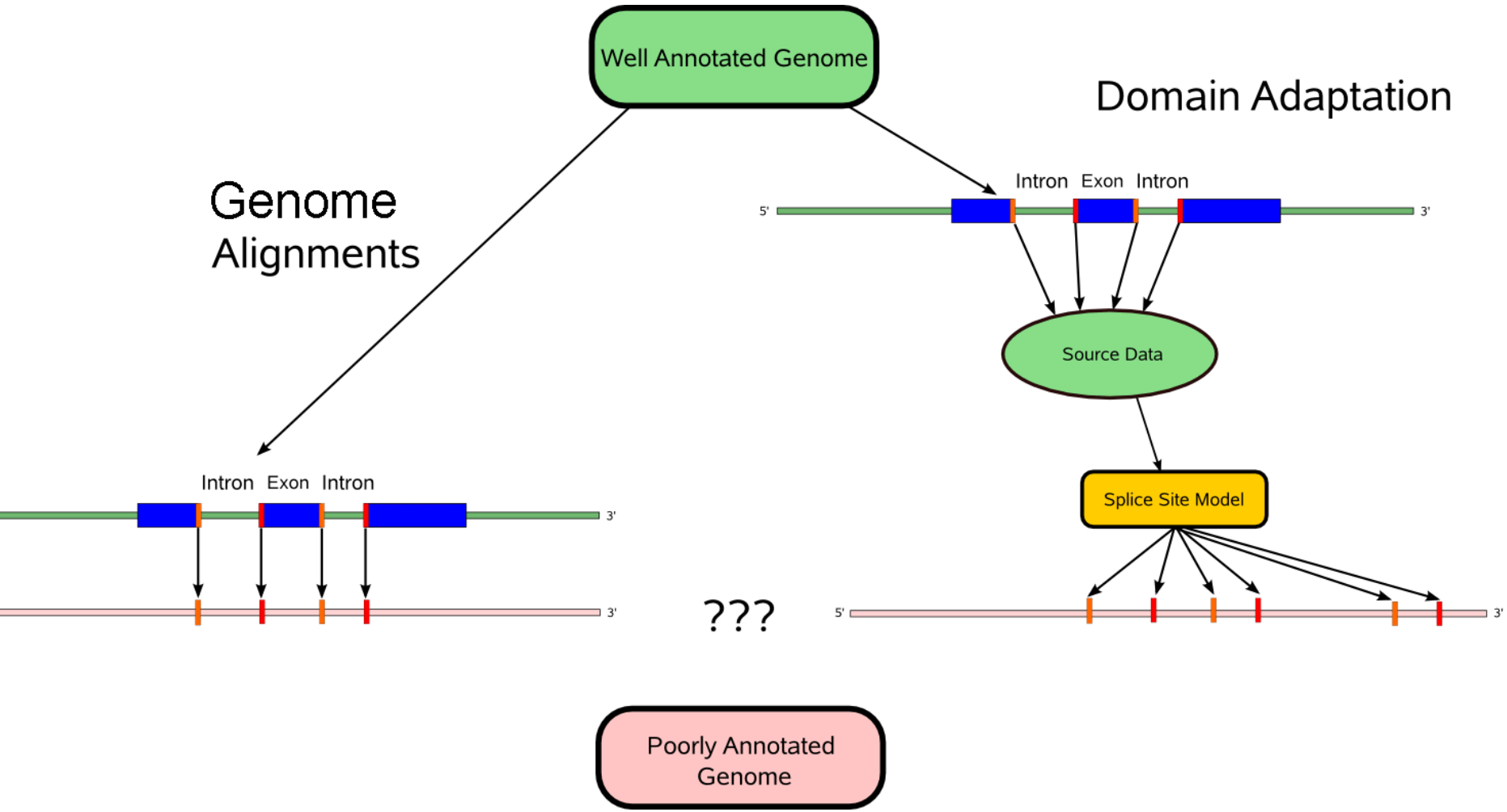
Works for closely related species, does not require any labeled data from target organism.

Domain Adaptation by Learning vs. Homology



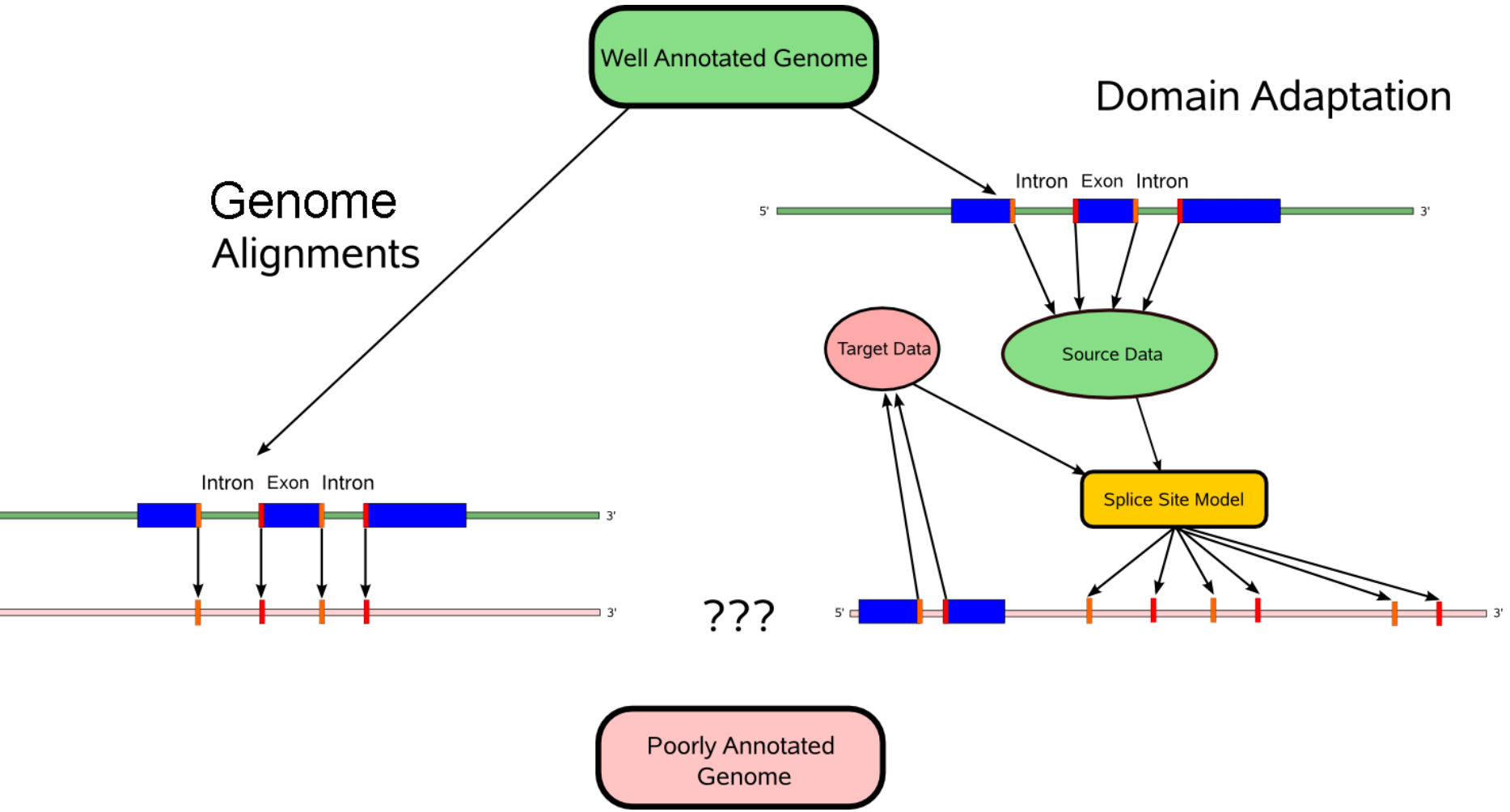
(Schweikert et al., 2008; Widmer et al., 2010)

Domain Adaptation by Learning vs. Homology



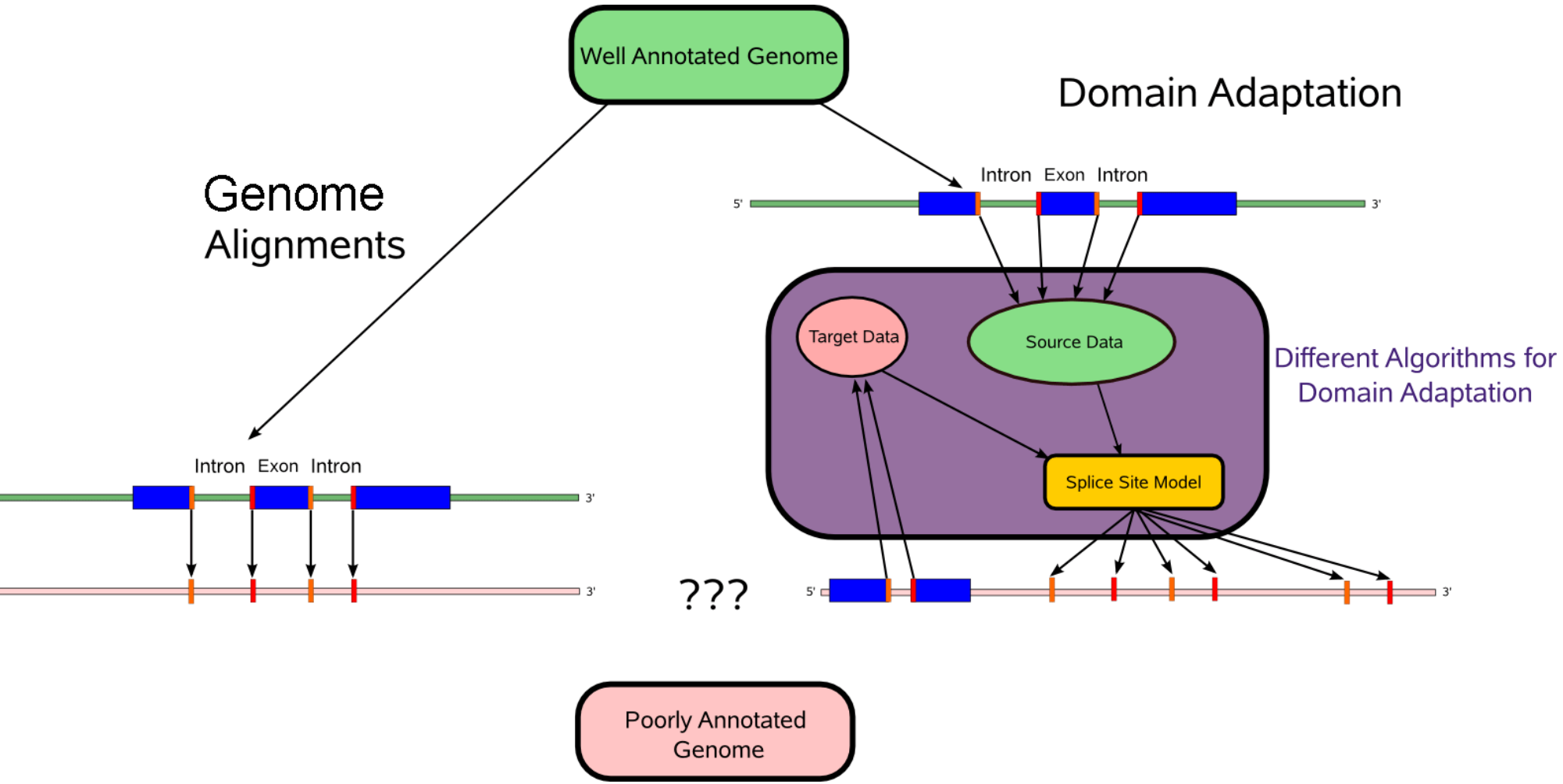
(Schweikert et al., 2008; Widmer et al., 2010)

Domain Adaptation by Learning vs. Homology



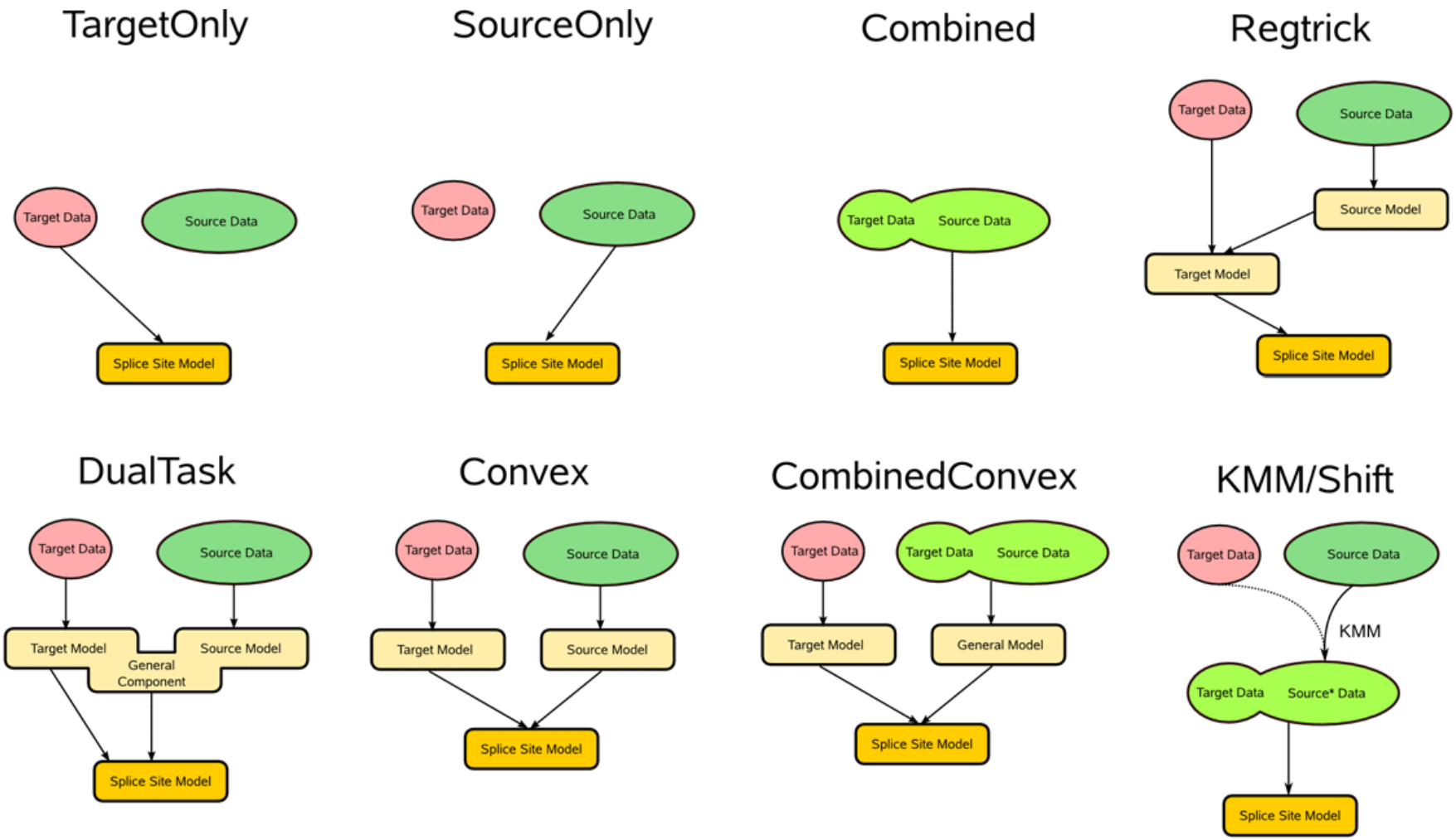
(Schweikert et al., 2008; Widmer et al., 2010)

Domain Adaptation by Learning vs. Homology



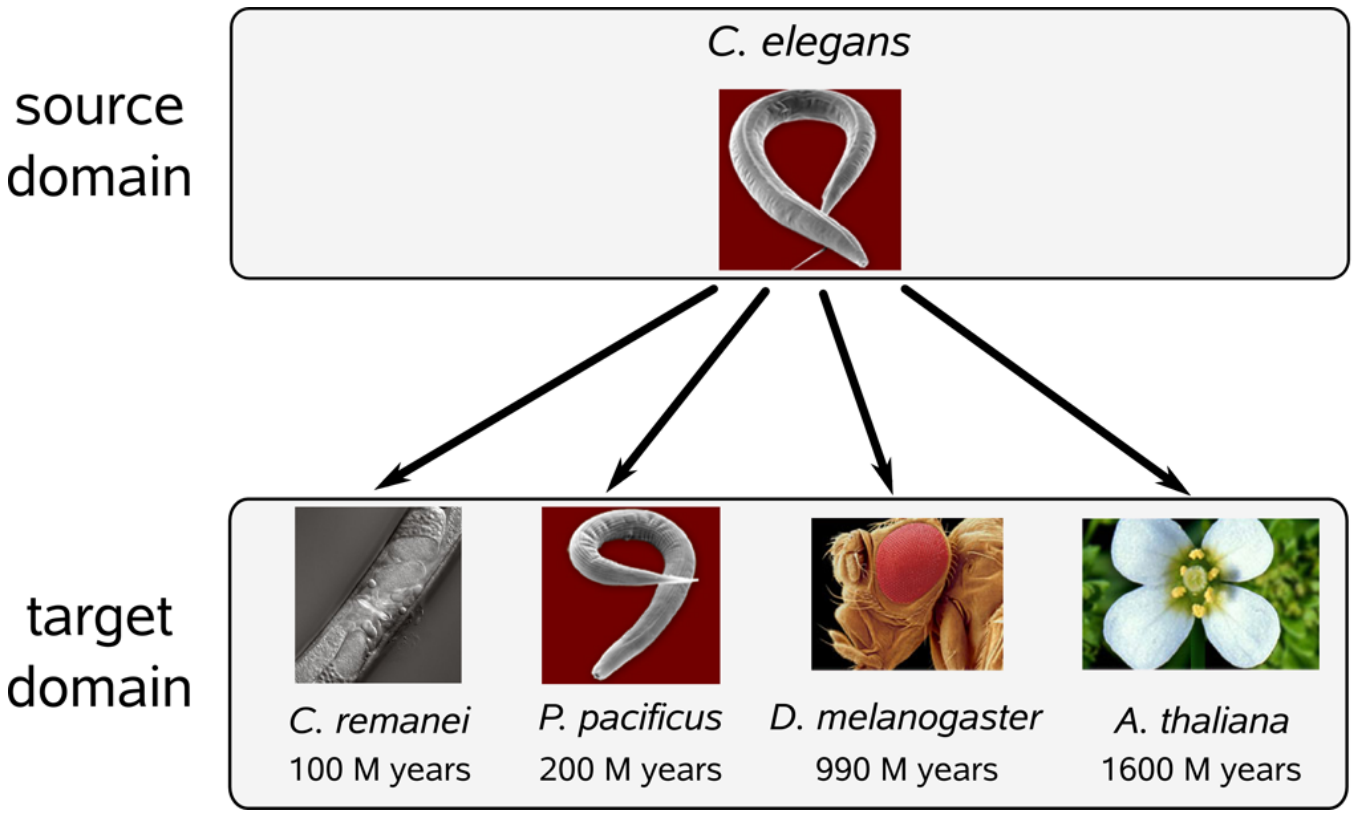
(Schweikert et al., 2008; Widmer et al., 2010)

Domain Adaptation Algorithms Overview



(Schweikert et al., 2008)

Large-Scale Empirical Comparison



- Varying distances
- Different data set sizes

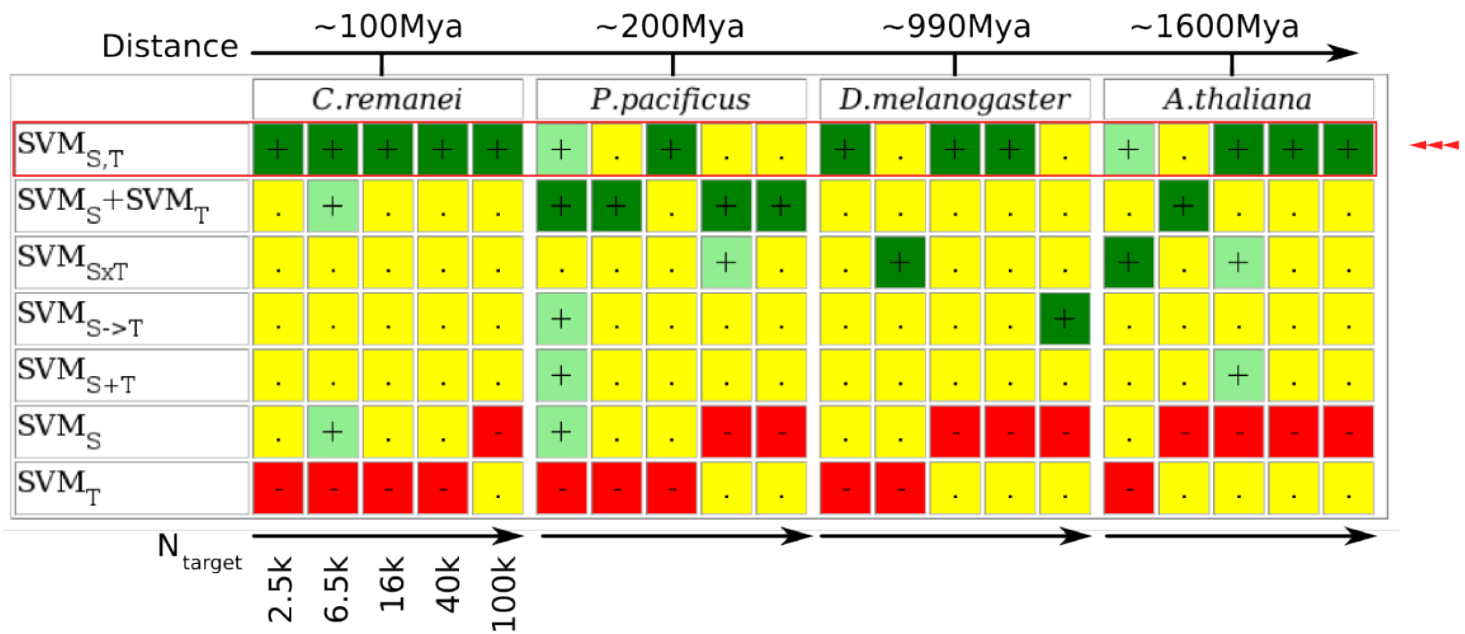
[MPI Developmental Biology and UCSC Genome Browser]

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

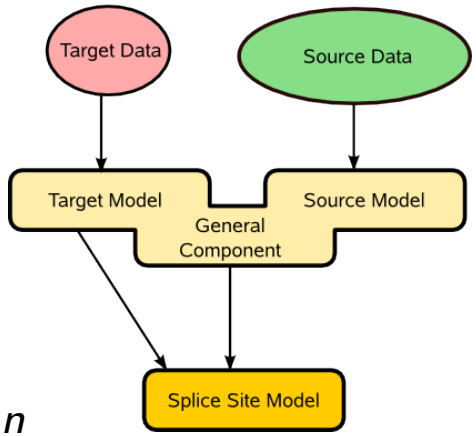
- Considerable improvements possible
- Sophisticated domain adaptation methods needed on distantly related organisms
- Best overall performance has *DualTask*
- Most cost effective *Convex/AdvancedConvex*



(Schweikert et al., 2008)

Domain Adaptation Methods

DualTask



Idea: (e.g., Caruana, 1997)

- Simultaneous optimization of both models
- Similarity between solution enforced

Approach:

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

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

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

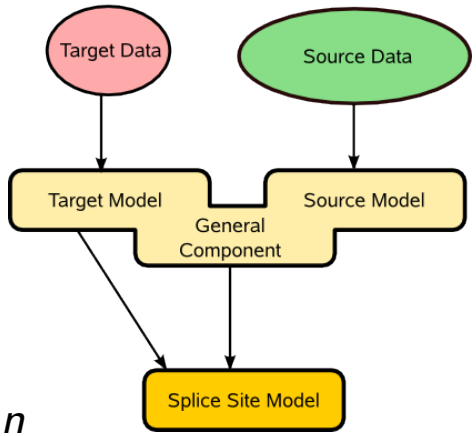
Equivalent to multi-task kernel learning: (Daume III, 2007)

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

for a suitably chosen Γ (p.s.d.).

Domain Adaptation Methods

DualTask



Idea: (e.g., Caruana, 1997)

- Simultaneous optimization of both models
- Similarity between solution enforced

Approach:

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

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

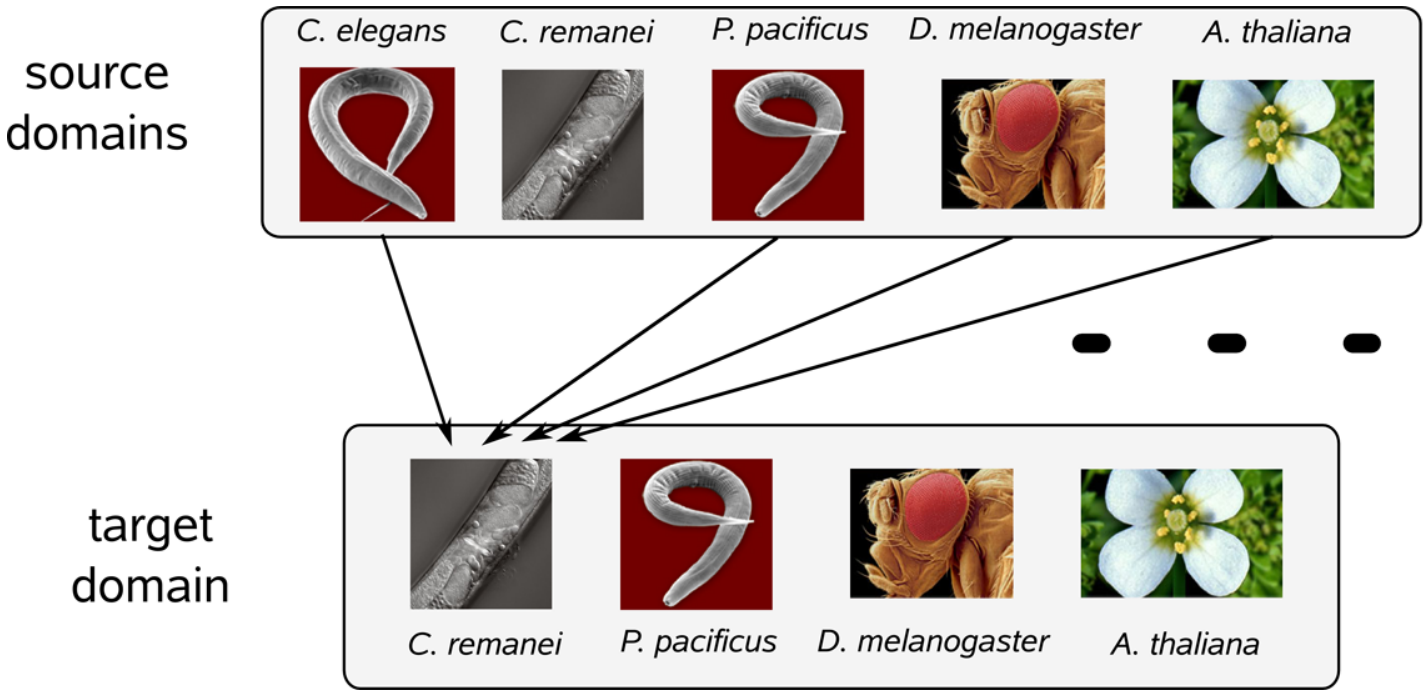
$$y_i(\langle \mathbf{w}_T, \Phi(\mathbf{x}_i)\rangle + b) \geq 1 - \xi_i \qquad i = m+1, \ldots, m+n$$

Equivalent to multi-task kernel learning: (Daume III, 2007)

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

for a suitably chosen Γ (p.s.d.).

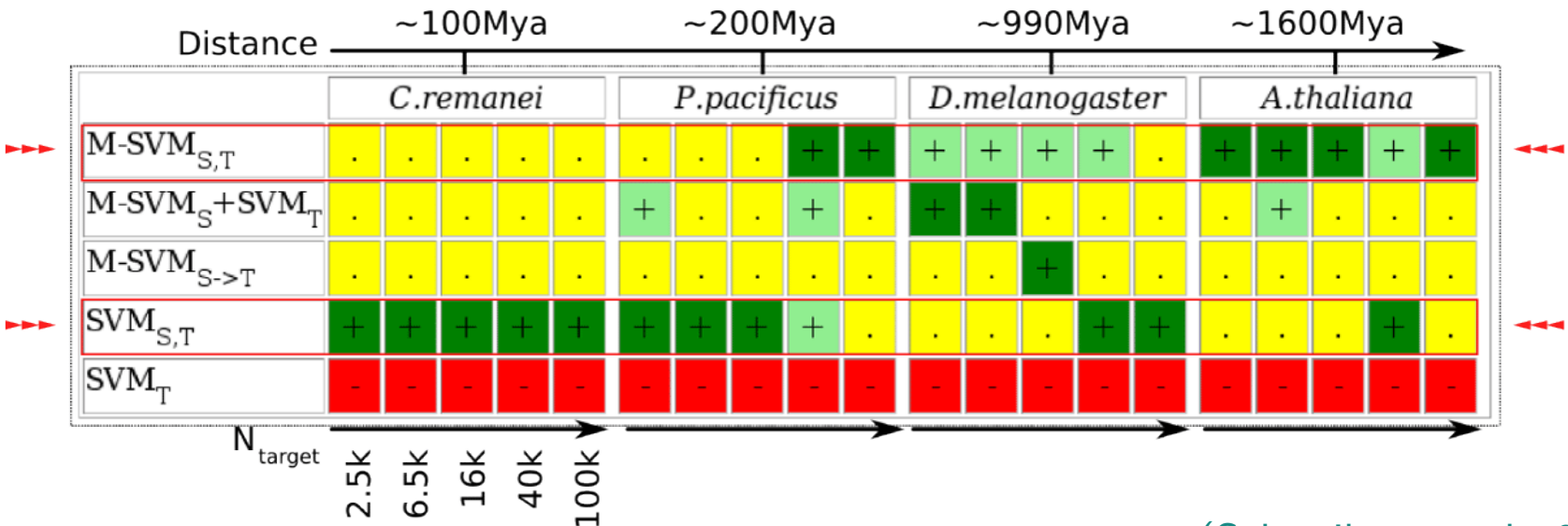
Multiple Source Domains



- 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

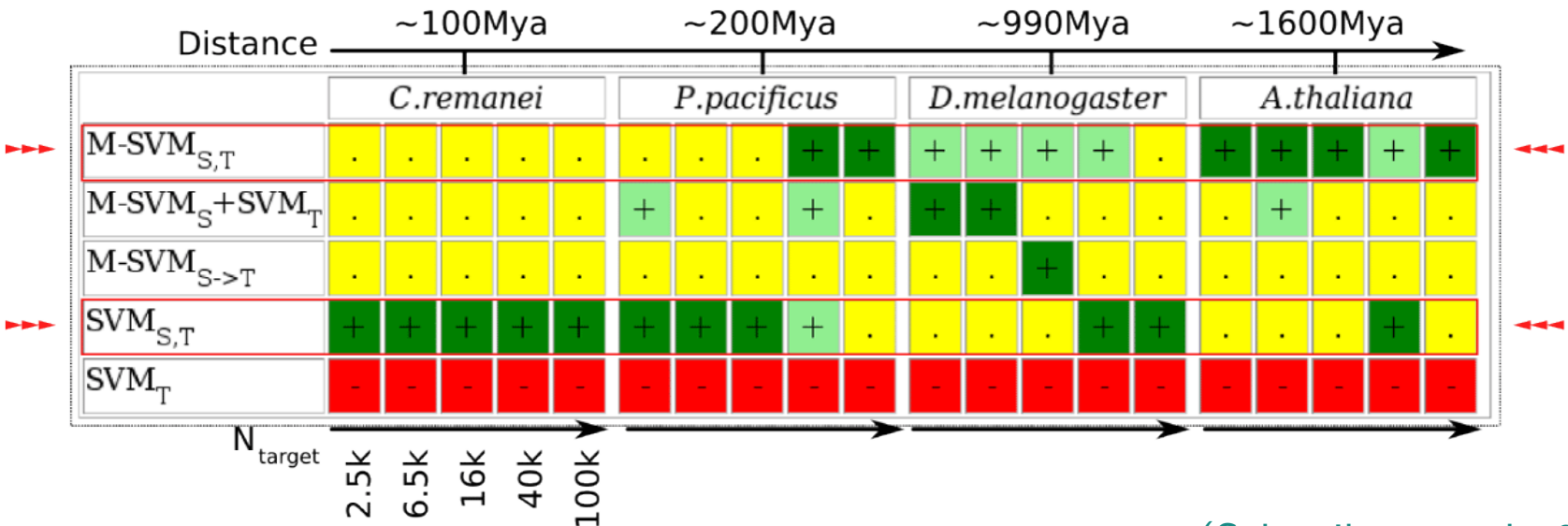


(Schweikert et al., 2008)

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



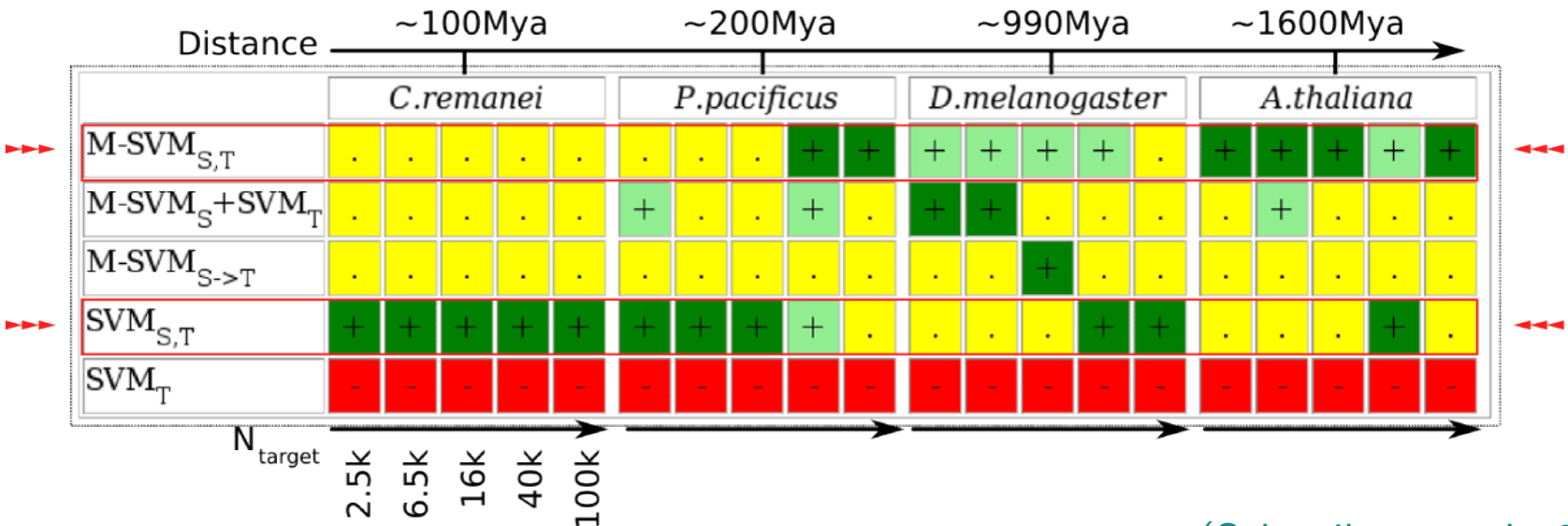
(Schweikert et al., 2008)

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., 2010)

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

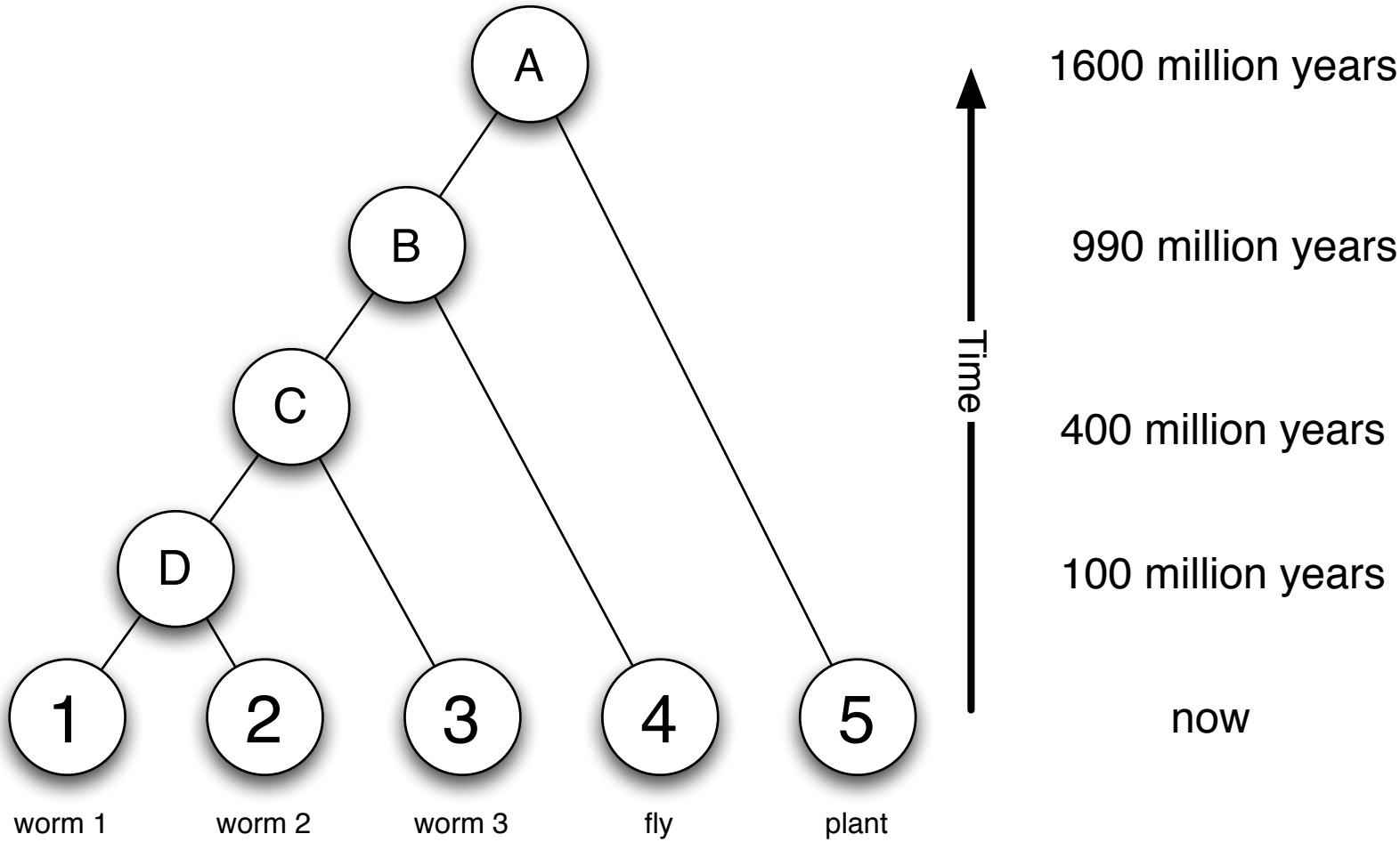
Top-Down

Multitask Kernel

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

(Widmer et al., 2010)

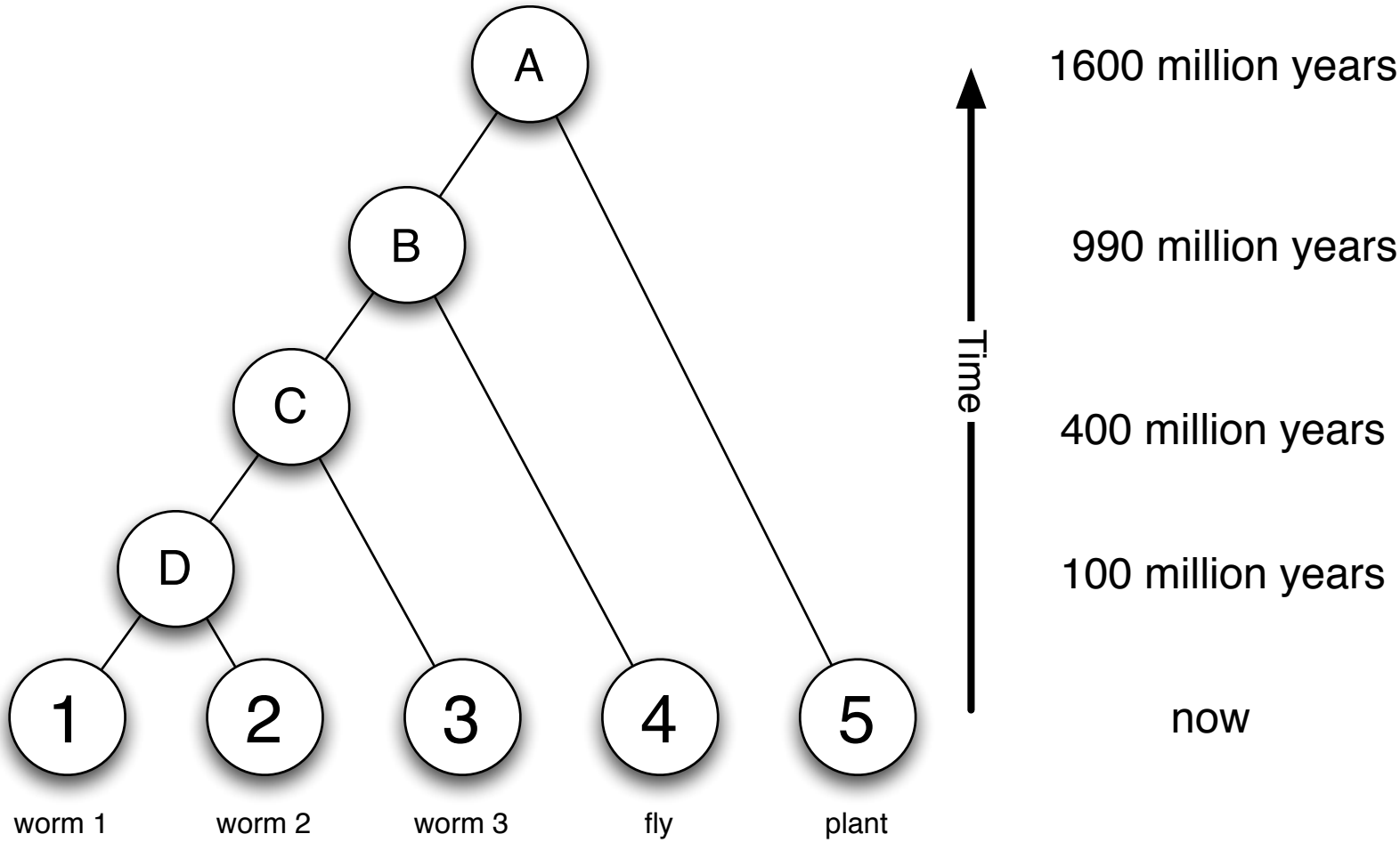
From Taxonomy to Γ



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

Strategies: $1/\text{years}$, *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*, ...

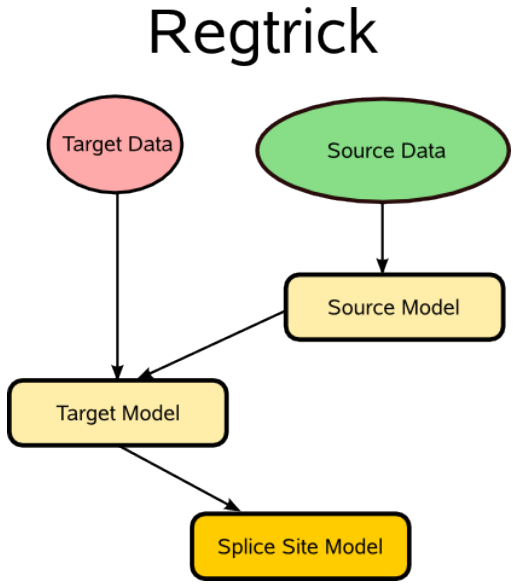
Hierarchical Top-Down Approach

Idea: Exploit taxonomy G in a top-down fashion

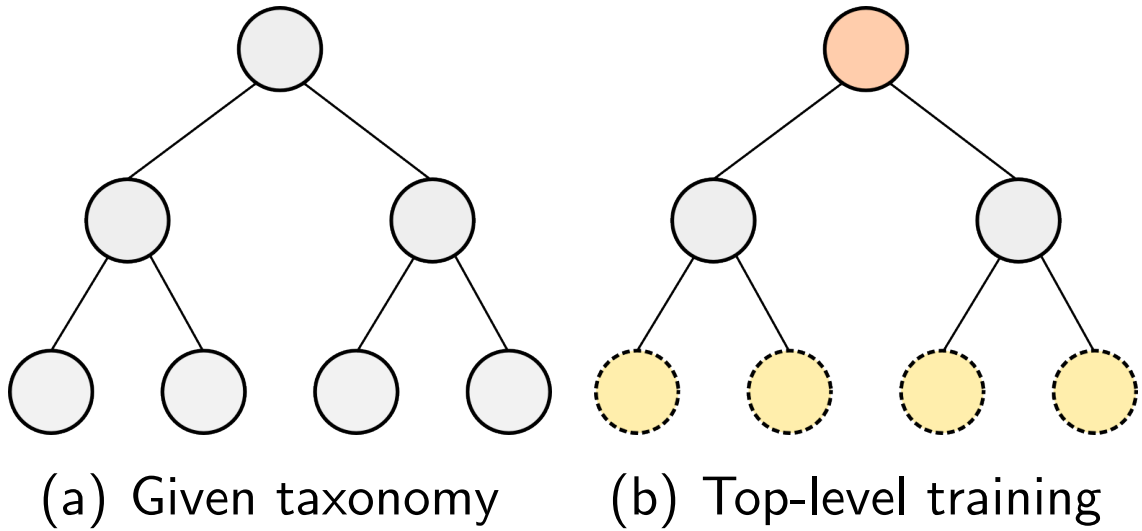
- Use taxonomy $\mathcal{T}$ in a top-down procedure
- Initialization: w_0 trained on union of all task datasets
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \succcurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{parent}$:

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

- Use leaf predictors for classification



Hierarchical Top-Down Approach: Illustration



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

(c) Intermediate training (d) Taxon training

Method Comparison

- Multitask Kernel Learning
  - Extension of existing kernel methods
  - Same run-time as regular SVM on union of all dataset
  - Requires appropriate *similarity matrix*
- Top-down Multitask Learning
  - Naturally exploits hierarchy
  - Does not need *task similarity matrix*
  - Heuristic
  - Needs training on unions of dataset
- Pairwise Multitask Learning
  - Optimization problem with well-defined global optimum
  - Practical issues due to decomposition
  - Requires appropriate *similarity matrix*

Method Comparison

- Multitask Kernel Learning
  - Extension of existing kernel methods
  - Same run-time as regular SVM on union of all dataset
  - Requires appropriate *similarity matrix*
- Top-down Multitask Learning
  - Naturally exploits hierarchy
  - Does not need *task similarity matrix*
  - Heuristic
  - Needs training on unions of dataset
- Pairwise Multitask Learning
  - Optimization problem with well-defined global optimum
  - Practical issues due to decomposition
  - Requires appropriate *similarity matrix*

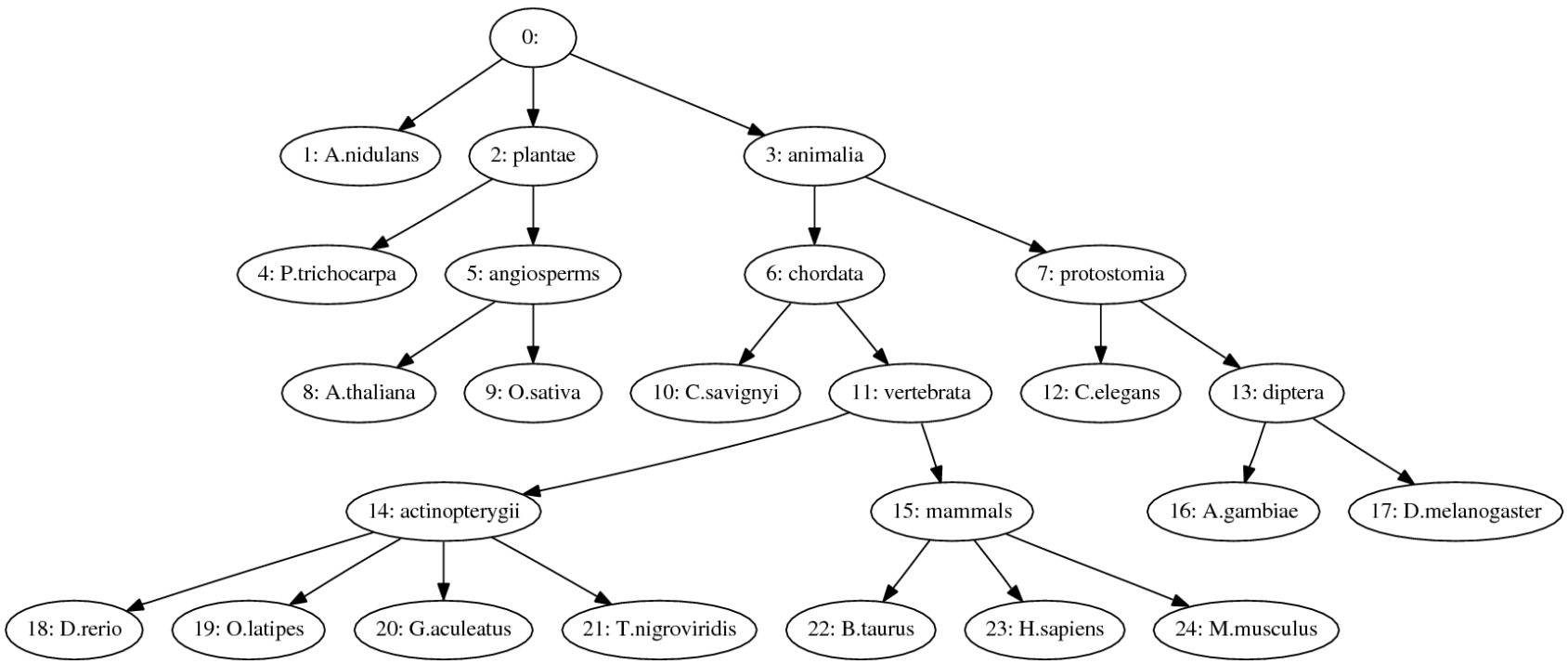
Application to Splicing Data

- Formulation as binary classification problem

≈150 nucleotides window around dimer

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

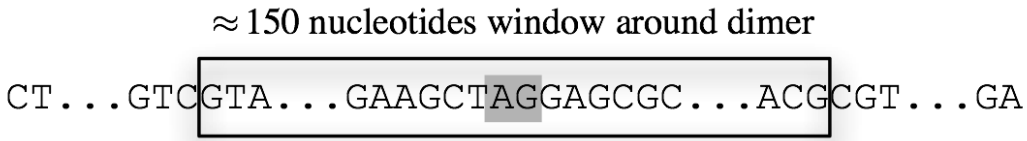
- Utilize 15 organisms related by taxonomy



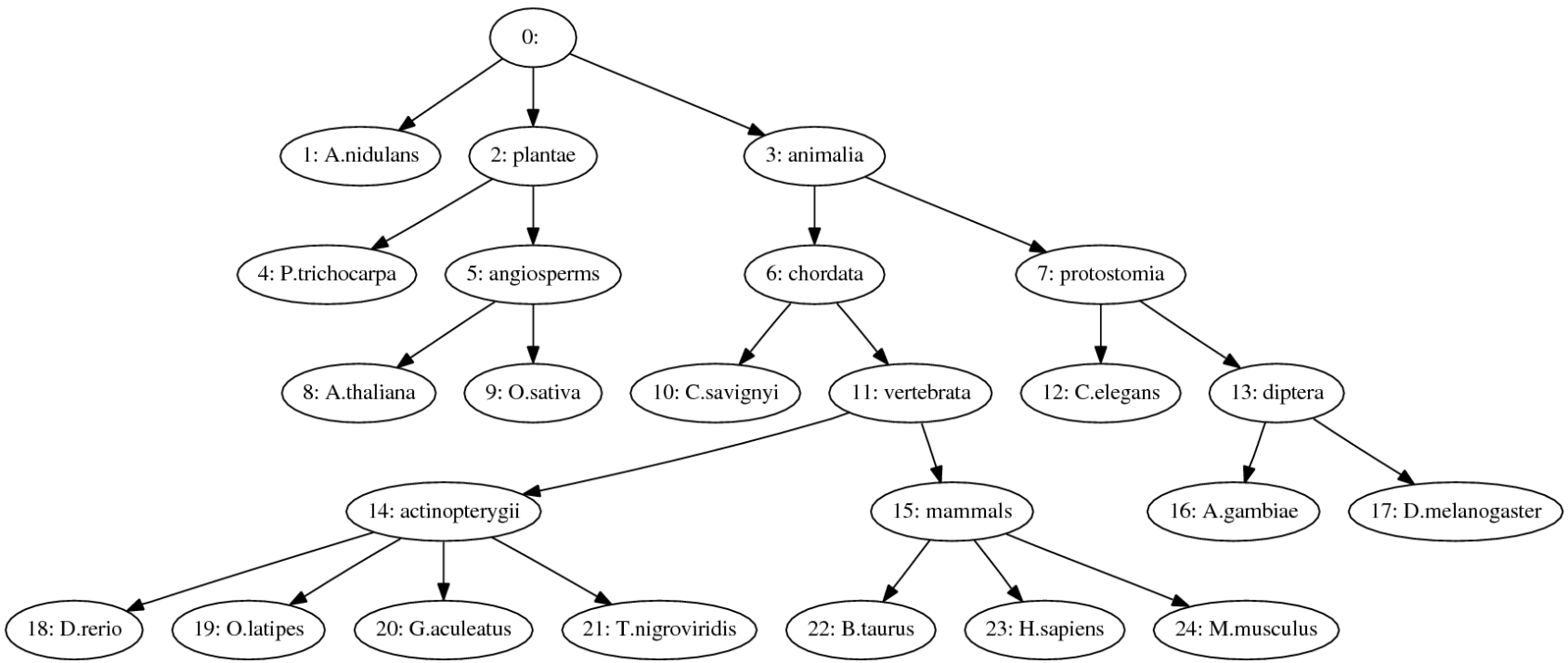
- Restricted to at most 10.000 examples per organism

Application to Splicing Data

- Formulation as binary classification problem

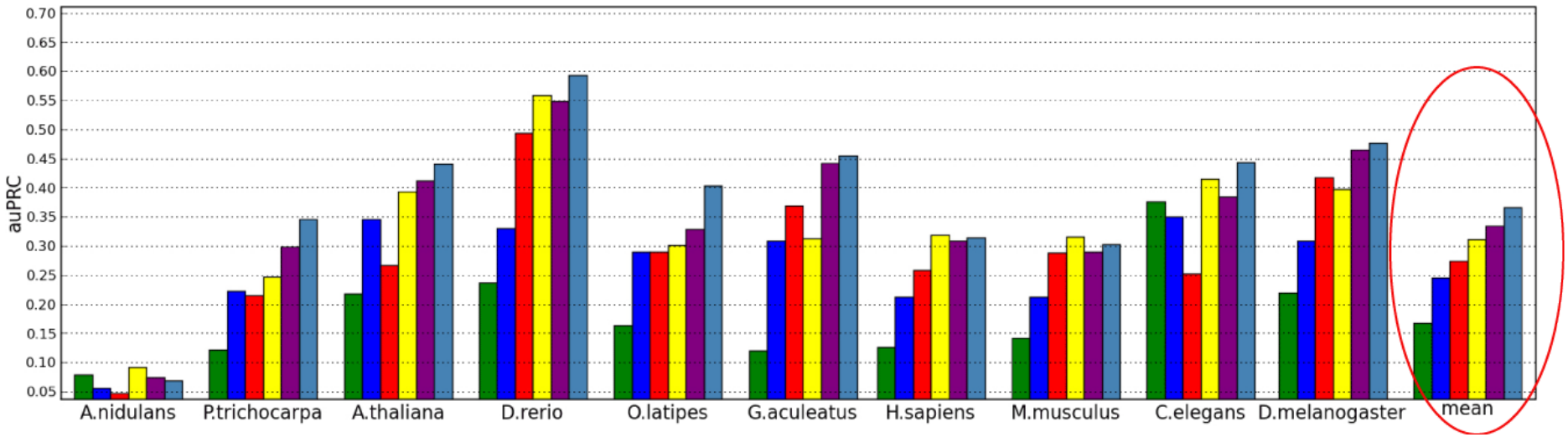


- Utilize 15 organisms related by taxonomy



- Restricted to at most 10.000 examples per organism

Results: Splicing Data



Observations:

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

Discussion

- 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.

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

- Idea for top-down approach: local cross-validation

Discussion

- 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.
    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$
- Idea for top-down approach: local cross-validation

Discussion

- 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.

    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$
- Idea for top-down approach: local cross-validation

Discussion

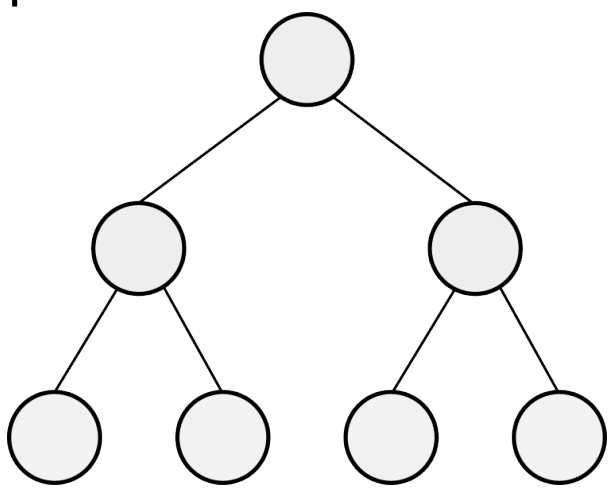
- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$
- Idea for top-down approach: local cross-validation

Discussion

- 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.

    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$
- Idea for top-down approach: local cross-validation



Multiple Kernel Learning with Meta-tasks



Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

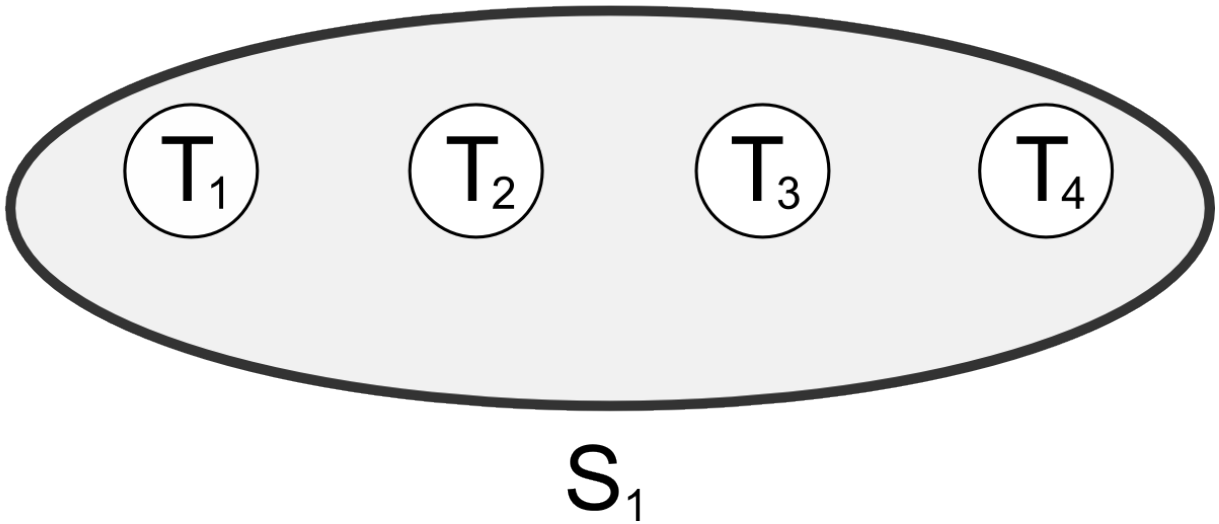


Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

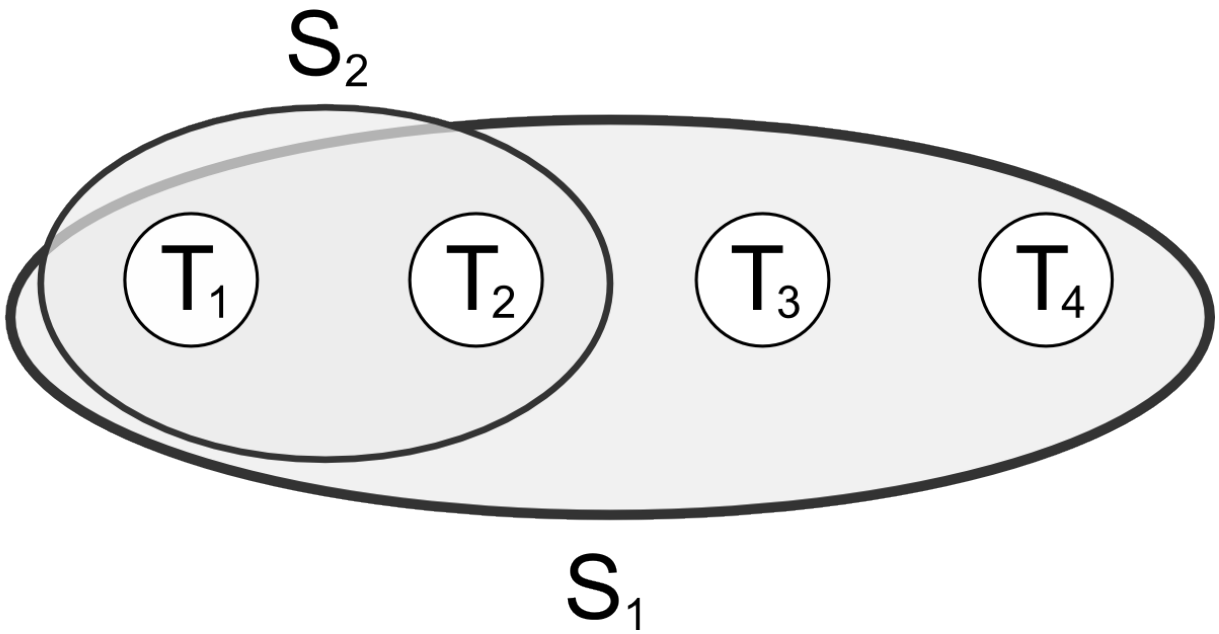


Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

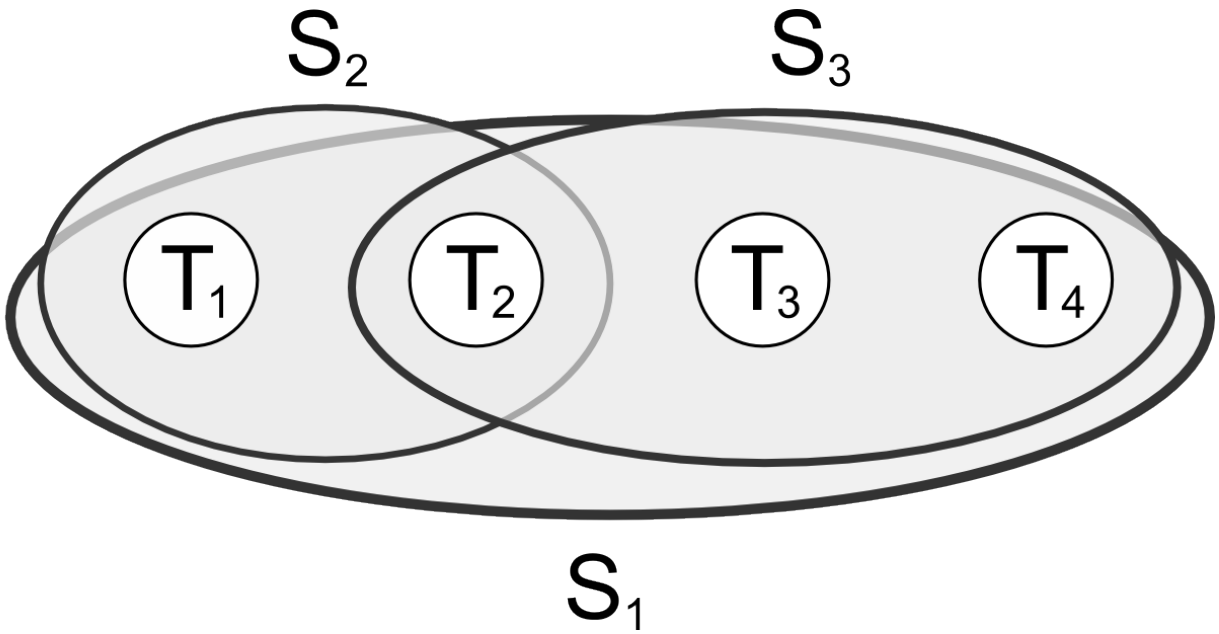


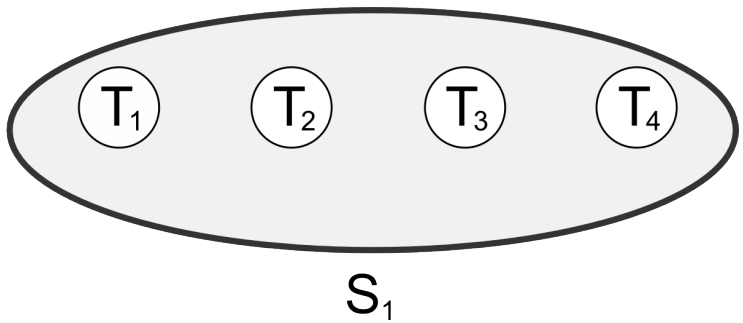
Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

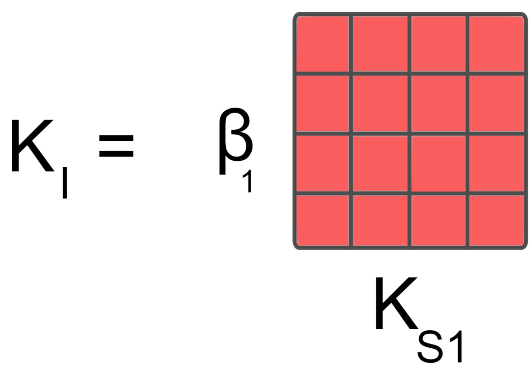
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



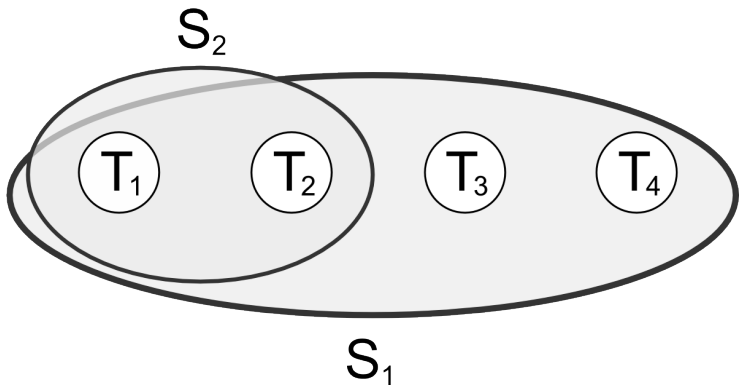
Example for collection of meta-tasks:



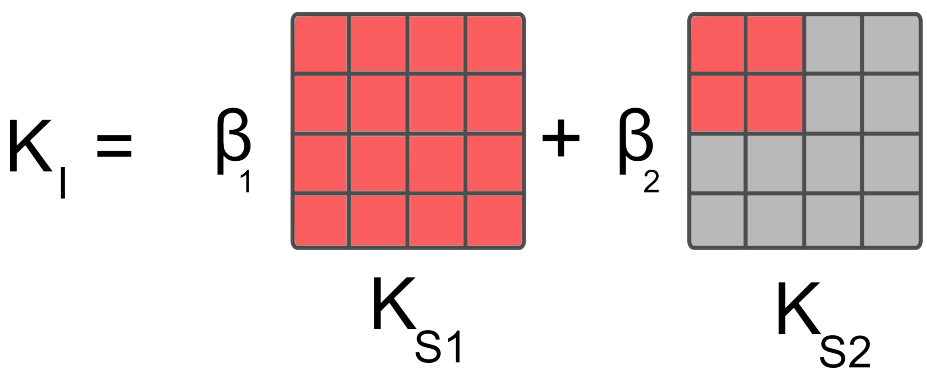
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



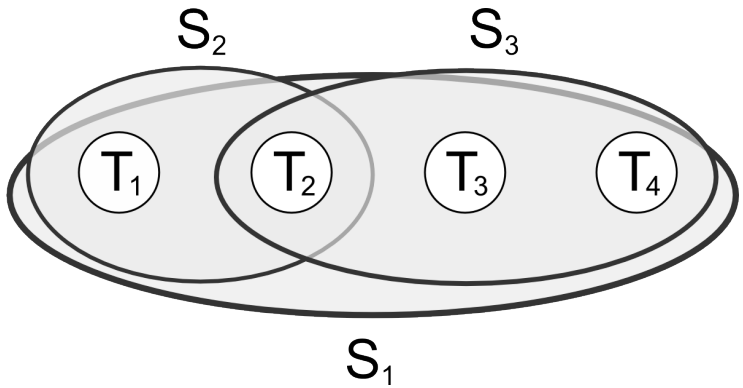
Example for collection of meta-tasks:



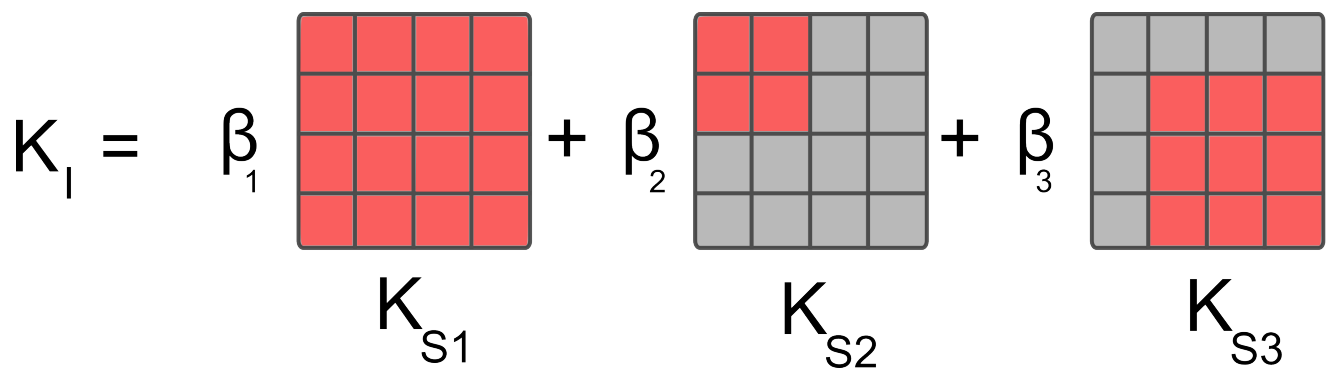
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



Example for collection of meta-tasks:



Optimization strategy: q-norm MKL

We use the MKL formulation: (Kloft et al., 2009)

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}(x_i,x_j)$$

$$\text{s.t.} \qquad ||\boldsymbol{\beta}||_q^q \le 1, \boldsymbol{\beta} \ge 0$$

$$\mathbf{Y}^T\boldsymbol{\alpha} = 0, 0 \le \boldsymbol{\alpha} \le C$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Optimization strategy: q-norm MKL

We use the MKL formulation: (Kloft et al., 2009)

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}(x_i,x_j)$$

$$\text{s.t.} \qquad ||\boldsymbol{\beta}||_q^q \le 1, \boldsymbol{\beta} \ge 0$$

$$\mathbf{Y}^T\boldsymbol{\alpha} = 0, 0 \le \boldsymbol{\alpha} \le C$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Power-set based approach

If no prior information available:

$$\mathcal{I}_{\mathcal{P}} = \{S | S \in \mathcal{P}(\mathcal{T}) \wedge S \neq \emptyset\}$$

- Consider Powerset $\mathcal{P}(\mathcal{T})$
- Most meta-tasks in power-set will be meaningless
  $\rightarrow$ learn sparse weights: $q = 1$
- Approach can be used to identify task structure *ab initio*
- 2^M meta-tasks $\rightarrow$ computationally expensive

Hierarchical decomposition

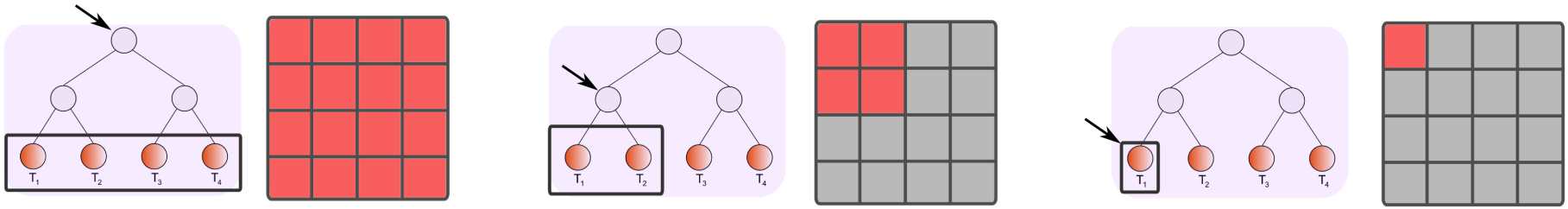
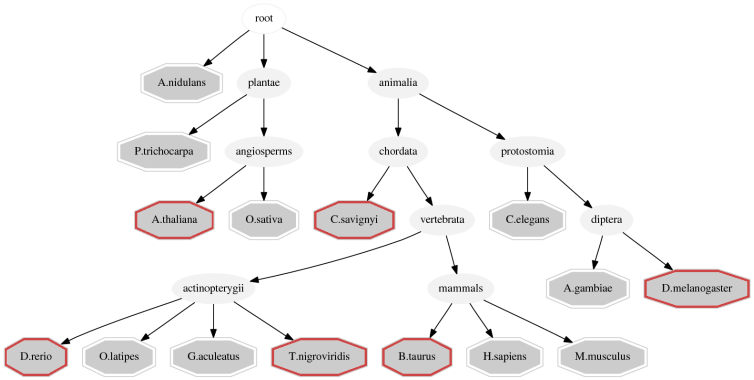


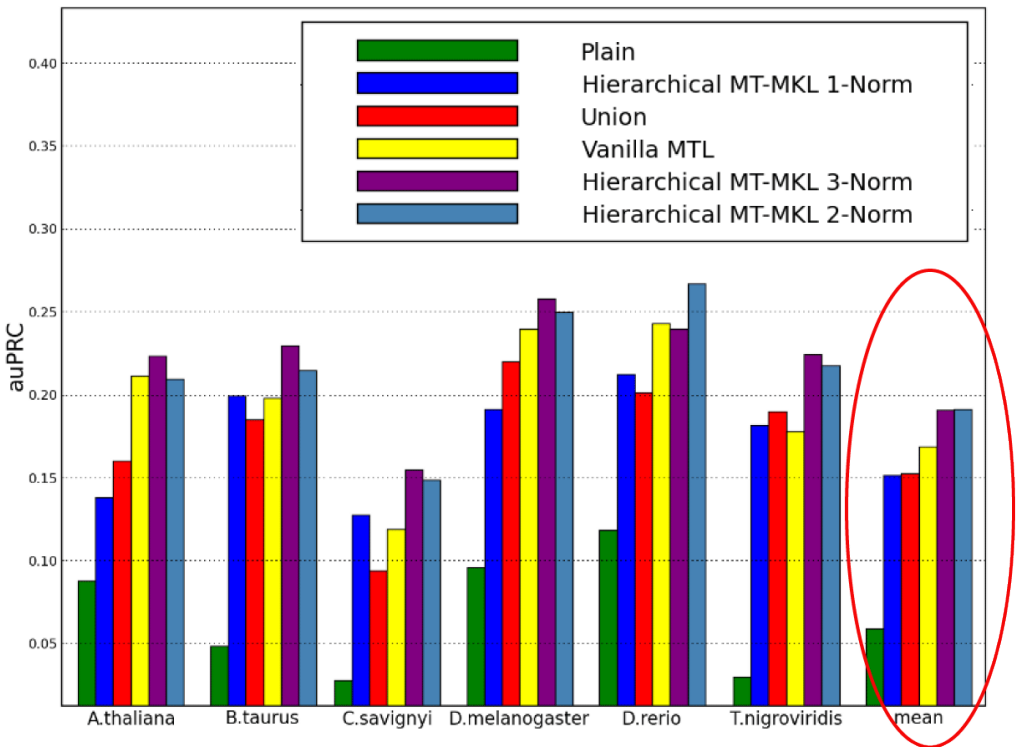
Figure: Example of taxonomy-based decomposition.

$$\mathcal{I}_{\mathcal{G}} = \{leaves(node) | node \in \mathcal{G}\}$$

- Meta-tasks are defined by taxonomy $\mathcal{G}$
- Taxonomy $\mathcal{G}$ gives us reasonable groups
  - Idea is to refine structure
  - Non-sparse combination ($q > 1$) for groupings

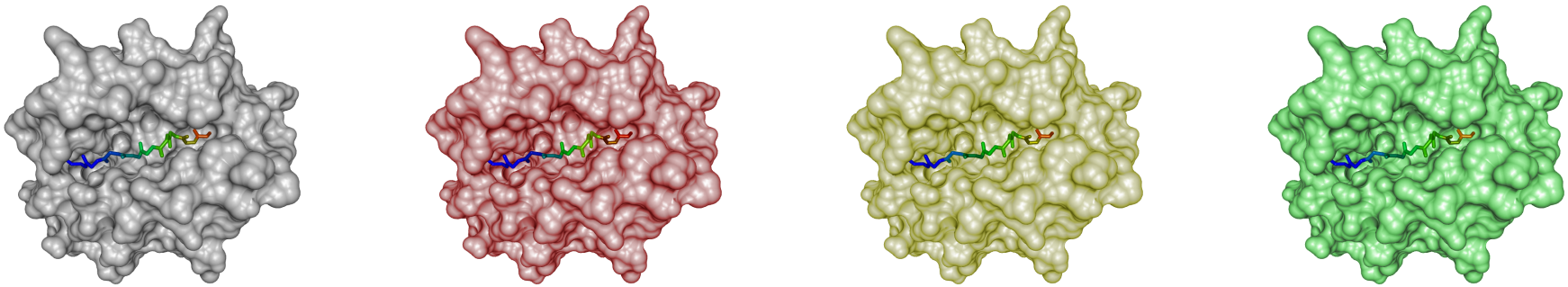
Experiments (a): Splice-site recognition





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

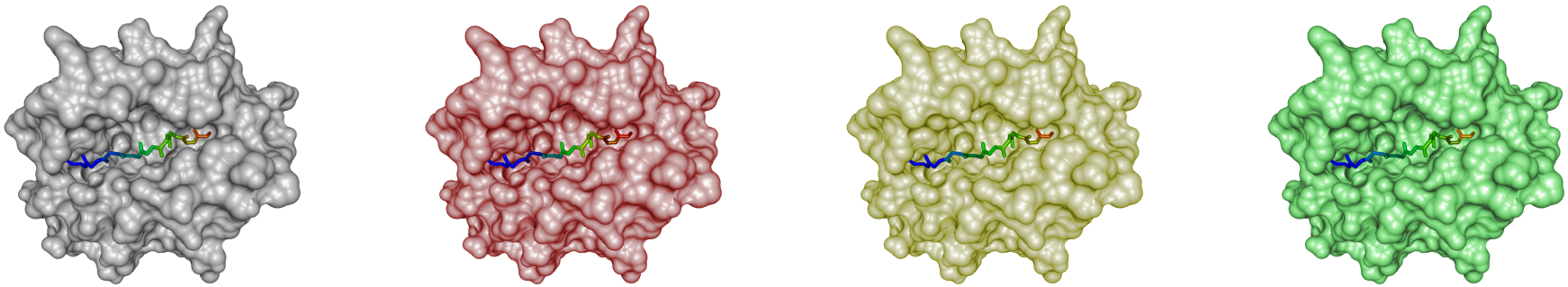
Experiments (b): MHC-I binding prediction



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?

Experiments (b): MHC-I binding prediction

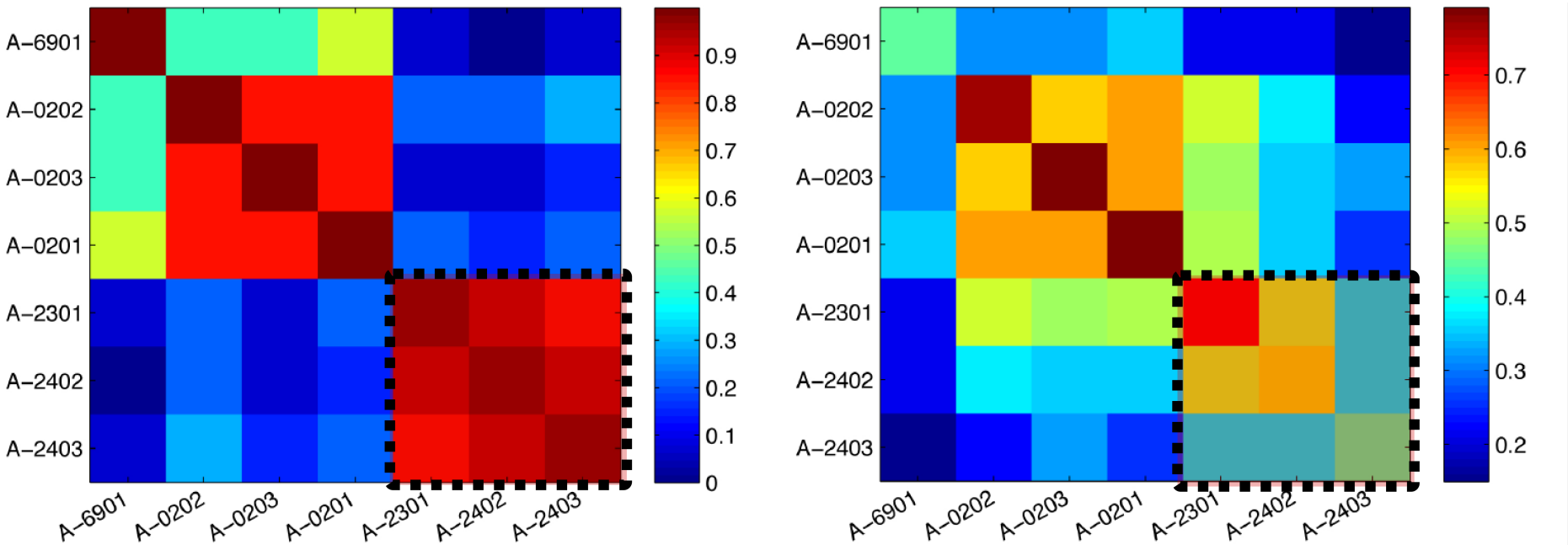


Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	69.9%

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

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

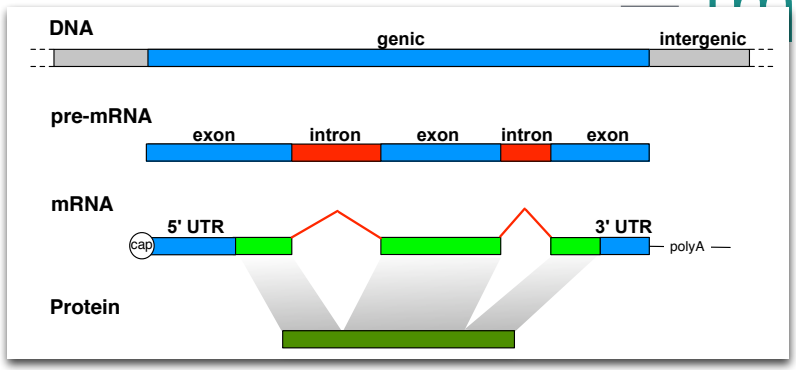
- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

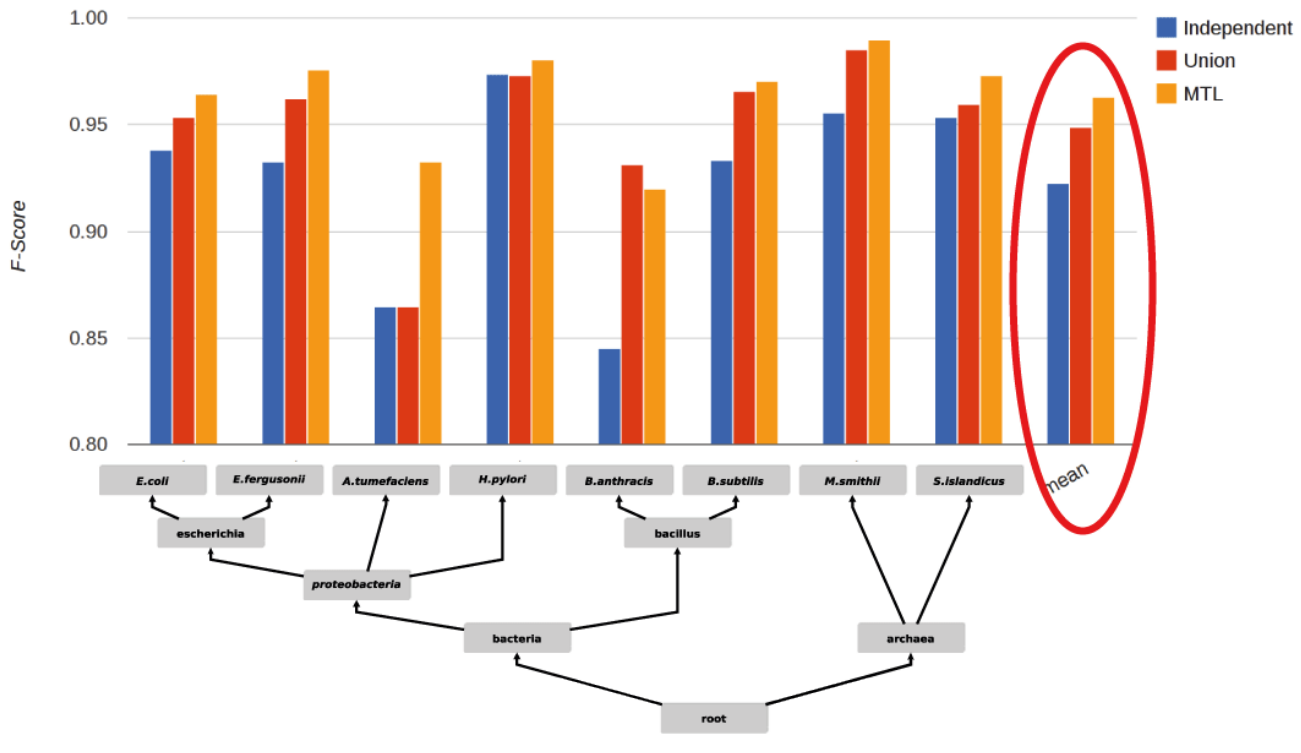
Extension to Structured Output Learning

Stratgies only work on regularizer

Also works 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 difficult & critical

Recently extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning

Material available at:
`www.fml.mpg.de/raetsch/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 difficult & critical

Recently extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning

Material available at:
`www.fml.mpg.de/raetsch/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 difficult & critical

Recently extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning

Material available at:
`www.fml.mpg.de/raetsch/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 difficult & critical

Recently extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning

Material available at:
`www.fml.mpg.de/raetsch/suppl/transfer-learning`

Acknowledgements



Christian Widmer

(FML, Tübingen)



Gabriele Schweikert

(MPI & FML, Tübingen)



Jose Leiva

(University of Madrid)

Also involved

- Nora Toussaint
- Nico Görnitz
- Sören Sonnenburg
- Yasemin Altun
- Bernhard Schölkopf

Funding by DFG &
Max Planck Society.

Thank you for your attention!

Acknowledgements



Christian Widmer

(FML, Tübingen)



Gabriele Schweikert

(MPI & FML, Tübingen)



Jose Leiva

(University of Madrid)

Also involved

- Nora Toussaint
- Nico Görnitz
- Sören Sonnenburg
- Yasemin Altun
- Bernhard Schölkopf

Funding by DFG &
Max Planck Society.

Thank you for your attention!

Domain Adaptation for LSL

Problem:

- Very little data
  - Relatively small amount of sequences available
  - Only ≈50 well analysed genes

Solution:

- Exploit that *P. pacificus* is closely related to *C. elegans*
- Can use *C. elegans* signal and content sensors
- But how can *adapt* the *C. elegans* parameters for gene structure prediction?

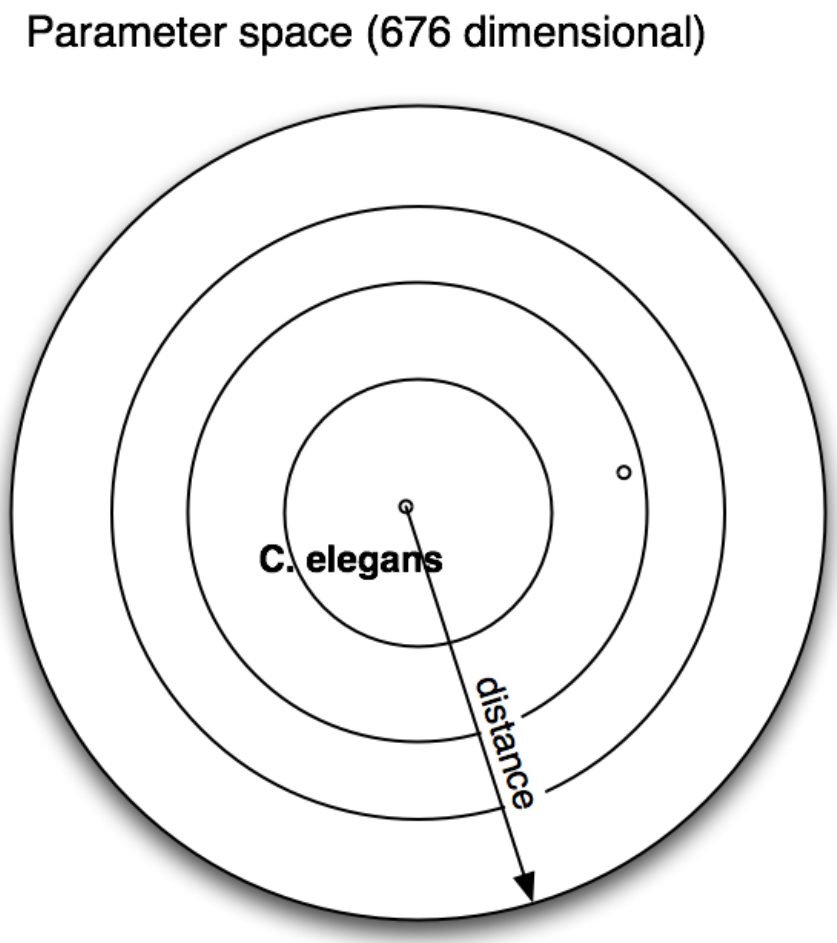
Domain Adaptation for LSL

Problem:

- Very little data
  - Relatively small amount of sequences available
  - Only ≈50 well analysed genes

Solution:

- Exploit that *P. pacificus* is closely related to *C. elegans*
- Can use *C. elegans* signal and content sensors
- But how can *adapt* the *C. elegans* parameters for gene structure prediction?



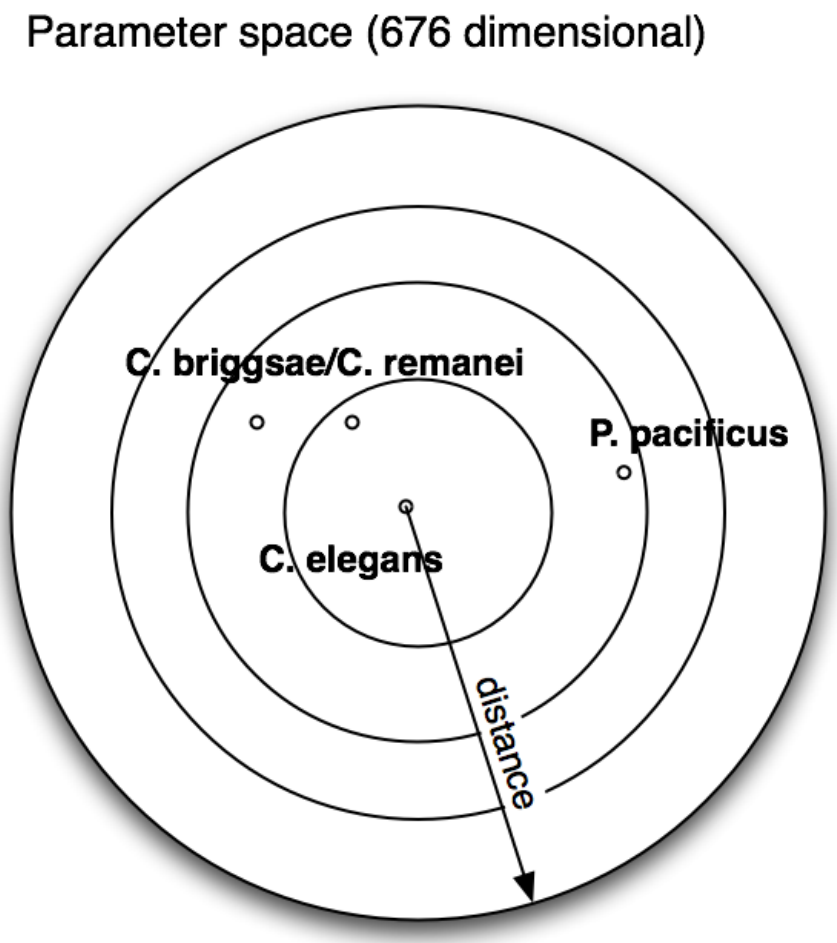
Domain Adaptation for LSL

Problem:

- Very little data
  - Relatively small amount of sequences available
  - Only ≈50 well analysed genes

Solution:

- Exploit that *P. pacificus* is closely related to *C. elegans*
- Can use *C. elegans* signal and content sensors
- But how can *adapt* the *C. elegans* parameters for gene structure prediction?



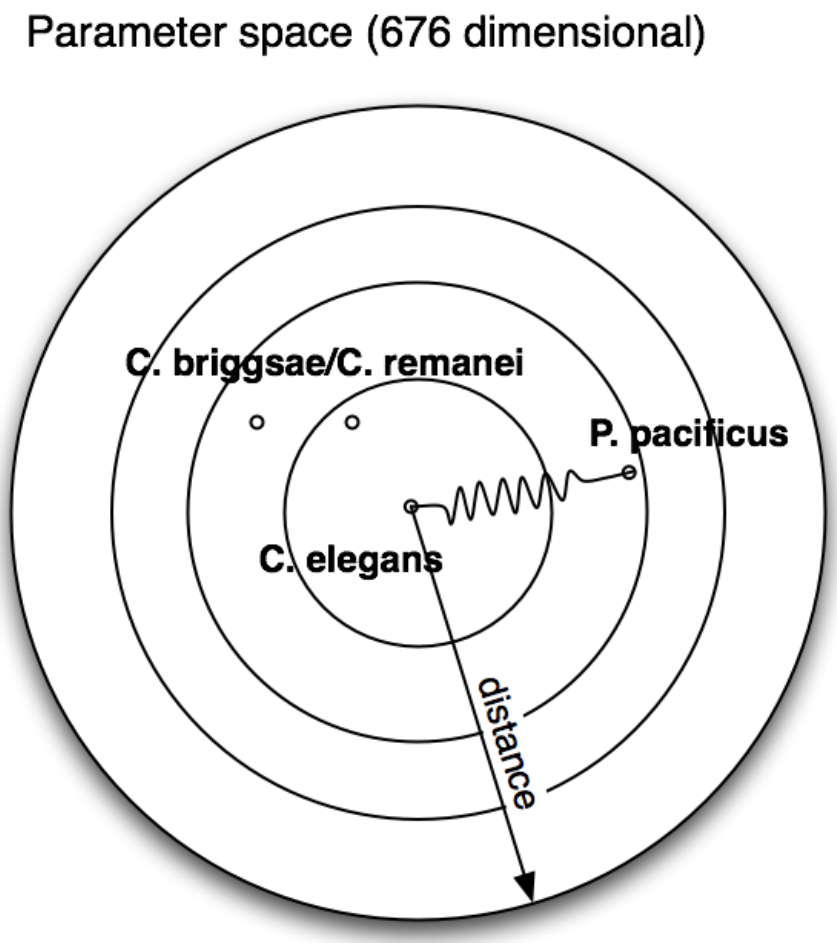
Domain Adaptation for LSL

Problem:

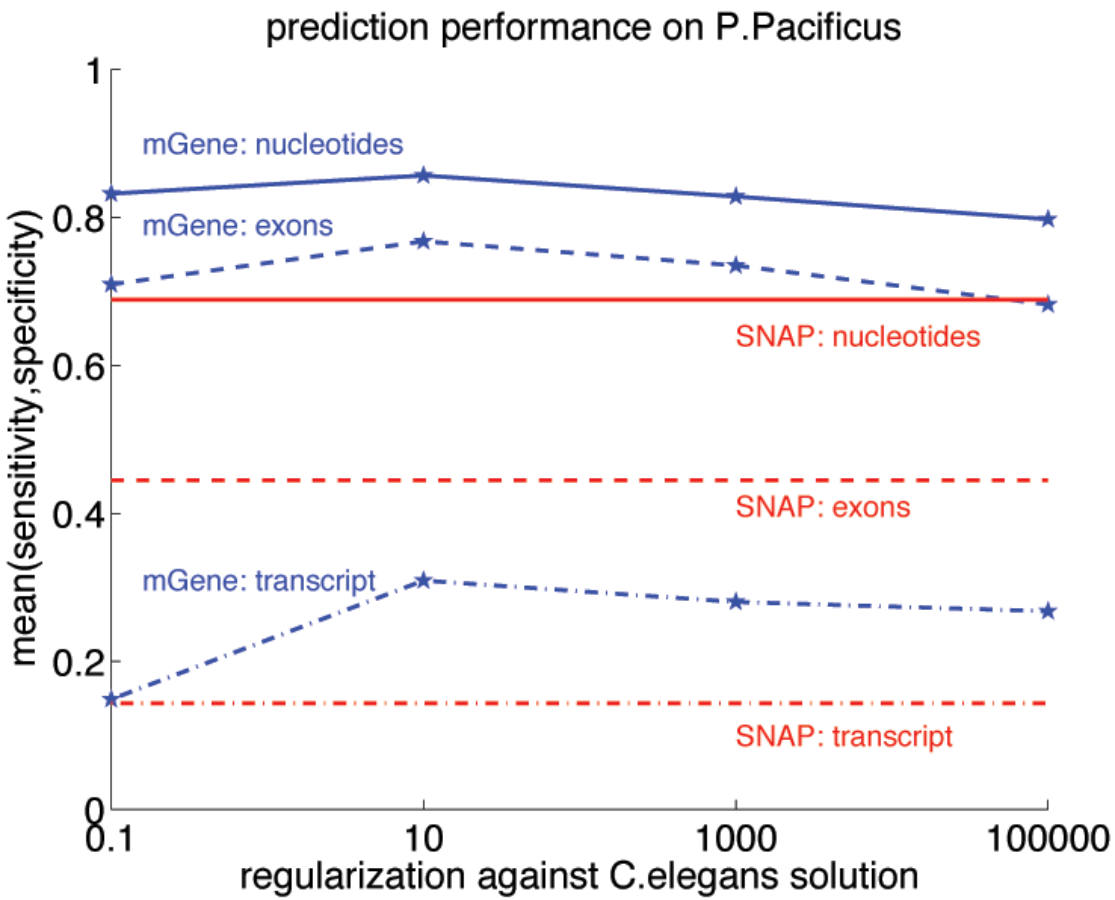
- Very little data
  - Relatively small amount of sequences available
  - Only ≈50 well analysed genes

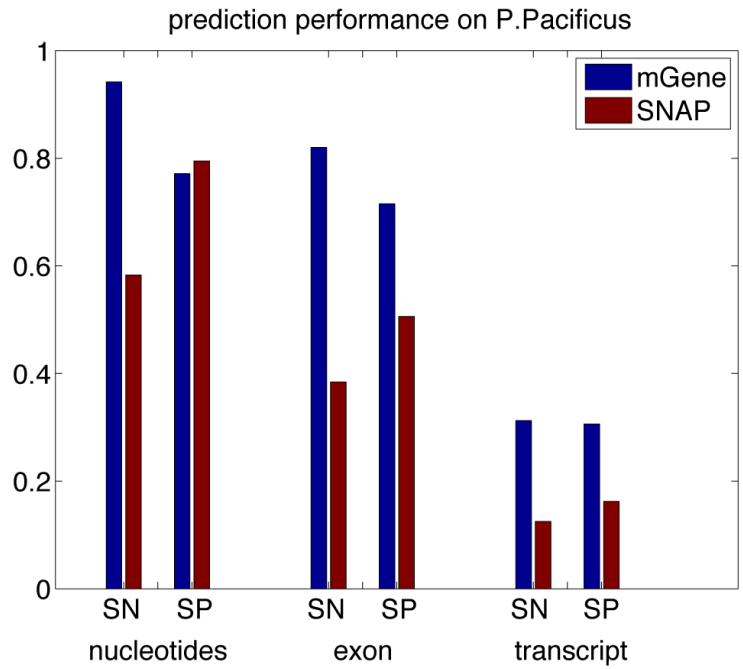
Solution:

- Exploit that *P. pacificus* is closely related to *C. elegans*
- Can use *C. elegans* signal and content sensors
- But how can *adapt* the *C. elegans* parameters for gene structure prediction?



Domain Adaptation for LSL

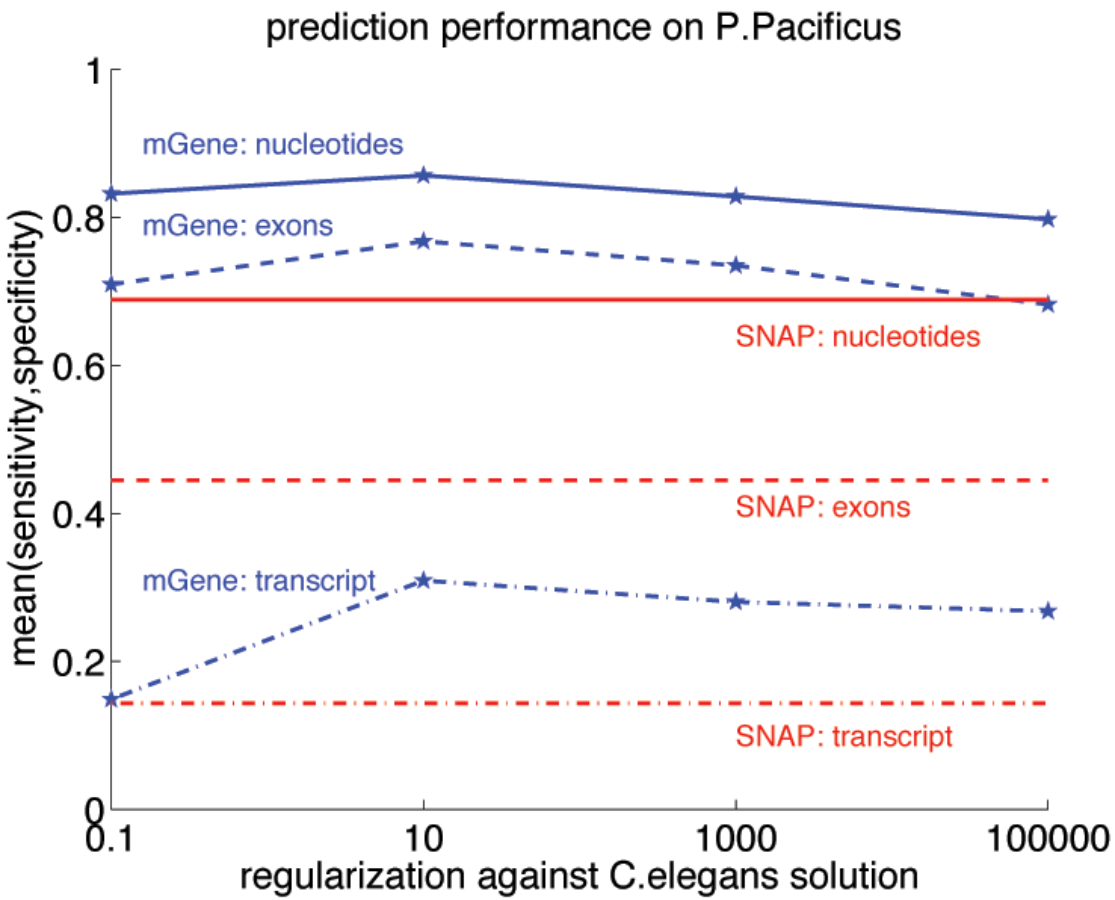


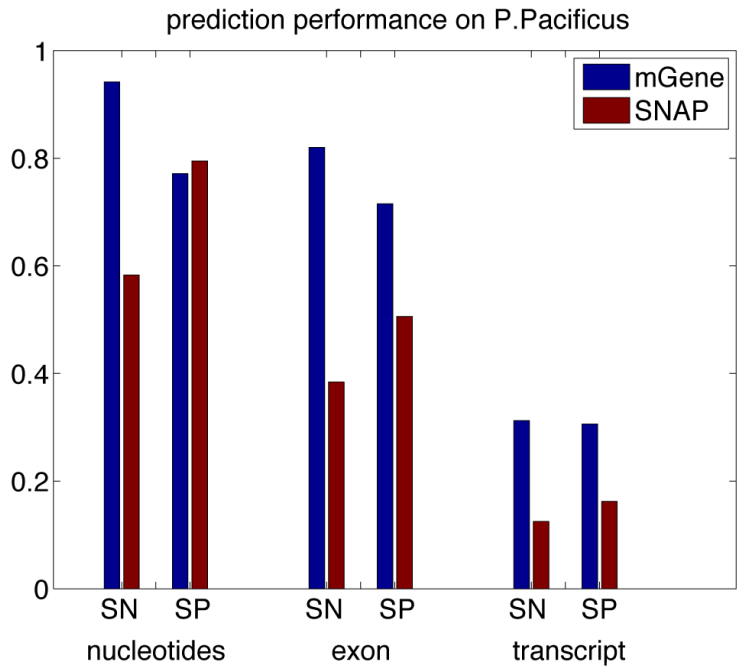


Details:

- Preliminary results!
- Signal predictions from *C. elegans*
- Training using 212 SNAP/EST gene models
- Parameter regularization against *C. elegans* solution
- Testing on 48 regions around known cDNAs ($\pm$1000nt)
- SNAP predictions provided by Christoph Dieterich (Sommer lab)

Domain Adaptation for LSL





Details:

- Preliminary results!
- Signal predictions from *C. elegans*
- Training using 212 SNAP/EST gene models
- Parameter regularization against *C. elegans* solution
- Testing on 48 regions around known cDNAs ($\pm$1000nt)
- SNAP predictions provided by Christoph Dieterich (Sommer lab)

References I

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jenn Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems 21*, Cambridge, MA, 2008. MIT Press.

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

C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4): 382–399, 2006.

Hal Daume III. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, 2007. URL `http://pub.hal3.name/#daume07easyadapt`.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 489–496. MIT Press, Cambridge, MA, 2008.

References II

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, Cambridge, MA, 2007. MIT Press.

A. Gretton, A. Smola, J. Huang, M. Schmittfull, and B. Schölkopf. Covariate shift by kernel mean matching. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, editors, *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2008.

J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, Cambridge, MA, 2007. MIT Press.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, Pavel Laskov, Klaus-Robert Müller, and Alexander Zien. Efficient and accurate lp-norm multiple kernel learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 997–1005. MIT Press, 2009.

References III

A Pires-DaSilva and RJ Sommer. Conservation of the global sex determination gene tra-1 in distantly related nematodes. *GENES & DEVELOPMENT*, 18(10):1198–1208, MAY 15 2004. ISSN 0890-9369. doi: $\{10.1101/\mathrm{gad}.293504\}$.

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

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*, September 2009.

L. Song, X. Zhang, A. Smola, A. Gretton, and B. Schölkopf. Tailoring density estimation via reproducing kernel moment matching. In *Proc. Intl. Conf. Machine Learning*, 2008.

References IV

S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch. Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, 8(Suppl. 10):S7, 2007. URL `http://www.biomedcentral.com/1471-2105/8/S10/S7`.

LD Stein, ZR Bao, D Blasiar, T Blumenthal, MR Brent, NS Chen, A Chinwalla, L Clarke, C Clee, A Coghlan, A Coulson, P D'Eustachio, DHA Fitch, LA Fulton, RE Fulton, S Griffiths-Jones, TW Harris, LW Hillier, R Kamath, PE Kuwabara, ER Mardis, MA Marra, TL Miner, P Minx, JC Mullikin, RW Plumb, J Rogers, JE Schein, M Sohrmann, J Spieth, JE Stajich, CC Wei, D Willey, RK Wilson, R Durbin, and RH Waterston. The genome sequence of Caenorhabditis briggsae: A platform for comparative genomics. *PLOS BIOLOGY*, 1(2):166+, NOV 2003. ISSN 1544-9173. doi: {10.1371/journal.pbio.0000045}.

References V

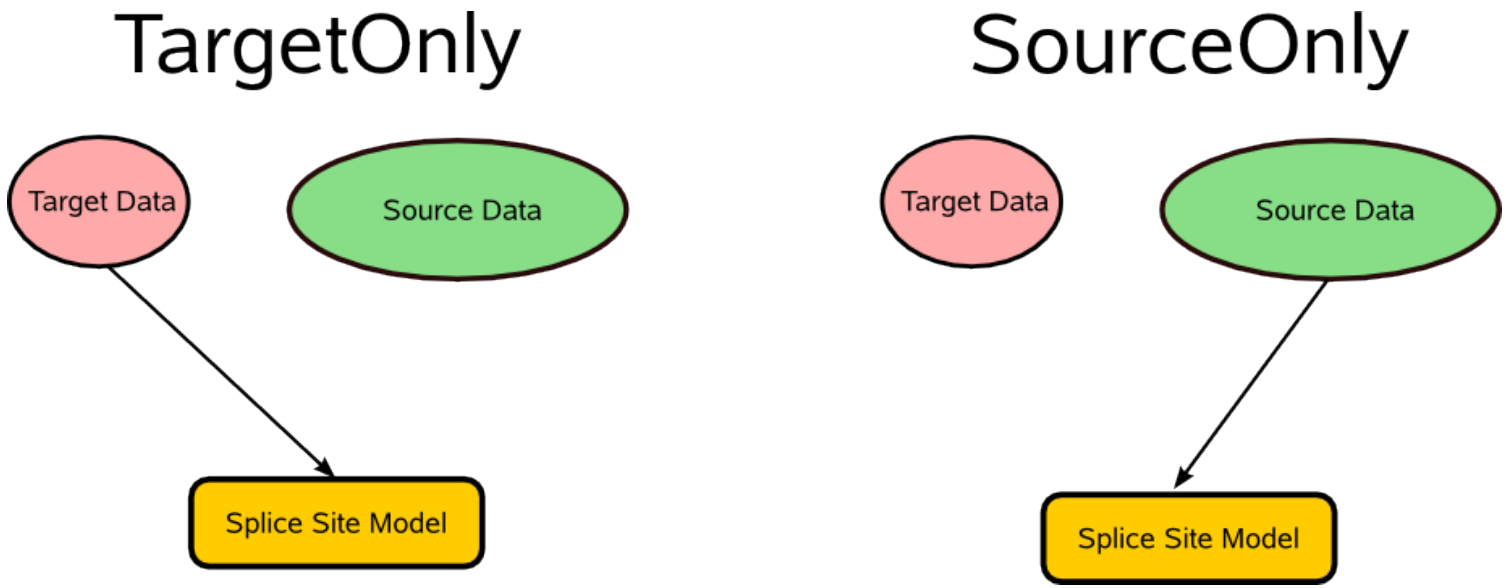
A Ureta-Vidal, L Ettwiller, and E Birney. Comparative genomics: Genome-wide analysis in metazoan eukaryotes. *NATURE REVIEWS GENETICS*, 4(4):251–262, APR 2003. ISSN 1471-0056. doi: {10.1038/nrg1043}.

C. Widmer. Domain adaptation in sequence analysis. Diplom thesis, University of Tübingen, August 2008.

C. Widmer, J.M. Leiva, Y. Altun, and G. Rätsch. Leveraging sequence classification by taxonomy-based multitask learning. In *Proc. RECOMB'10*, 2010. accepted.

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

Domain Adaptation Methods



Approach:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i$$

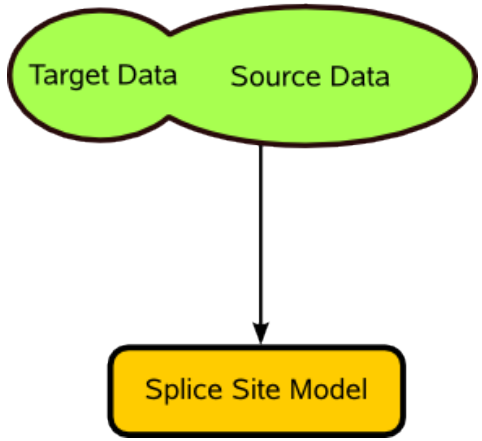
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) + \xi_i - 1 \geq 0 \quad i = 1,\ldots,n$$

$$\xi_i \geq 0 \quad i = 1,\ldots,n$$

Resulting Model: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

Domain Adaptation Methods

Combined



Idea: (e.g., Blitzer et al., 2008)

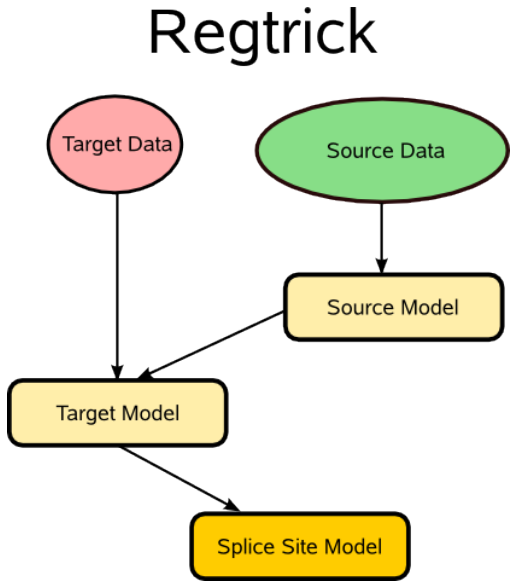
- Train on union of source and target
- Set trade-off via loss-term

Approach:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C_S \sum_{i=1}^{n} \xi_i + C_T \sum_{i=1}^{m} \xi_{n+i}$$

$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) + \xi_i - 1 \geq 0 \quad i = 1, \ldots, n+m$$

Domain Adaptation Methods



Idea: (e.g., Chelba and Acero, 2006)

- Previous solution contains prior information
- Modified regularization term

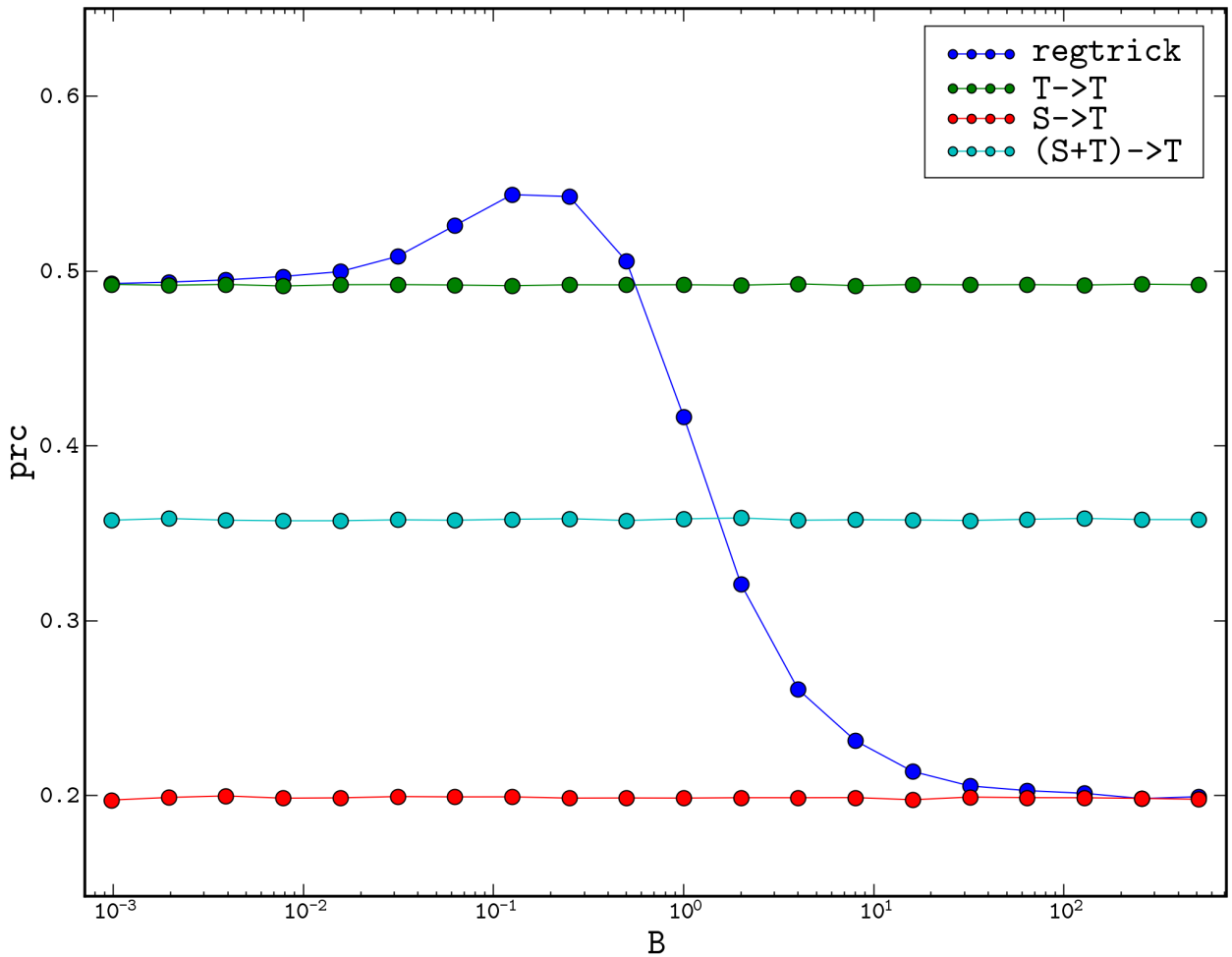
Approach:

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

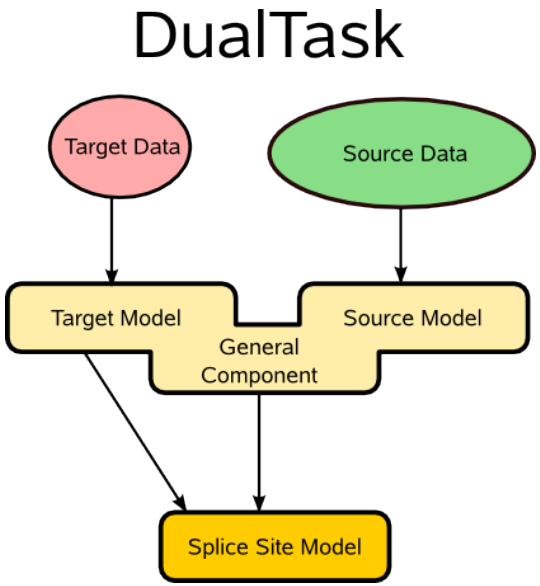
$$\text{s.t. } y_i(\mathbf{w}_T^T\mathbf{x}_i + b) + \xi_i - 1 \geq 0 \quad i = 1,\ldots,n$$

Example

Source: Verticillioides (50k), Target: Drosophila (20k)



Domain Adaptation Methods



Idea: (e.g., Caruana, 1997)

- Simultaneous optimization of both models
- Similarity between solution enforced

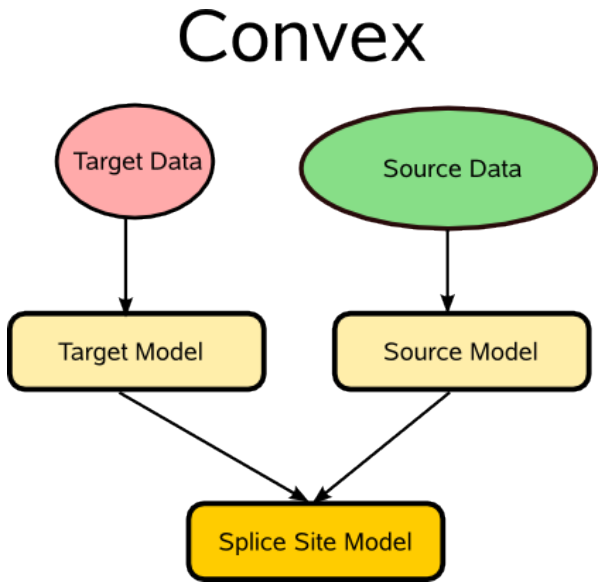
Approach:

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

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

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

Domain Adaptation Methods



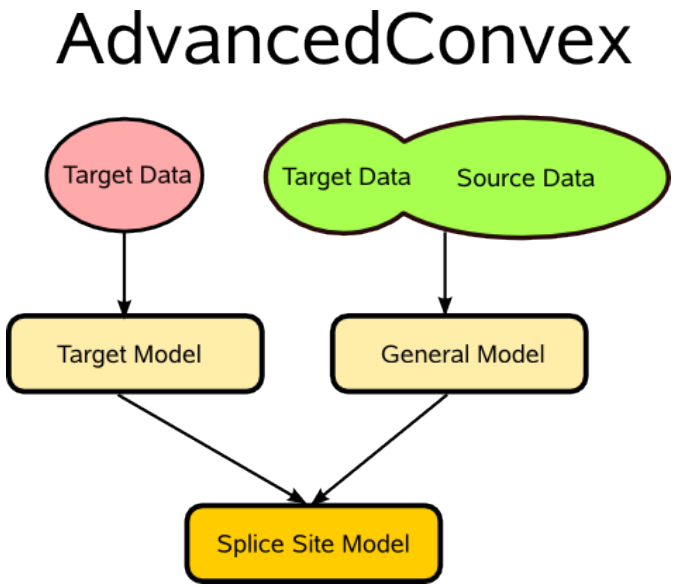
Idea: (e.g., Widmer, 2008; Schweikert et al., 2008)

- Combine trained models
- Efficient hyper-parameter optimization

Approach:

$$F(x) = \alpha f_S(x) + (1 - \alpha) f_T(x)$$

Domain Adaptation Methods



Idea: (e.g., Widmer, 2008; Schweikert et al., 2008)

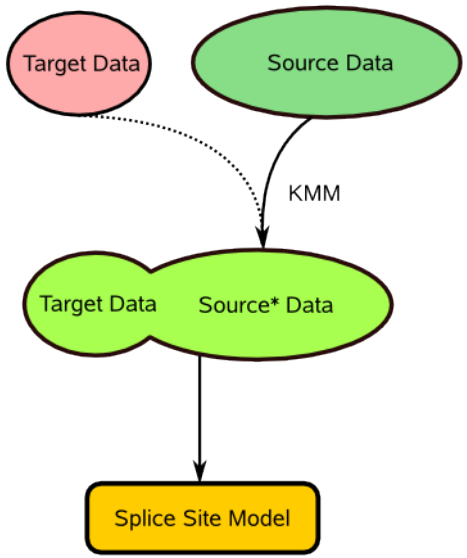
- Takes interactions between source and target examples into account
- Captures 'General' and 'Target-specific' component

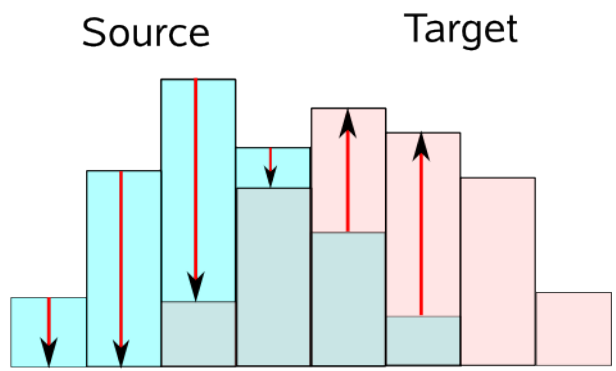
Approach:

$$F(x) = \alpha f_C(x) + (1 - \alpha) f_T(x)$$

Domain Adaption Methods

Kernel Mean Matching





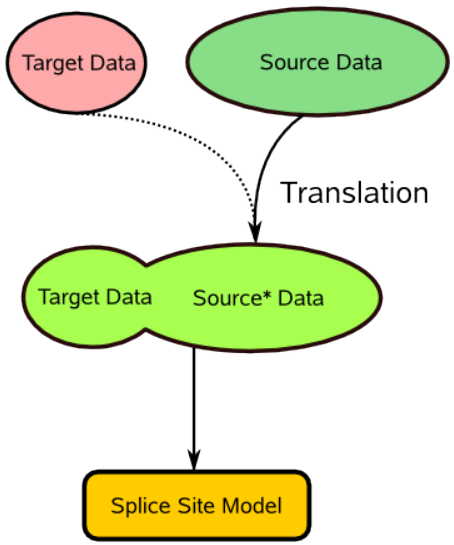
Idea:

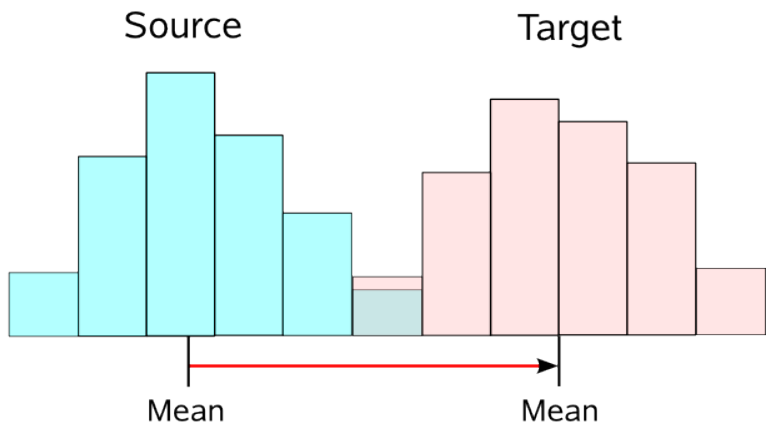
(Huang et al., 2007)

- Match mean of source and target by re-weighting examples
- Higher-order moments defined by mean when using a universal kernel

Domain Adaption Methods

Shift (KMM)





Idea:

(e.g., Widmer, 2008; Schweikert et al., 2008)

- Based on same assumption
- Mean matching via translation rather than re-weighting (for each class separately)