Statistical Relational Learning: Some Applications to Bioinformatics

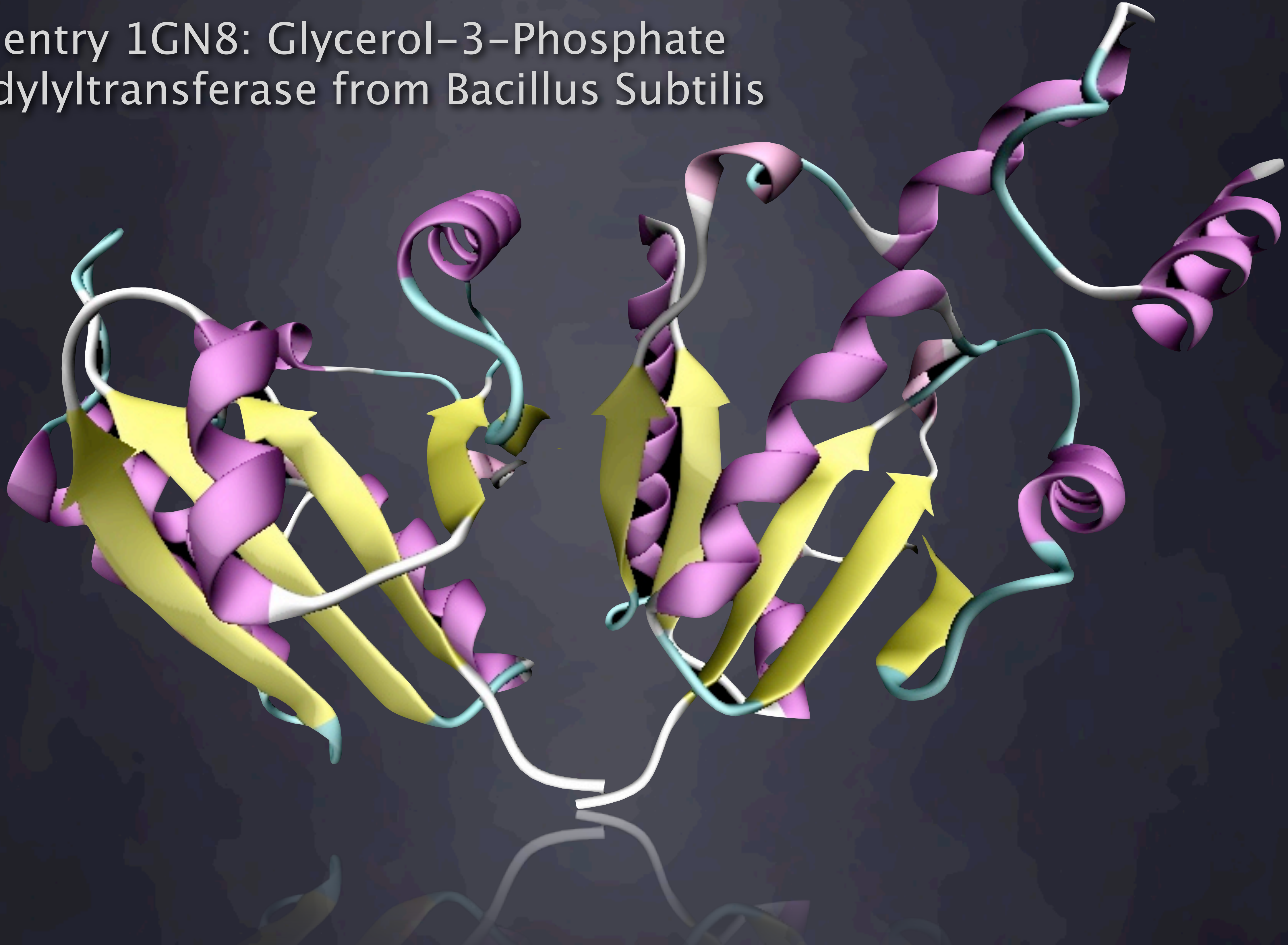
Paolo Frasconi
Machine Learning and Neural Networks Group
Università degli Studi di Firenze, Italy

http://www.dsi.unifi.it/~paolo/

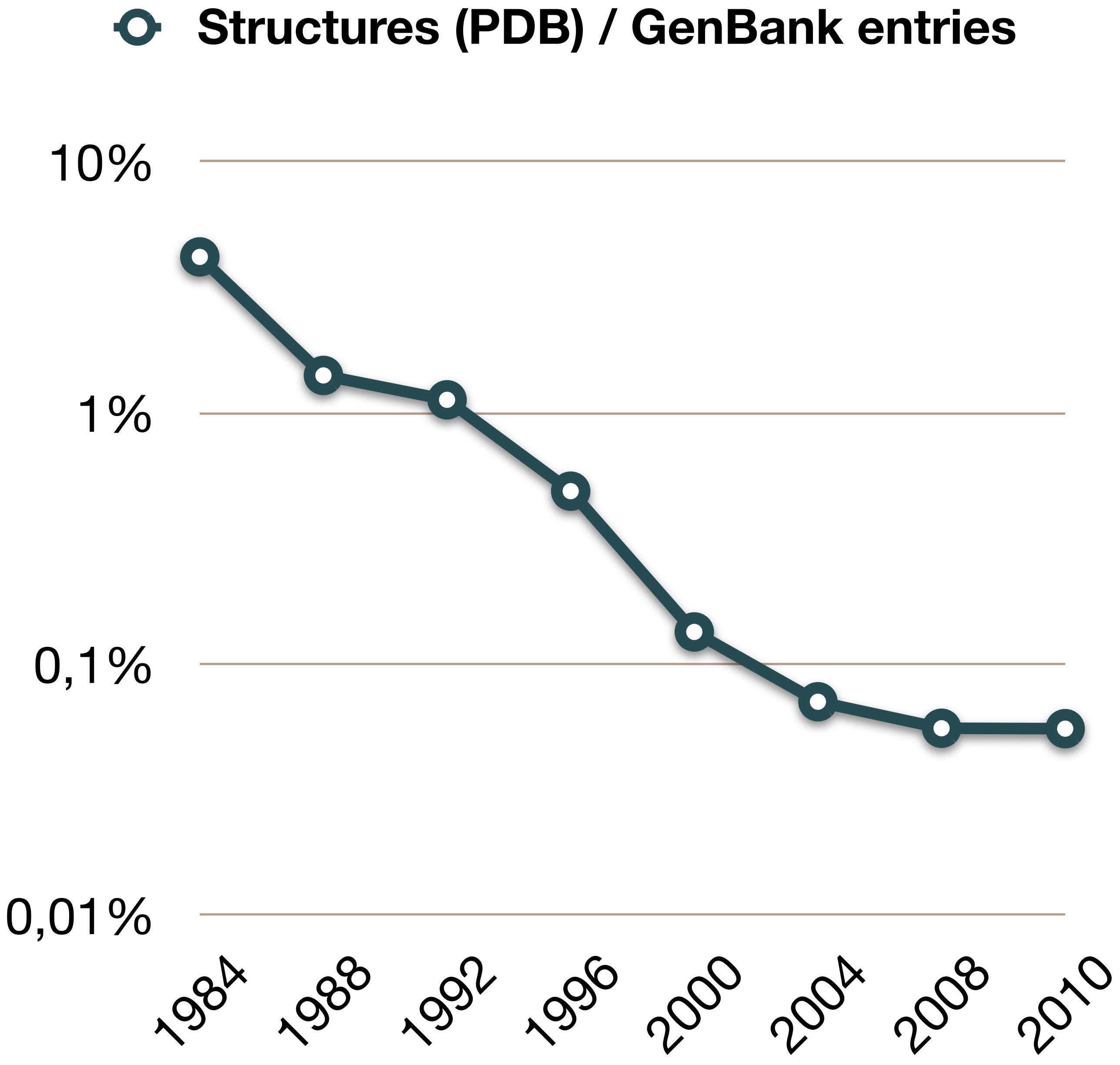
Colloquia@IASI - 15.01.2010

Protein structure

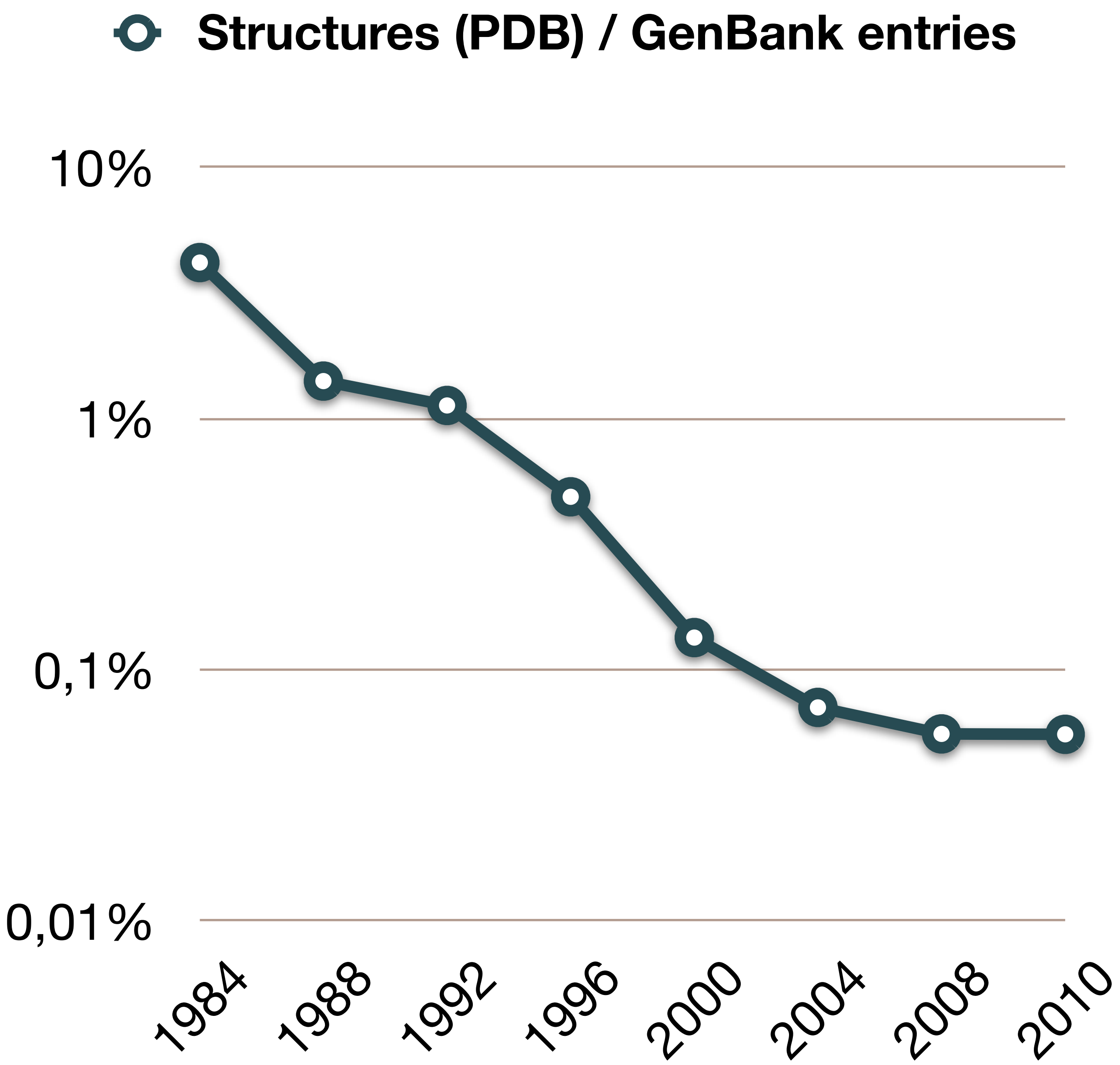
PDB entry 1GN8: Glycerol-3-Phosphate Cytidylyltransferase from Bacillus Subtilis

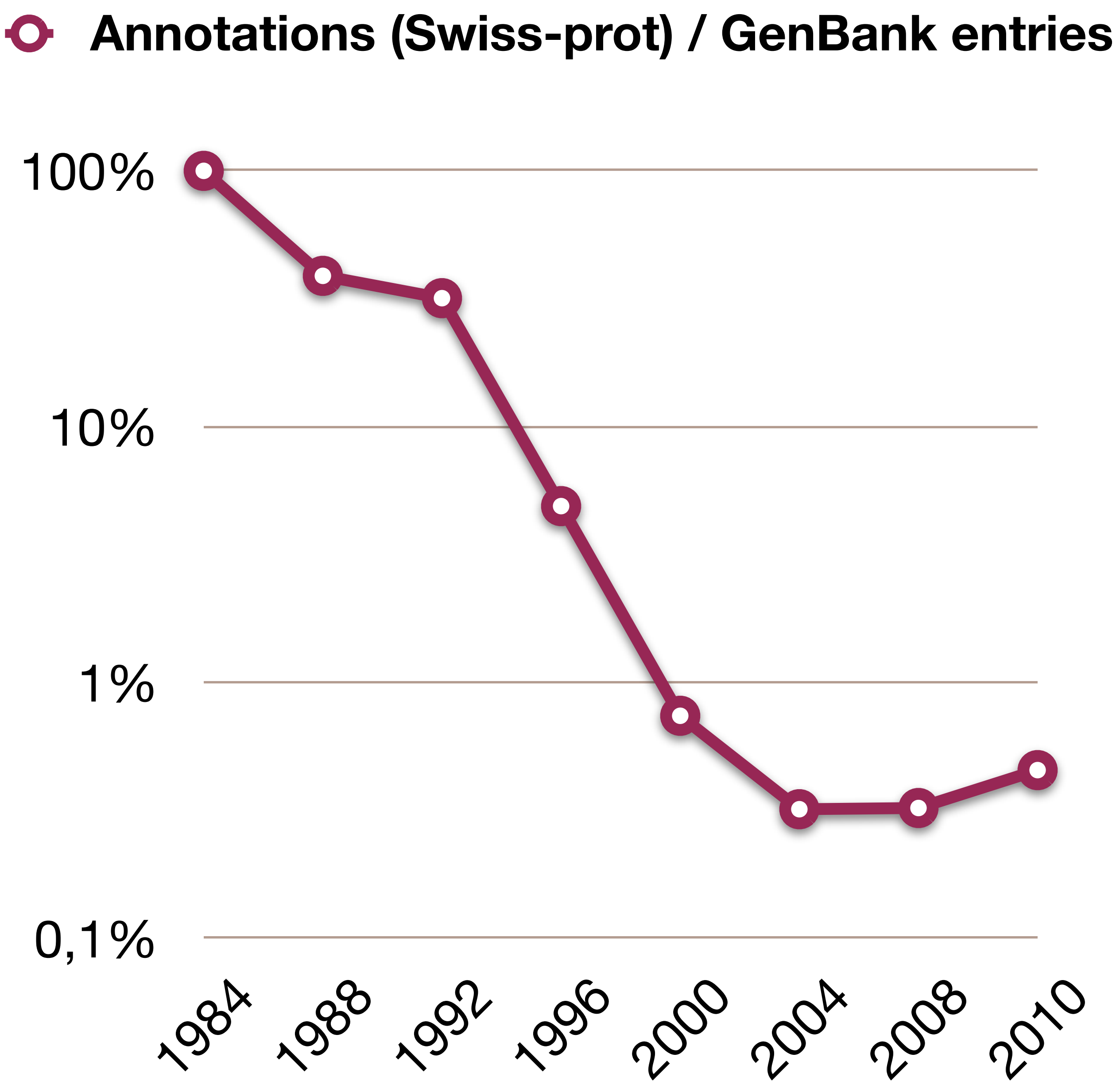


The annotation/determination crisis



The annotation/determination crisis





Prediction of ß-partners

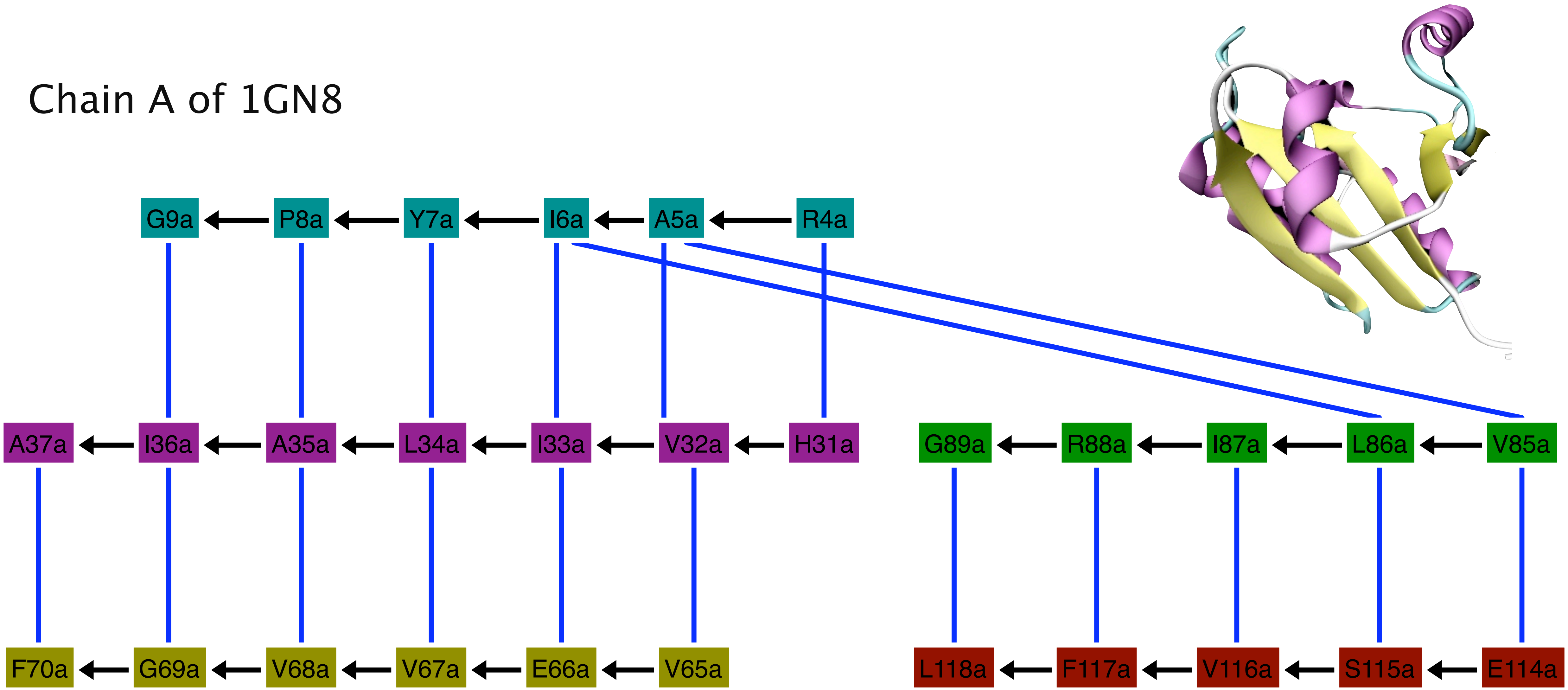
- ß–sheets: very common secondary structure of proteins
    - occur in ~ 85% of experimentally know structures
    - 15% of know structures entirely consist of ß–structures

A large fraction of distant contacts in contact maps involve two ß–residues

- other prominent cases: disulfide bridges and metal binding sites

ß-partners: a link prediction task

Chain A of 1GN8



G9a ← P8a ← Y7a ← I6a ← A5a ← R4a

A37a ← I36a ← A35a ← L34a ← I33a ← V32a ← H31a

G89a ← R88a ← I87a ← L86a ← V85a

F70a ← G69a ← V68a ← V67a ← E66a ← V65a

L118a ← F117a ← V116a ← S115a ← E114a

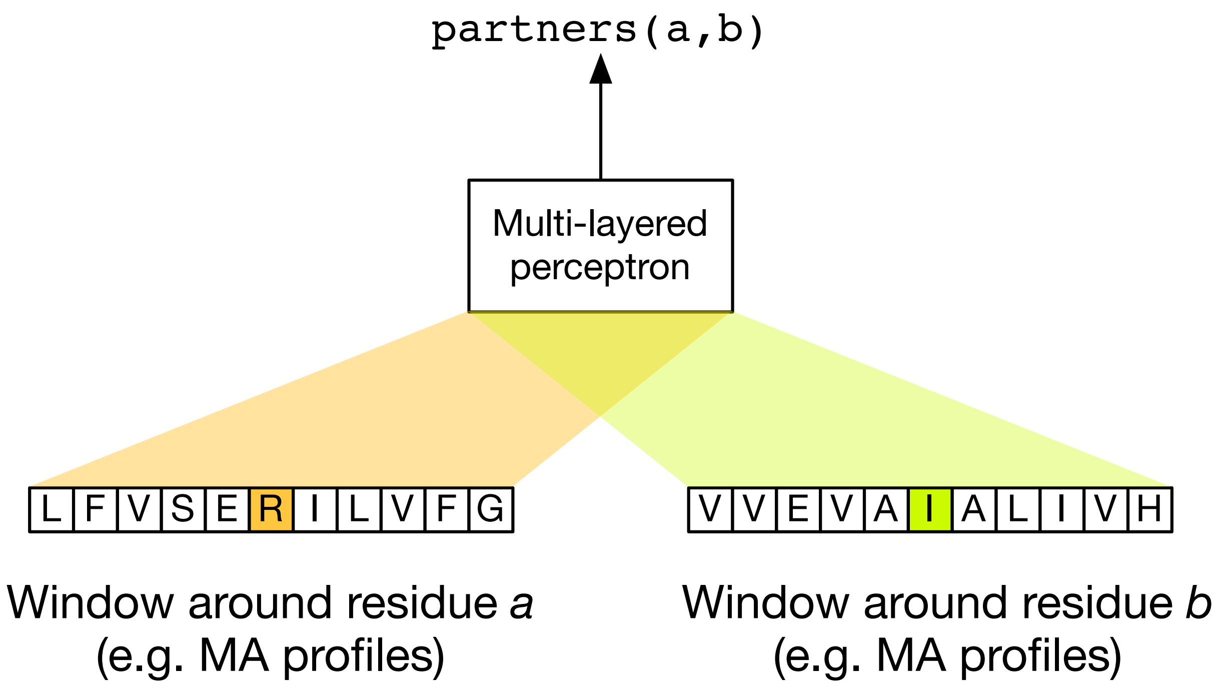
Formalization as a supervised learning task

- Input:
  - protein sequence (possibly enriched with evolutionary information)
  - secondary structure assignment to each residue (can be predicted by other machine learning tools)
- Output:
  - the relation `partner(a,b)` where `a` and `b` are ß-residues

Classic (i.i.d.) view of supervised learning

- Data comes as (x,y) pairs:
  - every pair generated independently
  - all pairs identically distributed, i.e. generated from the same (fixed but unknown) distribution
- In practice, these assumptions can be very well violated, e.g.:
  - can be safe to assume that proteins are independent
  - however links between ß-residues are definitely not!
  - the “right” setting is sometimes called “structured output learning”, i.e. y (not just x) is a structured object made of interdependent atomic variables

Early approaches


partners(a,b)
Multi-layered perceptron
L F V S E R I L V F G
V V E V A I A L I V H
Window around residue *a* (e.g. MA profiles)
Window around residue *b* (e.g. MA profiles)

- Plain neural networks (Baldi et al. 2000):
  - cast link prediction into binary classification of pairs
  - exaggeratedly imbalanced data set: 826 chains yield 37,000 positive examples and 44 million negative examples

State-of-the-art: BetaPro (Cheng & Baldi, 2005)

State-of-the-art: BetaPro (Cheng & Baldi, 2005)

- Secondary structure of residues is given

State-of-the-art: BetaPro (Cheng & Baldi, 2005)

- Secondary structure of residues is given
- 2D Recursive Neural Networks (2D–RNN)
  - 2D grid, target is the adjacency matrix of the β–partners graph
  - local inputs: 2 windows centered around residues a and b
  - smart compromise between iid and collective classification

State-of-the-art: BetaPro (Cheng & Baldi, 2005)

- Secondary structure of residues is given
- 2D Recursive Neural Networks (2D-RNN)
  - 2D grid, target is the adjacency matrix of the β-partners graph
  - local inputs: 2 windows centered around residues a and b
  - smart compromise between iid and collective classification
- Collective assignment is done via a non-adaptive post-processor that enforces some physical

Background knowledge

Background knowledge

- Partnership is symmetric and irreflexive

Background knowledge

- Partnership is symmetric and irreflexive
- A β-residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain

Background knowledge

- Partnership is symmetric and irreflexive
- A β–residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti–parallel)

Background knowledge

- Partnership is symmetric and irreflexive
- A β–residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti–parallel)
- Connection gaps in a given strand are rare

Background knowledge

- Partnership is symmetric and irreflexive
- A β–residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti–parallel)
- Connection gaps in a given strand are rare
- β–hairpins: two strands separated by <6 residues that include a proline or glicine are very often anti–parallel

Background knowledge

- Partnership is symmetric and irreflexive
- A β–residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti–parallel)
- Connection gaps in a given strand are rare
- β–hairpins: two strands separated by <6 residues that include a proline or glicine are very often anti–parallel
- Two strands surrounding a helix are very often parallel

Background knowledge

- Partnership is symmetric and irreflexive
- A β–residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti–parallel)
- Connection gaps in a given strand are rare
- β–hairpins: two strands separated by <6 residues that include a proline or glicine are very often anti–parallel
- Two strands surrounding a helix are very often parallel
- Residues in the same strand are never partners

Background knowledge

- Partnership is symmetric and irreflexive
- A β-residue is connected to 0 (rare) or 1 or 2 partners, very often in the same chain
- Adjacent strands can either run in the same direction (parallel) or opposite direction (anti-parallel)
- Connection gaps in a given strand are rare
- β-hairpins: two strands separated by <6 residues that include a proline or glicine are very often anti-parallel
- Two strands surrounding a helix are very often parallel
- Residues in the same strand are never partners
- No crossing edges: e.g. if (a,b) and (a+1,b+1) are partners, then (a+2,b-1) can't be partners

Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL

Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL
- Use first-order-logic as the underlying language

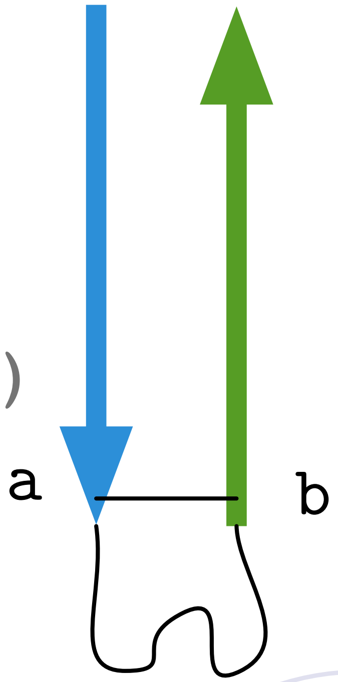
Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL
- Use first-order-logic as the underlying language
- Formulas are **weighted**, allowing uncertainty in the model

Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL
- Use first-order-logic as the underlying language
- Formulas are **weighted**, allowing uncertainty in the model
- E.g. simplified rule for β-hairpins:

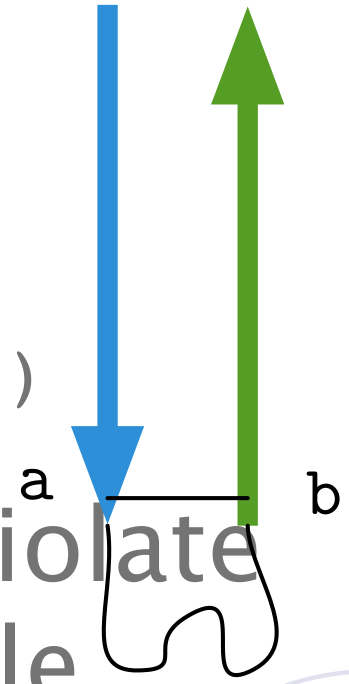
```
0.5: Last(a,s) ∧ First(b,r)
     ∧ Antiparallel(s,r) ∧ |s|=|r| ⇒ Partners(a,b)
```



Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL
- Use first-order-logic as the underlying language
- Formulas are **weighted**, allowing uncertainty in the model
- E.g. simplified rule for β-hairpins:

```
0.5: Last(a,s) ∧ First(b,r)
     ∧ Antiparallel(s,r) ∧ |s|=|r| ⇒ Partners(a,b)
```

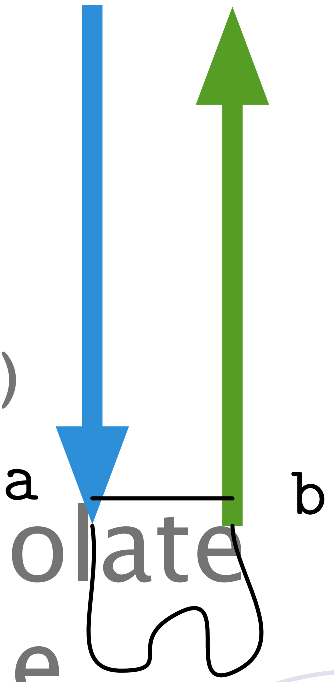


- A large **weight** makes interpretations that violate the formula less probable but not impossible

Markov logic (Richardson & Domingos, 2005)

- One of many possible approaches in SRL
- Use first-order-logic as the underlying language
- Formulas are **weighted**, allowing uncertainty in the model
- E.g. simplified rule for β-hairpins:

```
0.5: Last(a,s) ∧ First(b,r)
     ∧ Antiparallel(s,r) ∧ |s|=|r| ⇒ Partners(a,b)
```



- A large **weight** makes interpretations that violate the formula less probable but not impossible
- Weights are learned from data

Example: Friends & Smokers (from P. Domingos)

```
1.5: Smokes(x) ⇒ Cancer(x)

1.1: Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))
```

Example: Friends & Smokers (from P. Domingos)

1.5: Smokes(x) ⇒ Cancer(x)

1.1: Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))

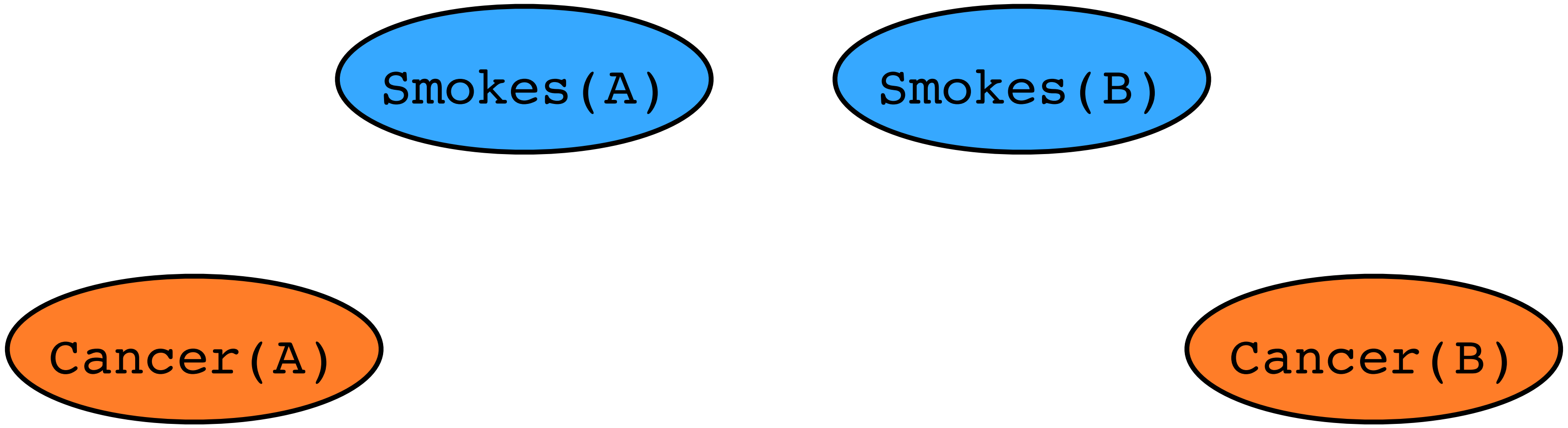
Domain constants: **Anna** (A) and **Bob** (B)

Example: Friends & Smokers (from P. Domingos)

1.5: Smokes(x) ⇒ Cancer(x)

1.1: Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))

Domain constants: **Anna** (A) and **Bob** (B)



Smokes(A)

Smokes(B)

Cancer(A)

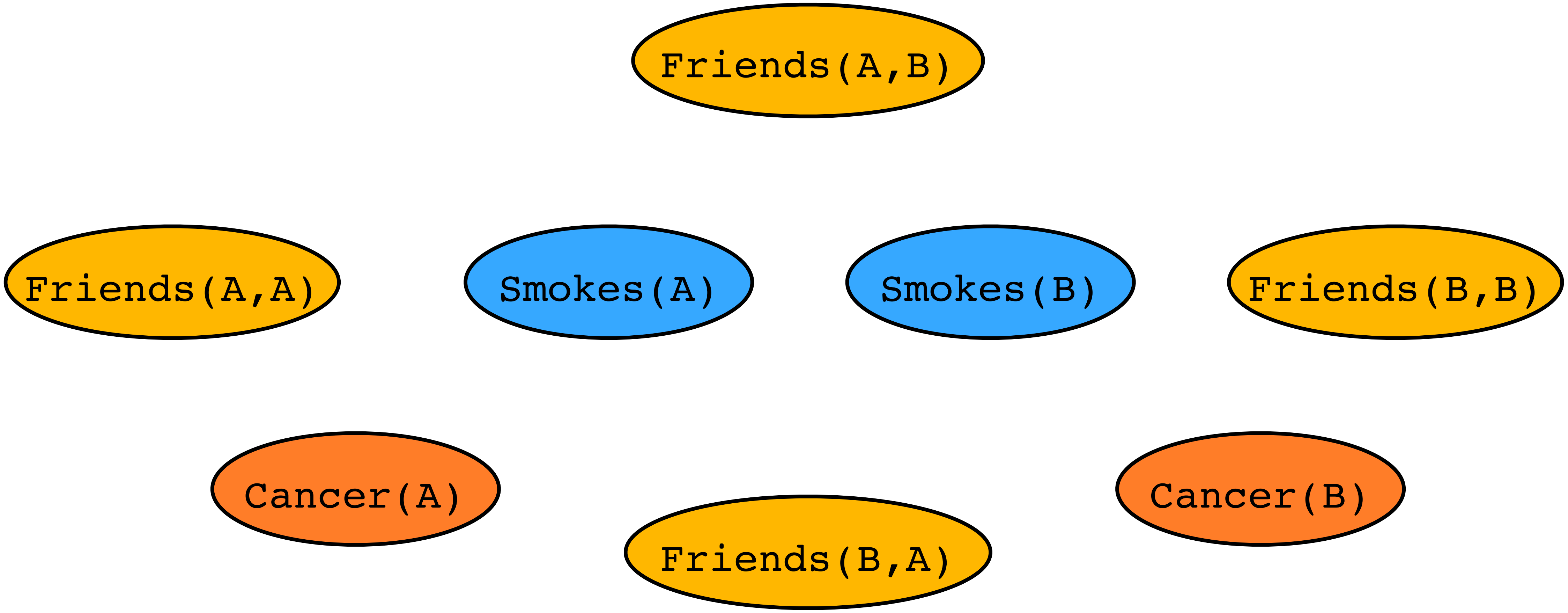
Cancer(B)

Example: Friends & Smokers (from P. Domingos)

1.5: Smokes(x) ⇒ Cancer(x)

1.1: Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))

Domain constants: **Anna** (A) and **Bob** (B)



Friends(A,B)

Friends(A,A)

Smokes(A)

Smokes(B)

Friends(B,B)

Cancer(A)

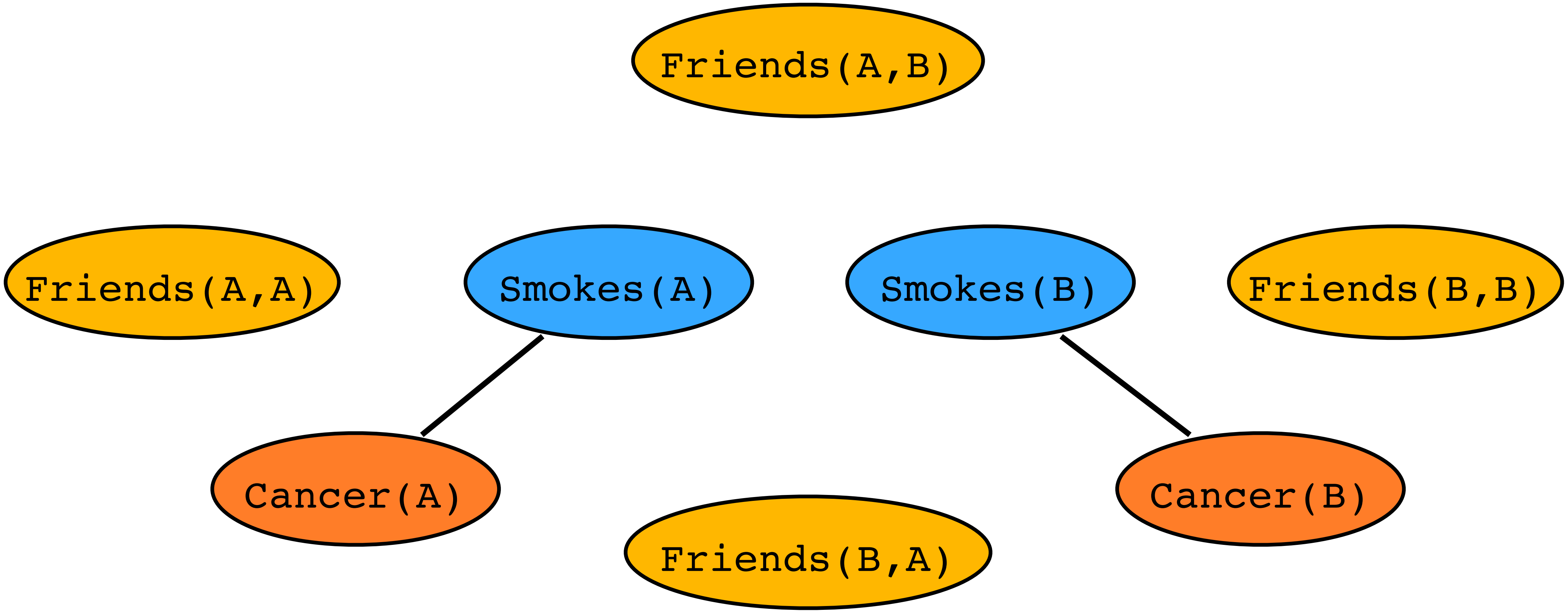
Friends(B,A)

Cancer(B)

Example: Friends & Smokers (from P. Domingos)

```
1.5: Smokes(x) ⇒ Cancer(x)

1.1: Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))
```

Domain constants: **Anna** (A) and **Bob** (B)

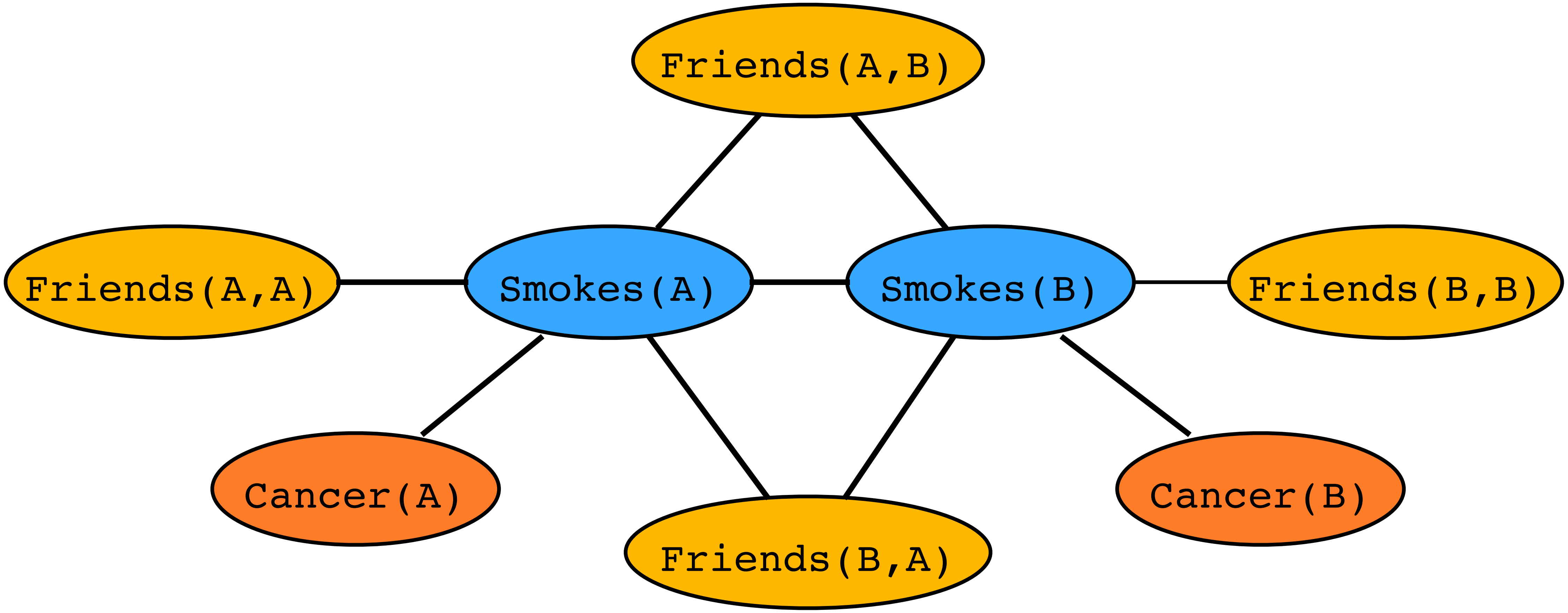


Example: Friends & Smokers (from P. Domingos)

1.5: `Smokes(x) ⇒ Cancer(x)`

1.1: `Friends(x,y) ⇒ (Smokes(x) ⇔ Smokes(y))`

Domain constants: **Anna** (`A`) and **Bob** (`B`)



Markov logic

- Discriminative learning: model p(y|x) since x always observed

- In this setting, an MLN defines a distribution over query interpretations, conditioned on evidence:

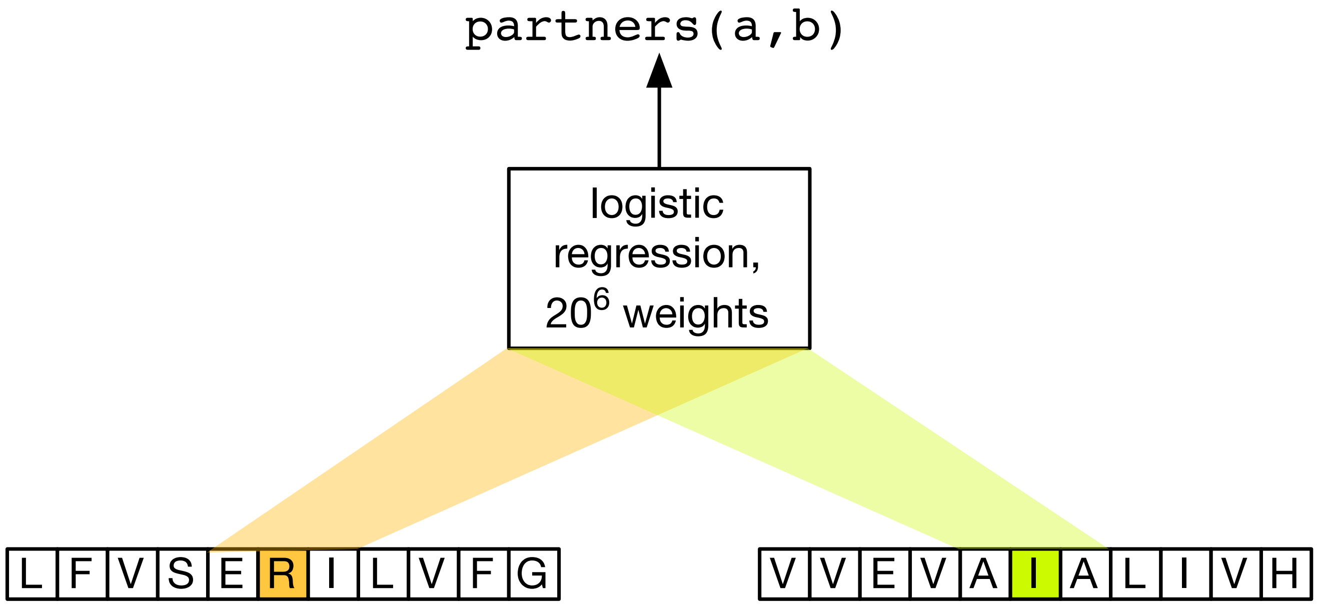
$$P(Y = y|X = x) = \frac{1}{Z_x} \exp\left(\sum_{F_i \in \mathcal{F}_y} w_i n_i(x, y)\right)$$

- w_i is the weight of formula F_i and $n_i(x,y)$ the number of true groundings of F_i in world (x,y)

Basic rule: partnership depends on amino acid windows

Toy example: propositional rule with only 3 residues

```
Res(+la,a-1) ∧ Res(+ca,a) ∧ Res(+ra,a+1)
Res(+lb,b-1) ∧ Res(+cb,b) ∧ Res(+ra,b+1) ⇒ Partners(a,b)
```



partners(a,b)

logistic regression, 20^6 weights

L F V S E R I L V F G

V V E V A I A L I V H

Window around residue *a*

Window around residue *b*

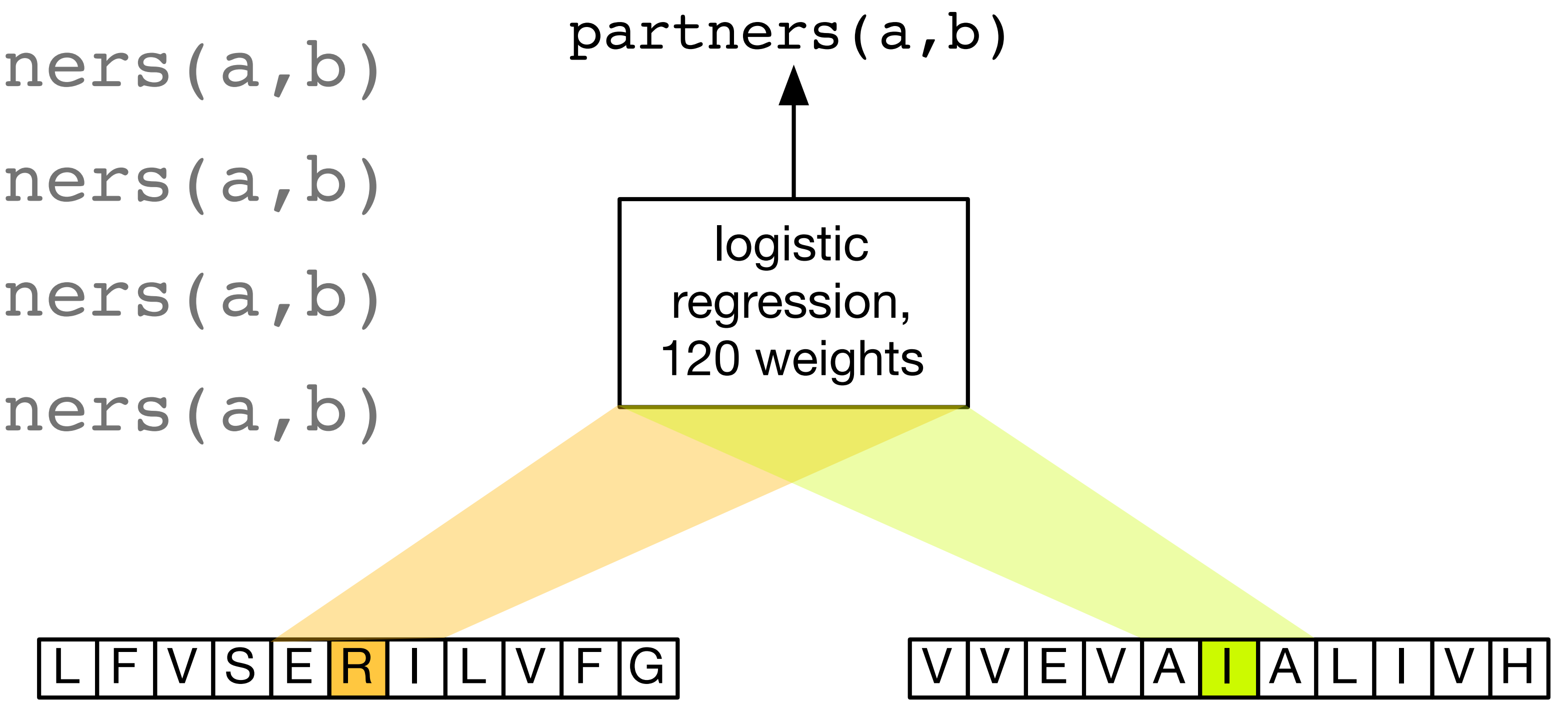
Clearly does not scale up with the window size!

Syntax note The + symbol means: *“generate multiple formulas (with distinct weights) by replacing the prefixed variable with all possible domain constants”*

Basic rule: partnership depends on amino acid windows

Could split the formula into separate formulae, one for each residue position:

```
Res(+la,a-1) ⇒ Partners(a,b)
Res(+ca,a)   ⇒ Partners(a,b)
Res(+ra,a+1) ⇒ Partners(a,b)
Res(+lb,b-1) ⇒ Partners(a,b)
Res(+cb,b)   ⇒ Partners(a,b)
Res(+ra,b+1) ⇒ Partners(a,b)
```



Window around residue *a*

Window around residue *b*

Small model size but this is a linear model in the amino acid features

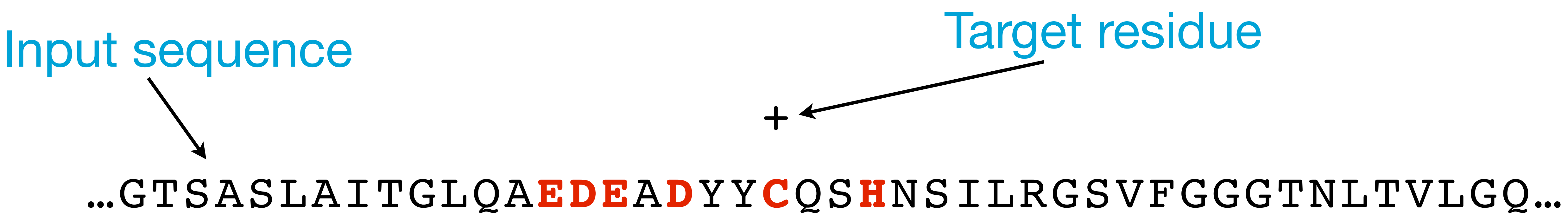
Additional issue: multiple alignment profiles

Additional issue: multiple alignment profiles

Input sequence

…GTSASLAITGLQA**EDE**A**D**YY**C**QS**H**NSILRGSVFGGGTNLTVLGQ…

Additional issue: multiple alignment profiles



Input sequence

Target residue

+

…GTSASLAITGLQA**EDE**A**D**YY**C**QS**H**NSILRGSVFGGGTNLTVLGQ…

Additional issue: multiple alignment profiles

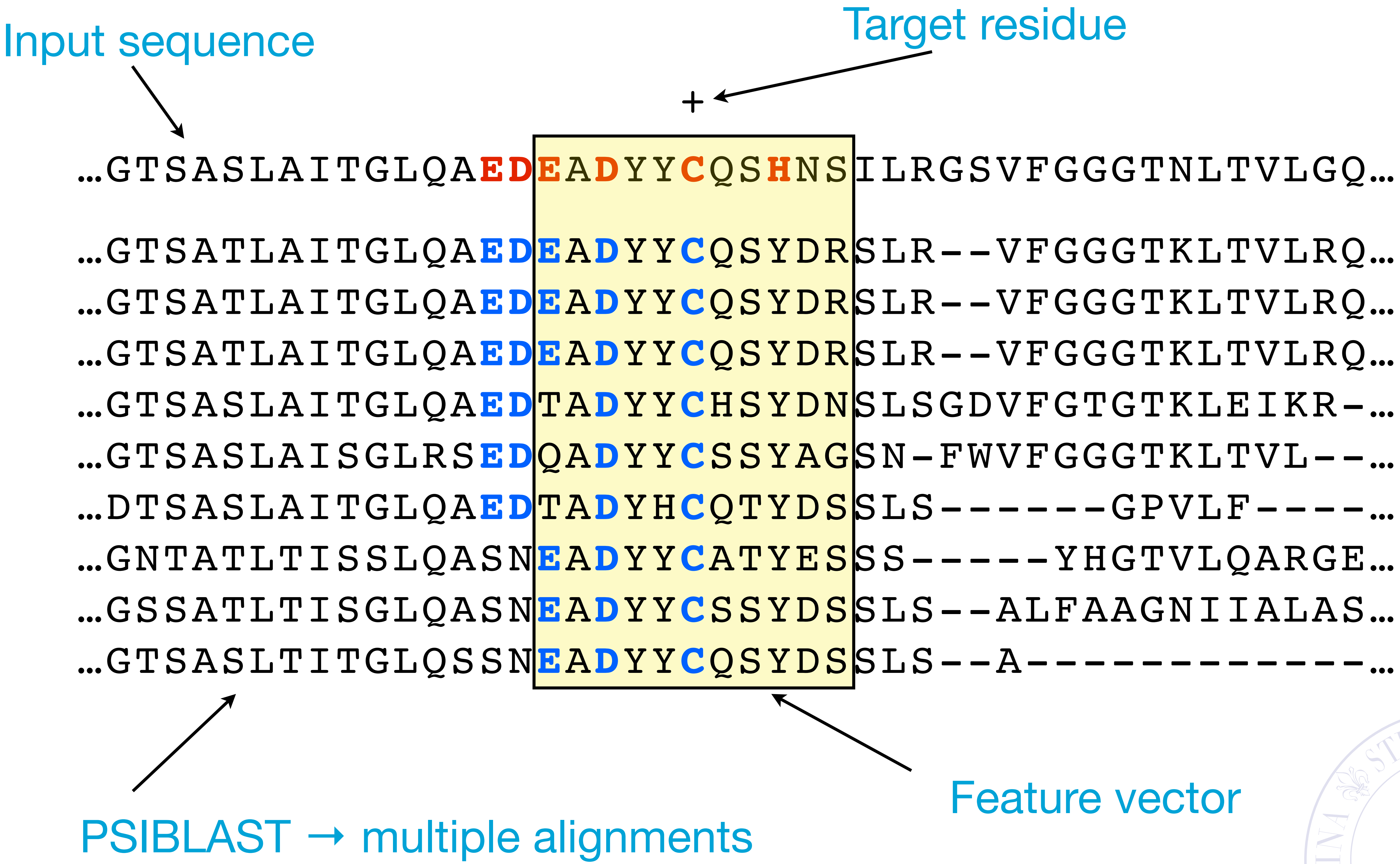
Input sequence

Target residue

```
                     +
…GTSASLAITGLQAEDEADYYCQSHNSILRGSVFGGGTNLTVLGQ…

…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSASLAITGLQAEDTADYYCHSYDNSLSGDVFGTGTKLEIKR-…
…GTSASLAISGLRSEDQADYYCSSYAGSN-FWVFGGGTKLTVL--…
…DTSASLAITGLQAEDTADYHCQTYDSSLS------GPVLF----…
…GNTATLTISSLQASNEADYYCATYESSS-----YHGTVLQARGE…
…GSSATLTISGLQASNEADYYCSSYDSSLS--ALFAAGNIIALAS…
…GTSASLTITGLQSSNEADYYCQSYDSSLS--A------------…
```

PSIBLAST → multiple alignments

Additional issue: multiple alignment profiles



Input sequence

Target residue

```
                                    +
…GTSASLAITGLQAEDEADYYCQSHNSILRGSVFGGGTNLTVLGQ…

…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSATLAITGLQAEDEADYYCQSYDRSLR--VFGGGTKLTVLRQ…
…GTSASLAITGLQAEDTADYYCHSYDNSLSGDVFGTGTKLEIKR-…
…GTSASLAISGLRSEDQADYYCSSYAGSN-FWVFGGGTKLTVL--…
…DTSASLAITGLQAEDTADYHCQTYDSSLS------GPVLF----…
…GNTATLTISSLQASNEADYYCATYESSS-----YHGTVLQARGE…
…GSSATLTISGLQASNEADYYCSSYDSSLS--ALFAAGNIIALAS…
…GTSASLTITGLQSSNEADYYCQSYDSSLS--A------------…
```

PSIBLAST → multiple alignments

Feature vector

MLN with grounding specific weights (Lippi & Frasconi 2009)

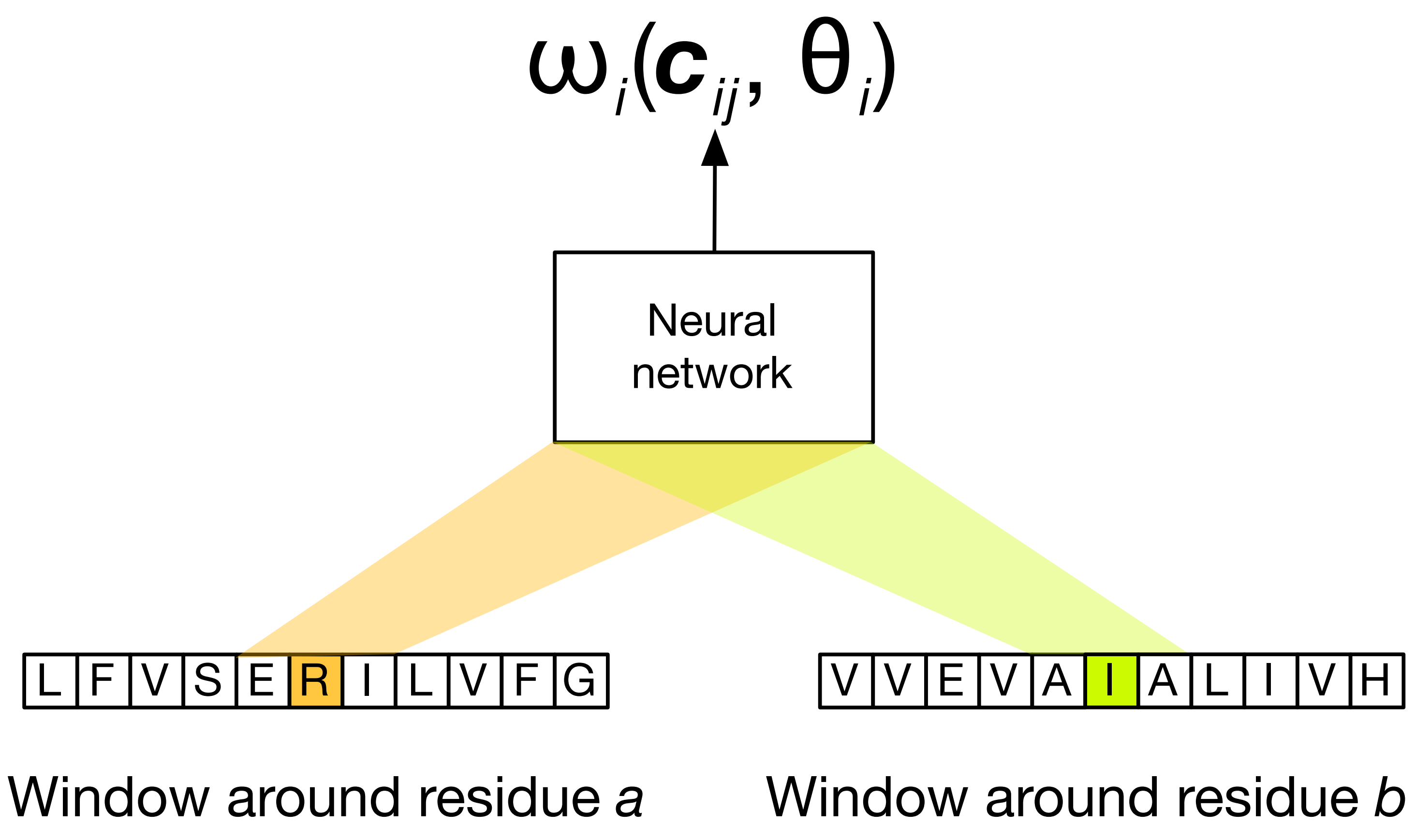
- Choose a set of dependency variables in formula F_i
- Let $\mathbf{c}_{ij}$ be the j-th ground configuration for these variables in F_i
- Let the weight depend on these specific groundings:

$$P(Y=y|X=x)=\frac{1}{Z_x}\exp\left(\sum_{F_i\in\mathcal{F}_y}\sum_{j}\omega_i(\mathbf{c}_{ij},\theta_i)n_{ij}(x,y)\right)$$

- where ω_i is a function of the ground configuration and some optional parameters θ_i

MLN with grounding specific weights

- $\omega_i(\mathbf{c}_{ij}, \theta_i)$ can be implemented e.g. by a neural network with weights θ_i, taking as input an encoding of the grounding $\mathbf{c}_{ij}$
- Adding multiple alignment profiles becomes very easy
- Similar ideas combining neural networks and conditional random fields recently exploited by Peng et al (NIPS 2009)


$\omega_i(\boldsymbol{c}_{ij}, \theta_i)$
Neural network
L F V S E R I L V F G
V V E V A I A L I V H
Window around residue *a*
Window around residue *b*

Inference and learning

- Inference: same as MLN
  - MC-SAT for conditional probabilities P(query | evidence)
  - (lazy) MaxWalkSAT for MAP inference (most likely query given evidence)
- Learning: gradient descent
  - after gradients of the log-likelihood with respect to weights have been computed, use them as delta error for backpropagation in the neural network
  - use stochastic gradient descent with mini-batches associated with connected components of the relational domain (e.g. individual protein chains)

Learning by gradient descent

- Neural networks gradient:

$$\frac{\partial P_\omega(y|x)}{\partial \theta_k} = \frac{\partial P_\omega(y|x)}{\partial \omega_i} \frac{\partial \omega_i}{\partial \theta_k}$$

Learning by gradient descent

- Neural networks gradient:

$$\frac{\partial P_\omega(y|x)}{\partial \theta_k} = \frac{\partial P_\omega(y|x)}{\partial \omega_i} \frac{\partial \omega_i}{\partial \theta_k}$$

- In the case of full (MC-SAT) inference, the first term is the difference between evidence and inference counts,

$$n_i(y,x) - E_\omega[n_i(y,x)]$$

Learning by gradient descent

- Neural networks gradient:

$$\frac{\partial P_\omega(y|x)}{\partial \theta_k} = \frac{\partial P_\omega(y|x)}{\partial \omega_i}\frac{\partial \omega_i}{\partial \theta_k}$$

- In the case of full (MC-SAT) inference, the first term is the difference between evidence and inference counts,

$$n_i(y,x) - E_\omega[n_i(y,x)]$$

- The second term is computed by backpropagation

MAP inference and active learning

- MAP approximation of gradient:

$$\frac{\partial P_\omega(y|x)}{\partial \omega_i} = n_i(x, y) - E_\omega[n_i(x, y)] \approx n_i(x, y) - n_i(x, y^*_\omega)$$

where y* is the MAP solution

MAP inference and active learning

- MAP approximation of gradient:

$$\frac{\partial P_\omega(y|x)}{\partial \omega_i} = n_i(x,y) - E_\omega[n_i(x,y)] \approx n_i(x,y) - n_i(x,y^*_\omega)$$

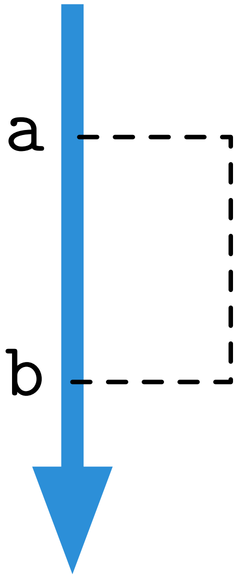
  where y* is the MAP solution

- The contribution of a single grounding is:
  - 0 if the MAP state of the grounding matches its target
  - +1 or -1 when targets and MAP inference disagree
  - MAP inference actively chooses examples for the neural network
  - other online active learners may fit well this framework, e.g. LaSVM (Bordes et al. 2005)

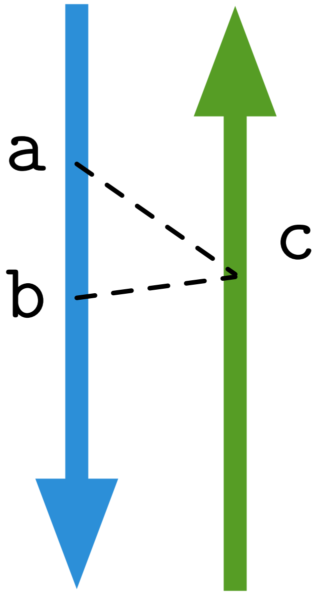
Experimental setting

- Data set from Cheng & Baldi (2005):
  - 916 protein chains from the Protein Data Bank, filtered for redundancy using UniqueProt @ HSSP=0 (<20% sequence identity)
  - 48,996 β-residues, computed by DSSP
  - 31,638 interstrand residue pairs
  - multiple alignment profiles obtained from PSI-BLAST against the nr database

Rules (some examples)

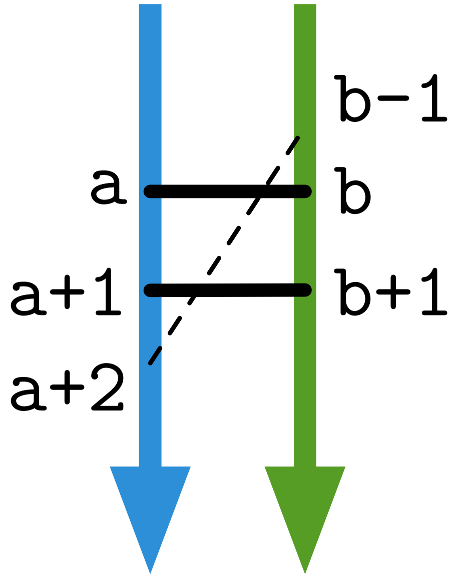


```
// Residues in the same strand are not partners

In(a,r) ∧ In(b,r) ⇒ ¬Partners(a,b)
```

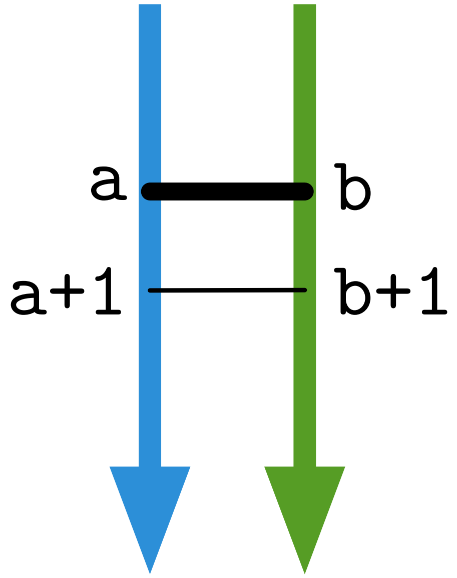


```
// No residue has two partners in the same strand

In(a,r) ∧ In(b,r) ⇒ ¬(Partners(c,a) ∧ Partners(c,b))
```

Rules (some examples)

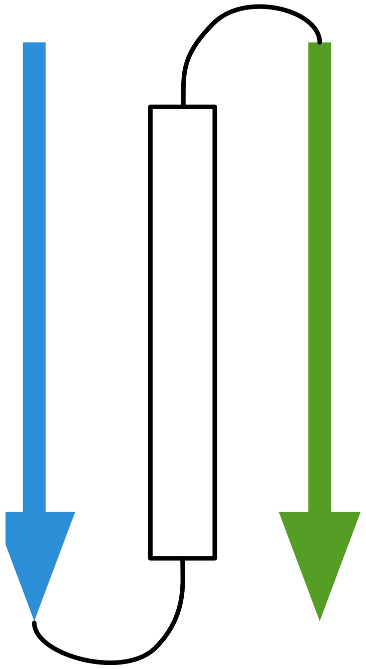


```
// Prohibit crossing edges

Partners(a,b) ∧ Partners(a+1,b+1) ⇒
                 ¬Partners(a+2,b-1)
```

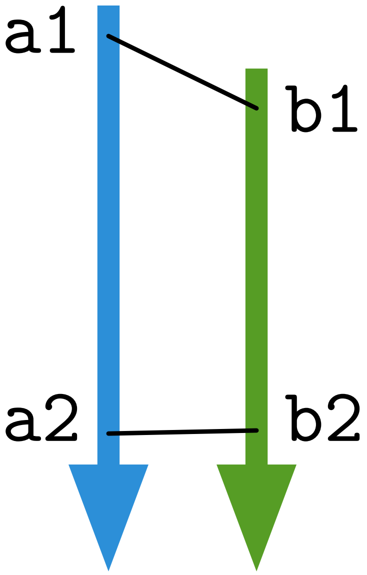


```
// Encourage partnership of successors

Parallel(r,s) ∧ Partners(a,b) ∧
          In(a,r) ∧ In(b,s) ⇒ Partners(a+1,b+1)
```

Rules (some examples)



```
// β-α-β configuration

HelixBetween(r,s) ∧ |r|≥4 ∧ |s|≥4 ⇒ Parallel(r,s)

¬HelixBetween(r,s) ∧ |r|≥2 ∧ |s|≥2 ⇒ ¬Parallel(r,s)
```



```
// Encourage partnership at endpoints

Parallel(r,s) ∧ First(a1,r) ∧
     Last(a2,r) ∧ First(b1,s) ∧ Last(b2,s)
     ⇒ Partner(a1,b1) ∨ Partner(b1,b2)
```

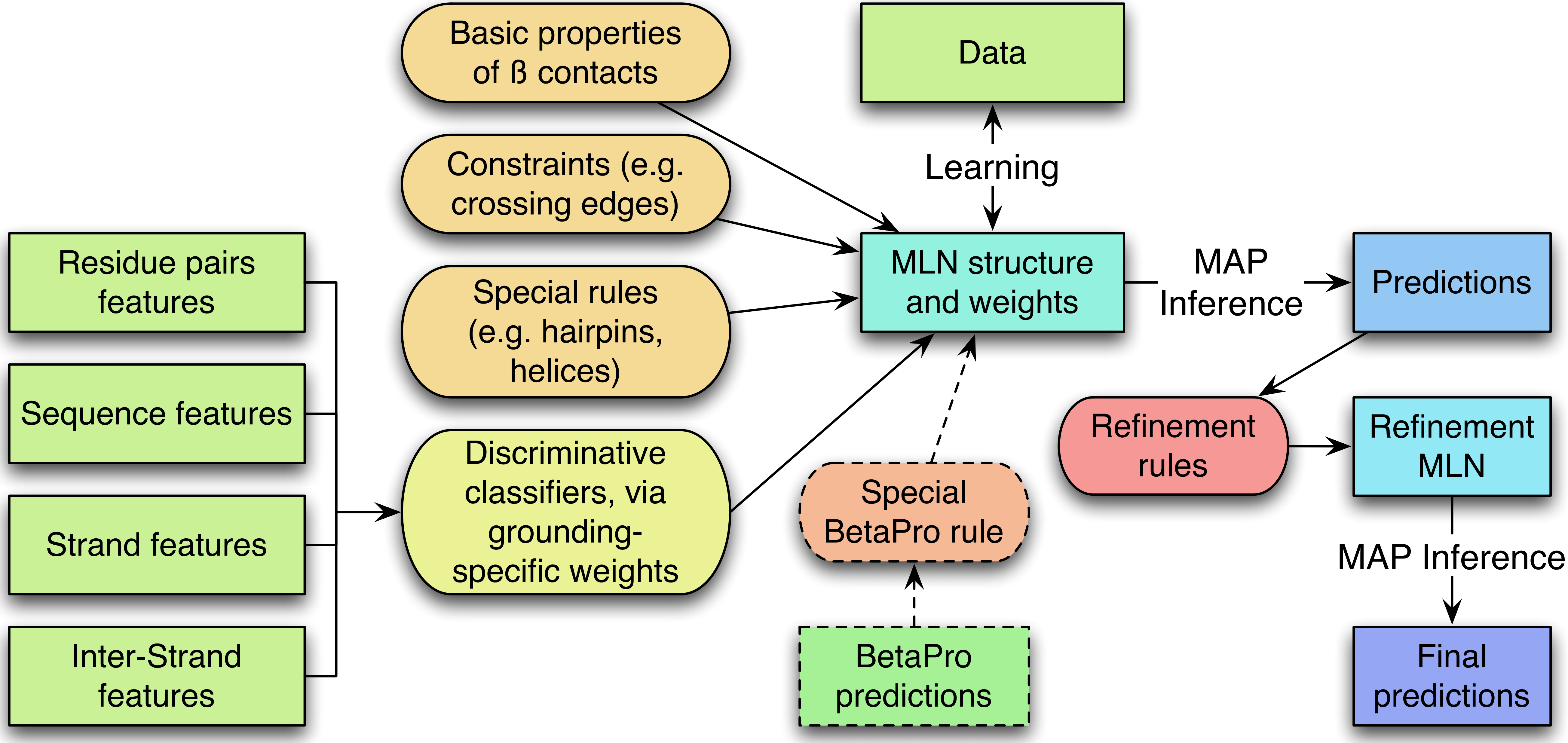
More rules...

- 54 formulae, expressing domain knowledge derived from literature and inspection of available data
- No structure learning
- Special rules embedding discriminative classifiers, e.g.

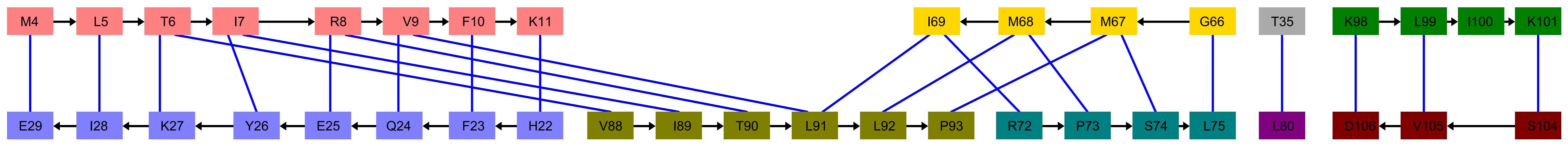
  ```
  Features(a,$wa) ∧ Features(b,$wb) ⇒ Partners(a,b)
  ```

  where `$wa` is a special macro that retrieves the vector of multiple alignment profiles (then used to construct the neural network input)
- Special rules for iterative relabeling

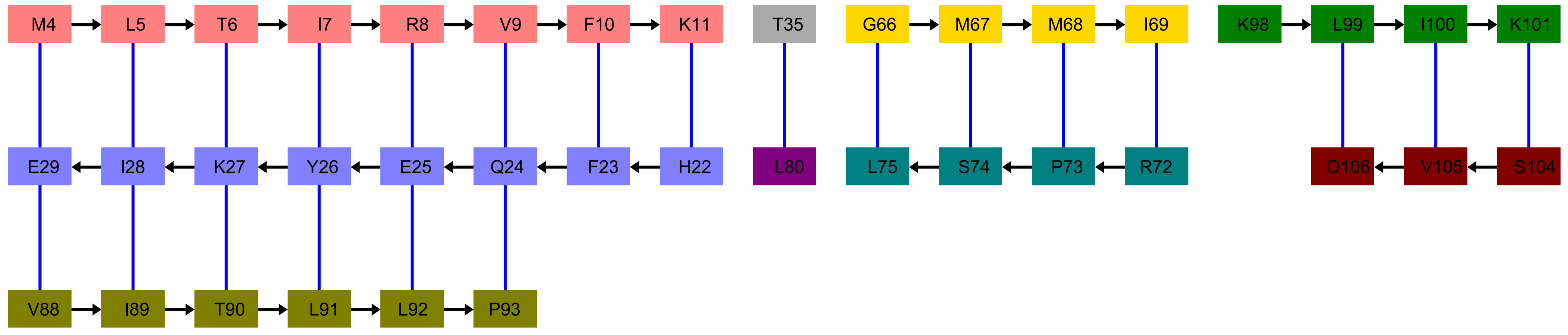
Overall architecture



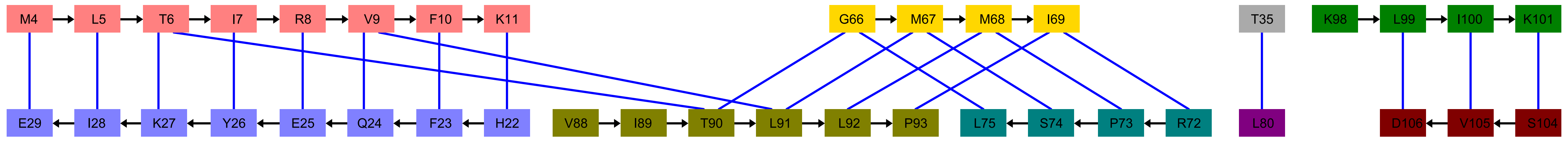
venerdì 15 gennaio 2010



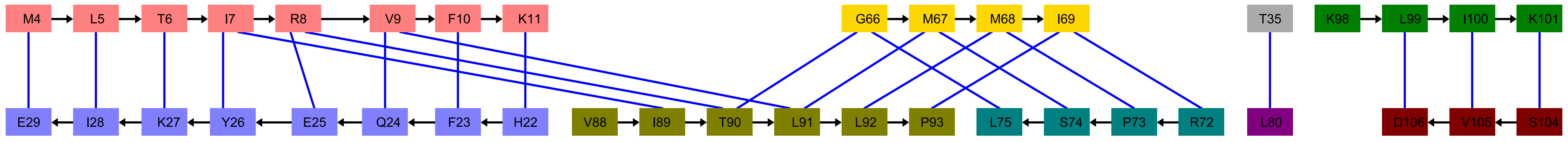
1QLAE TRUE MAP



1QLAE BETAPRO MAP

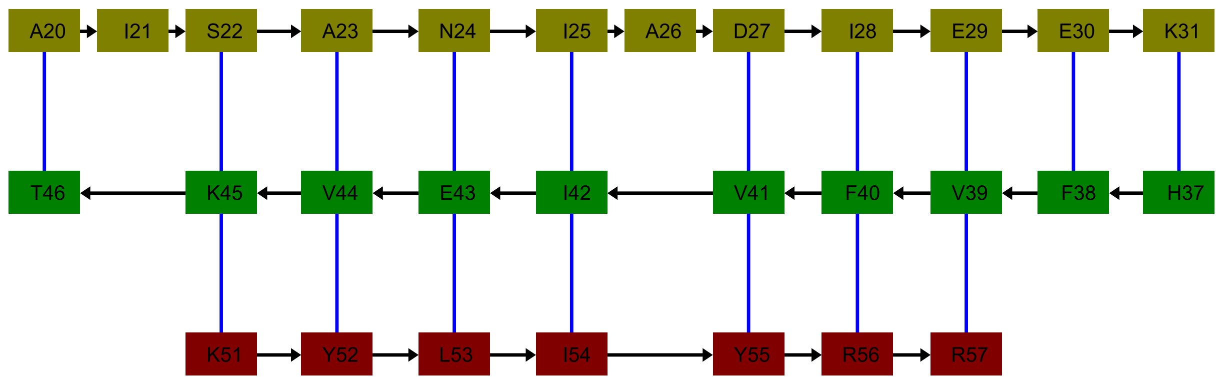


1QLAE MLN MAP

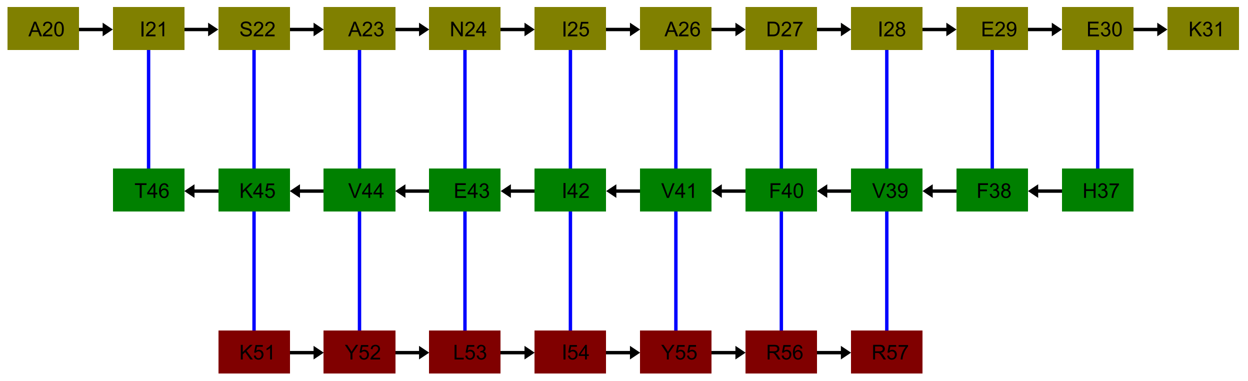


1QLAE MLN-2S MAP

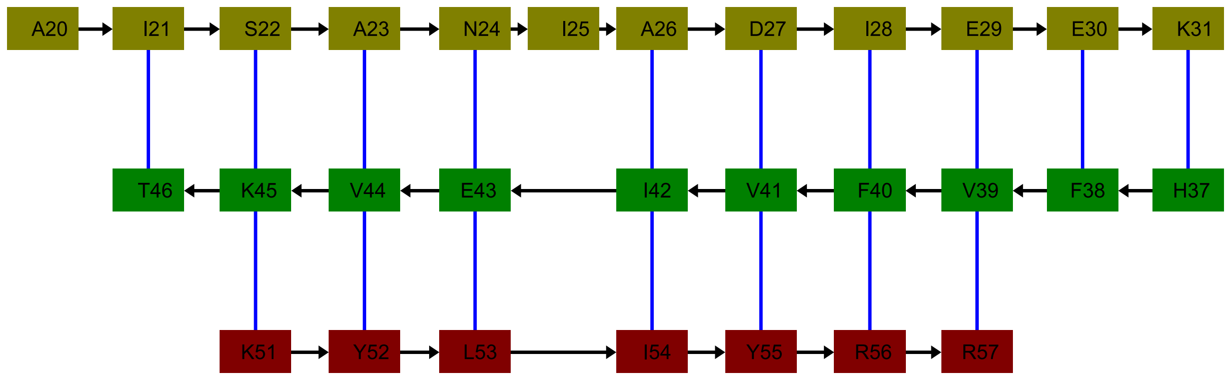
venerdì 15 gennaio 2010



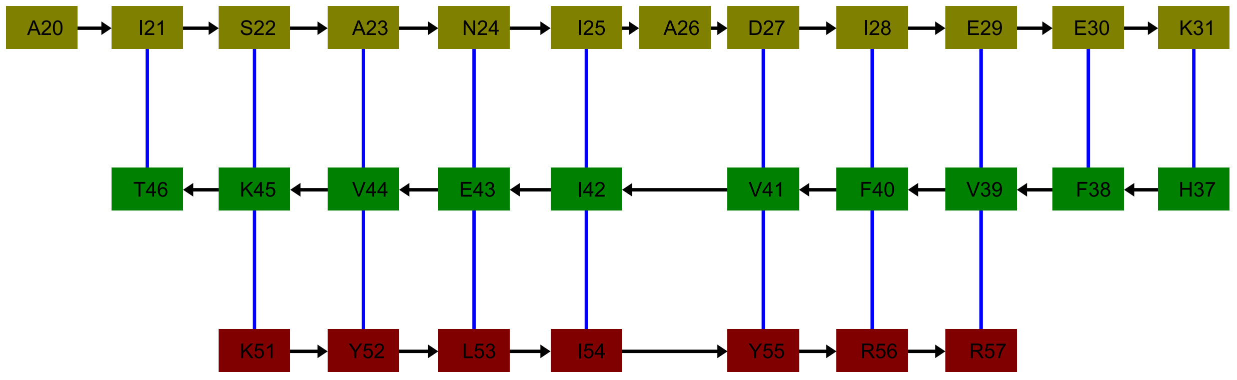
1H6HA TRUE MAP



1H6HA BETAPRO MAP



1H6HA MLN MAP

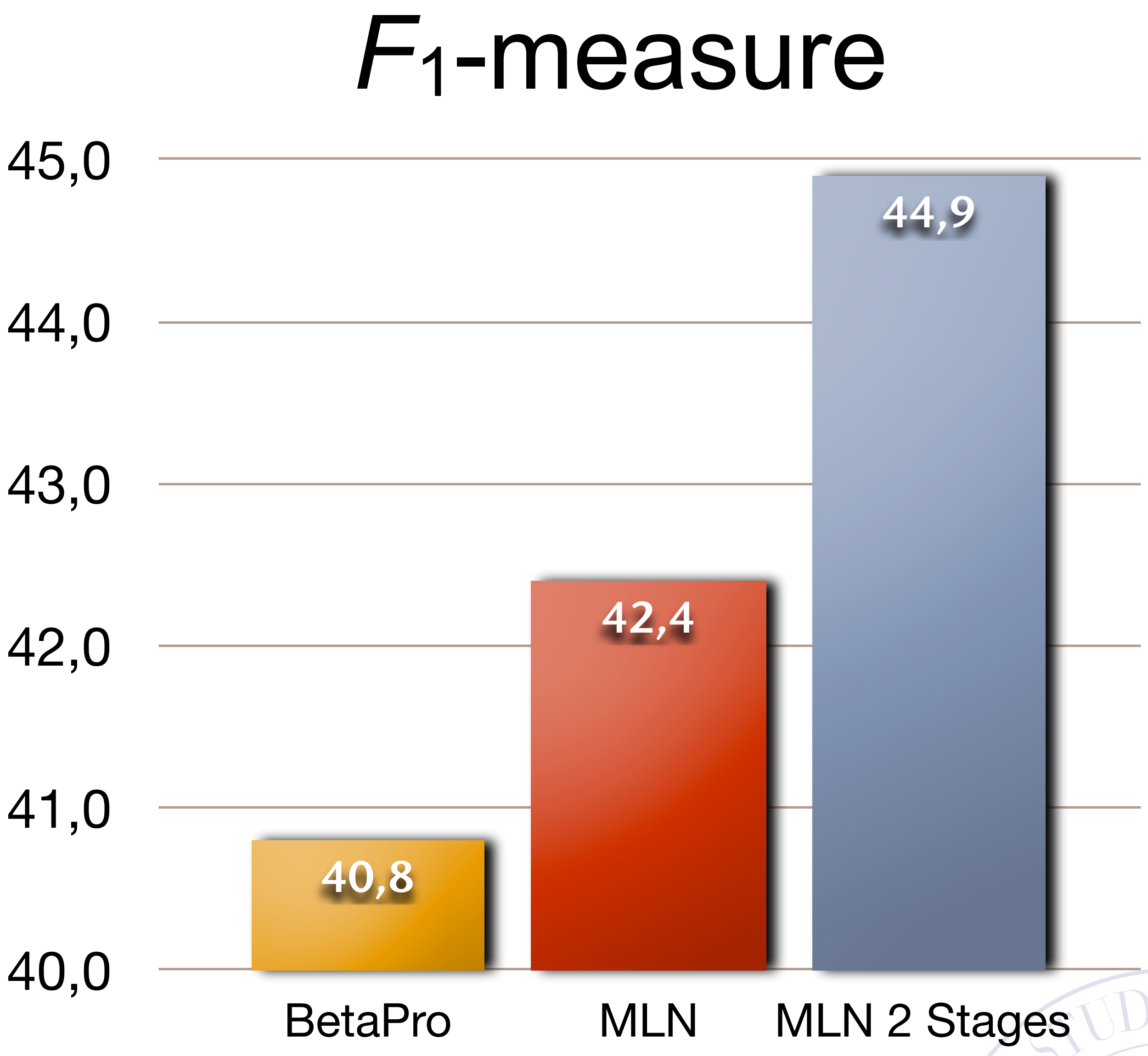


1H6HA MLN-2S MAP

Quantitative results at residue level

- F1-measure: harmonic mean of precision/recall), where:
  - P = ratio of correct/predicted links
  - R = ratio of correct/existing links
- BetaPro = 2D-RNN + Energy based alignment

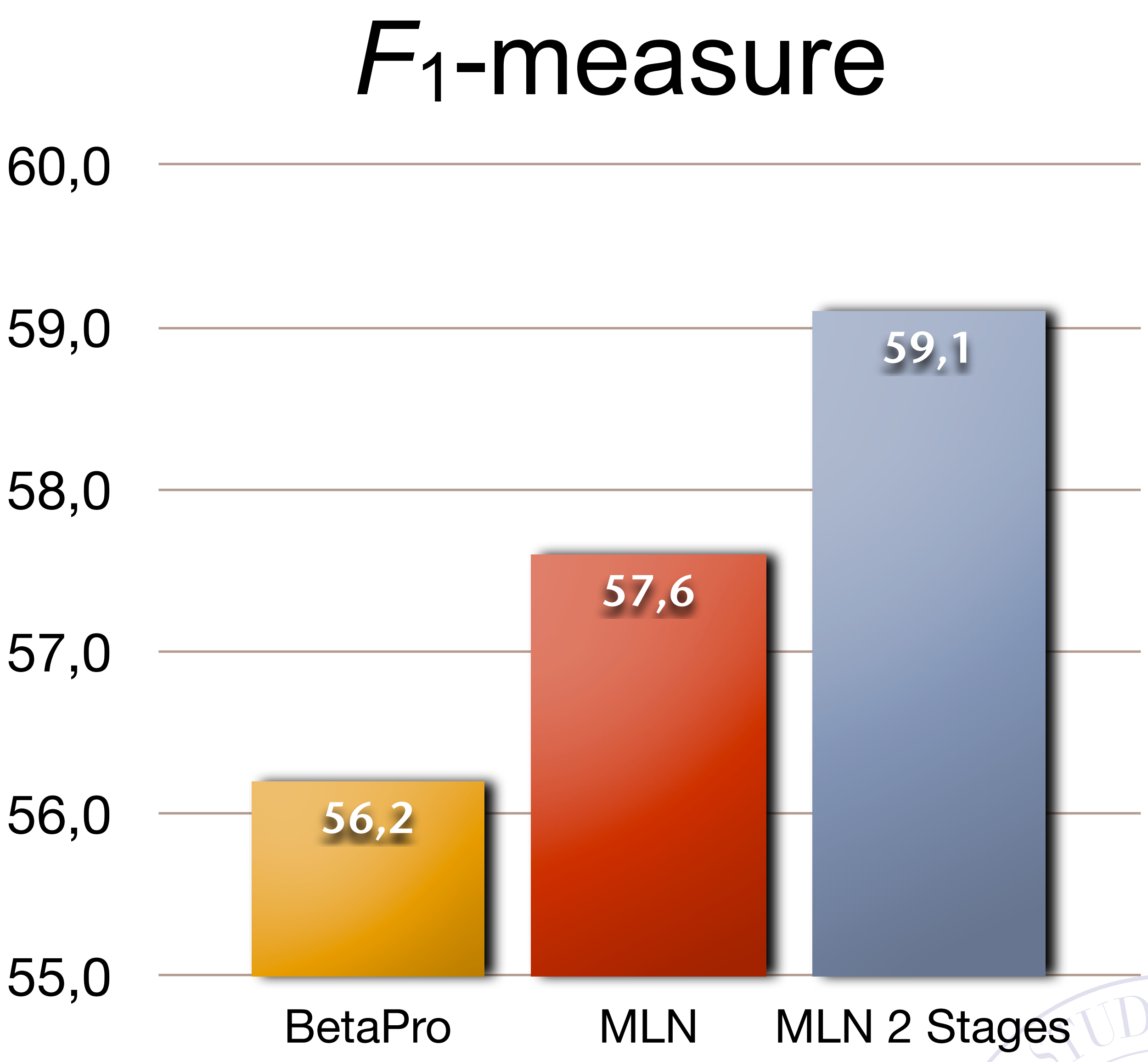
Differences are significant ($p < 0.01$)



Quantitative results at strand level

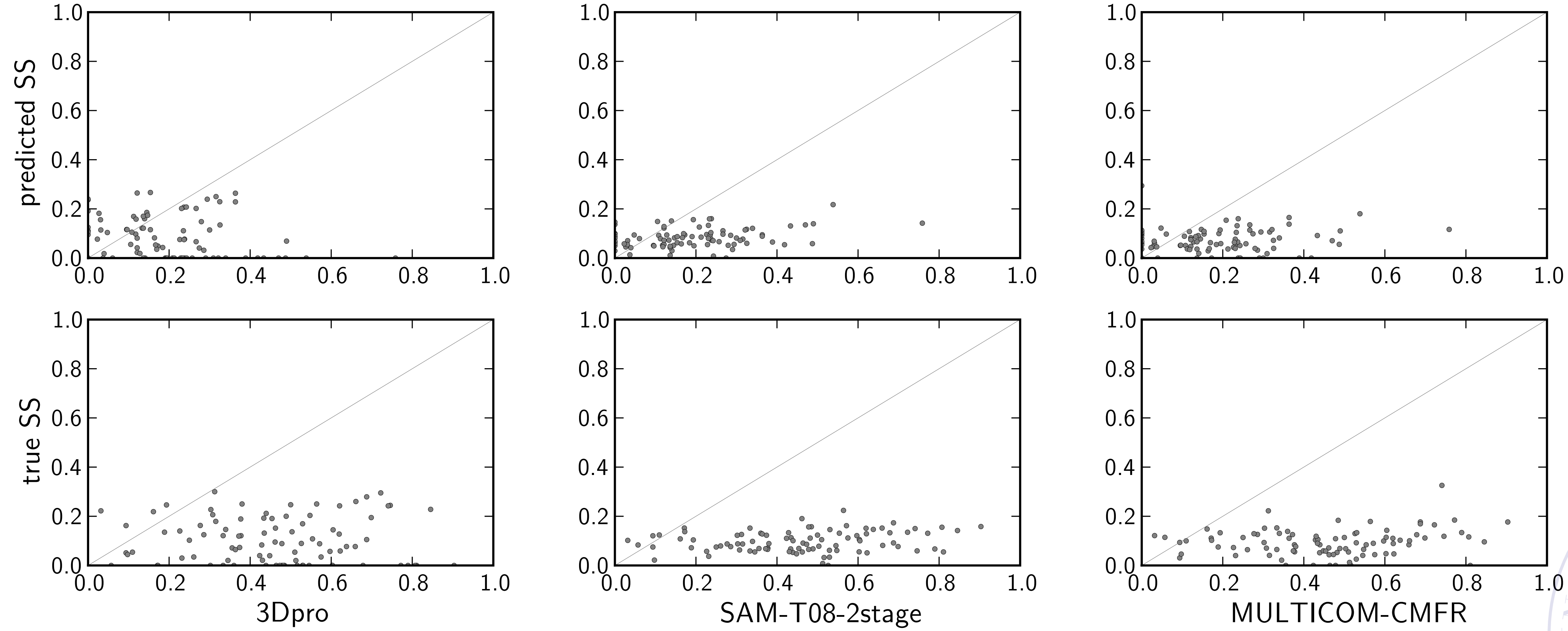
- F1-measure: harmonic mean of precision/recall), where:
  - P = ratio of correct/predicted links
  - R = ratio of correct/existing links
- BetaPro = 2D-RNN + Energy based alignment

Differences are significant ($p < 0.01$)



CASP 2008 data set

- We compared GS-MLN against three state-of-the-art CASP 2008 residue contact predictors
- Data set: 90 chains containing at least 10 ß-residues, X-ray determined structures



Summary

- Markov logic succeeded in a difficult and relatively large scale task
- Improvements over a highly-engineered state-of-the-art system
- Grounding-specific weights enable to incorporate complex (nonlinear) local decision functions, the idea is possibly reusable in different application domains
- Inference is the bottleneck: Better (faster) approximate algorithms are required for scaling this approach to even larger problems

Metalloproteins

- Metal binding plays important roles in protein function and structure: about 1/3 of all proteins are associated with a metal

Metalloproteins

- Metal binding plays important roles in protein function and structure: about 1/3 of all proteins are associated with a metal
- Metalloproteins involved in many biological processes (apoptosis, aging) and diseases (cancer, Parkinson, dementia, AIDS)

Metalloproteins

- Metal binding plays important roles in protein function and structure: about 1/3 of all proteins are associated with a metal

- Metalloproteins involved in many biological processes (apoptosis, aging) and diseases (cancer, Parkinson, dementia, AIDS)

Metalloproteins

- Metal binding plays important roles in protein function and structure: about 1/3 of all proteins are associated with a metal
- Metalloproteins involved in many biological processes (apoptosis, aging) and diseases (cancer, Parkinson, dementia, AIDS)
- Metallomics and metalloproteomics: emergent “omics”

Protein metal binding

- Metals with a prominent biological role:
  - Alkali (K, Na) – about 6% in PDB
  - Alkaline earth (Mg, Ca) – about 37% in PDB
  - **Transition metals** (Mn, Fe, Cu, Zn, Cd) – about 66% in PDB
  - The picture at the genomic scale is largely unknown

Protein metal binding

- Metals with a prominent biological role:
  - Alkali (K, Na) – about 6% in PDB
  - Alkaline earth (Mg, Ca) – about 37% in PDB
  - **Transition metals** (Mn, Fe, Cu, Zn, Cd) – about 66% in PDB
  - The picture at the genomic scale is largely unknown
- Alkali and alkaline earth metals: binding is mainly due to electrostatic interactions (⇒ low affinity)

Protein metal binding

- Metals with a prominent biological role:
  - Alkali (K, Na) – about 6% in PDB
  - Alkaline earth (Mg, Ca) – about 37% in PDB
  - **Transition metals** (Mn, Fe, Cu, Zn, Cd) – about 66% in PDB
  - The picture at the genomic scale is largely unknown
- Alkali and alkaline earth metals: binding is mainly due to electrostatic interactions (⇒ low affinity)
- **Transition metals**: ligands donate an electron pair to form coordinate covalent bonds (⇒ high affinity)

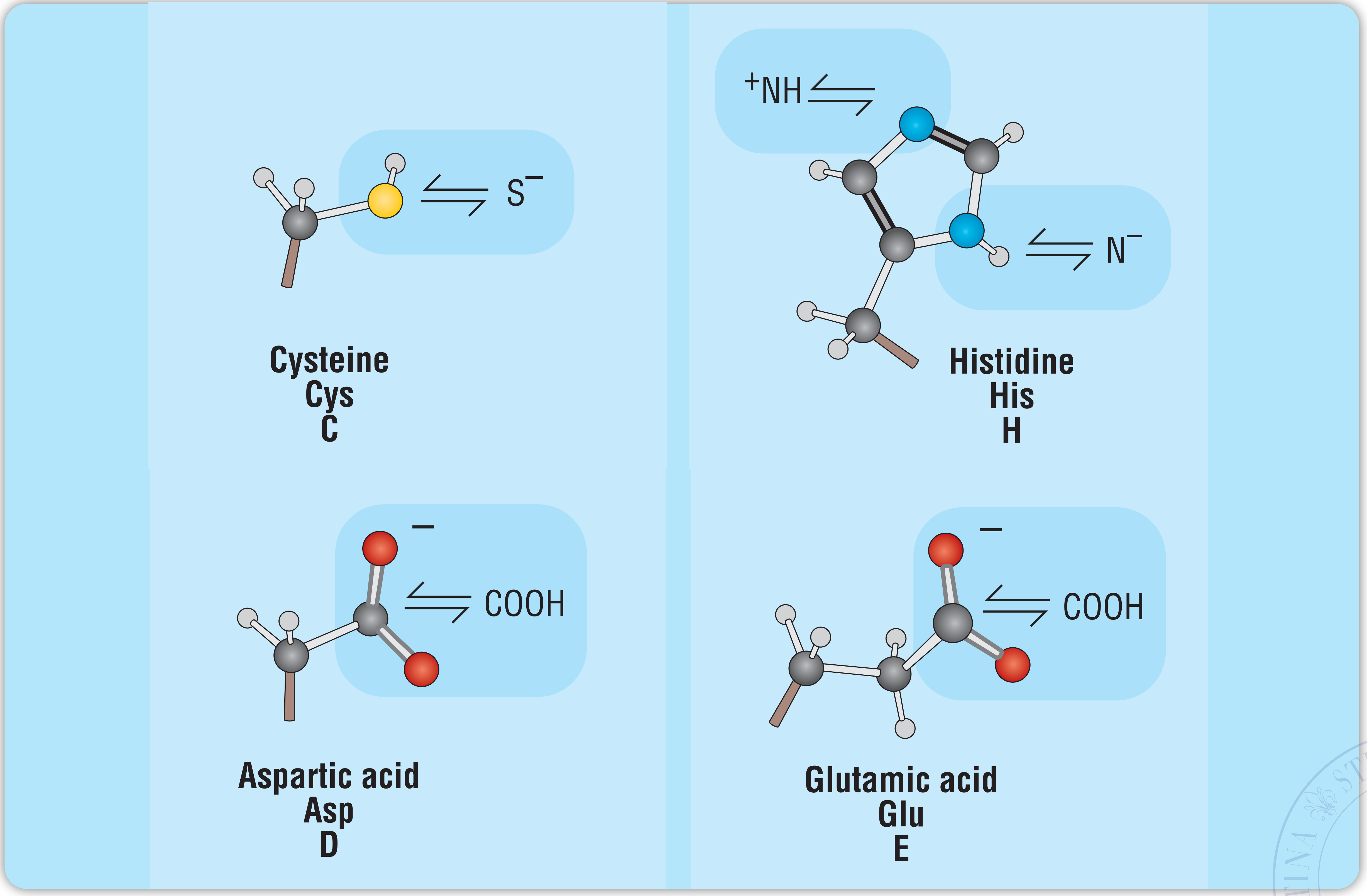
Binding to transition metals

- Coordination number varies considerably from a minimum of 1 to a maximum of 8

- Sites involving ≤ 2 residues tend to be located on the protein surface

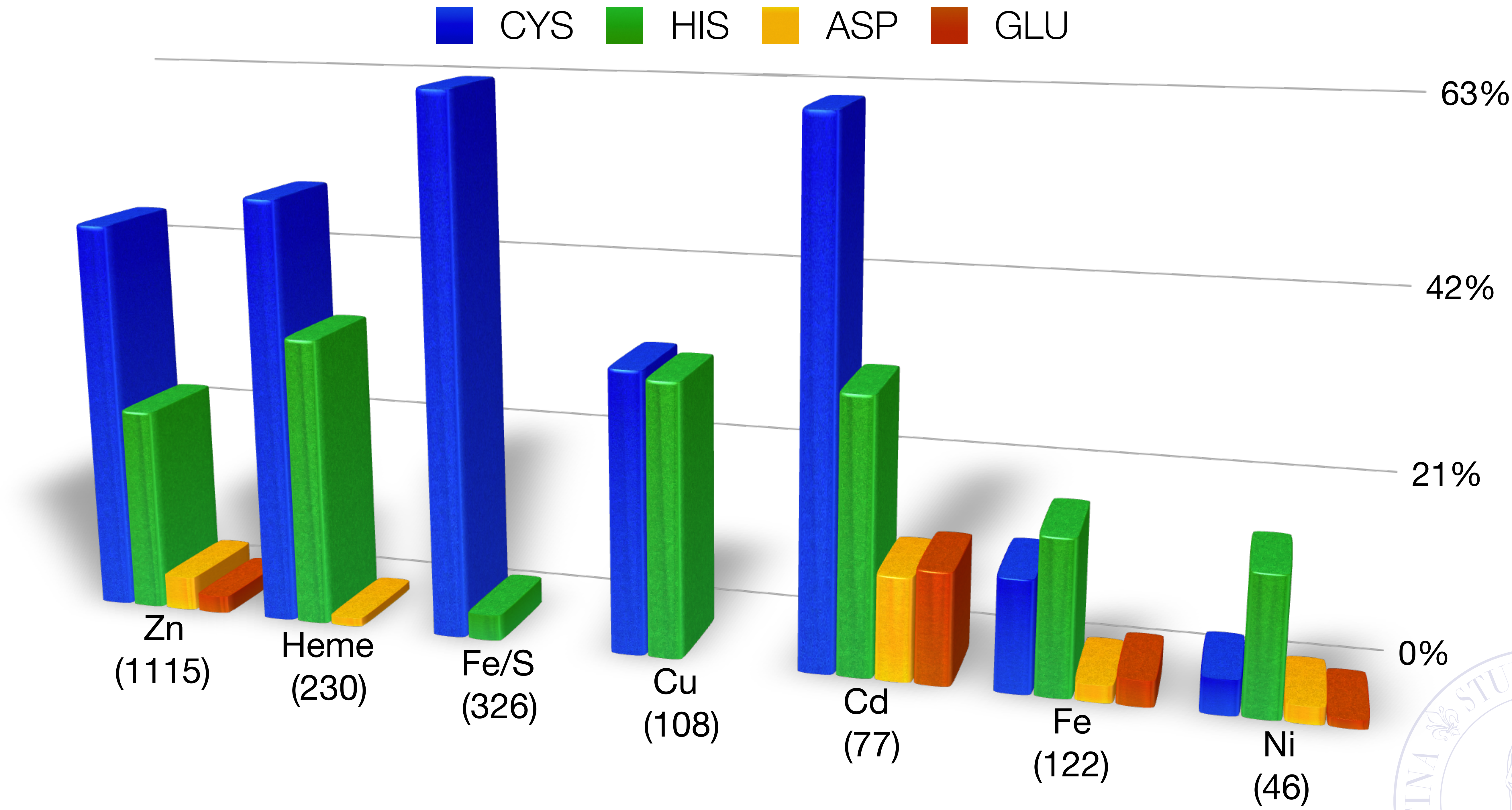
- Many transition metals (particularly Zn) coordinated by ≥ 3 residues are involved in catalytic, co-catalytic, or structural sites

Only some amino acids in Nature usually act as ligands and coordinate a metal ion



% times a given amino acid type binds a specific metal ion/complex in chains containing a binding site for that ion

Non redundant set of 2,727 protein chains (UniqueProt)



Identifying metalloproteins and binding sites

- High throughput technologies can identify metalloproteins but not binding sites (Shi & Chance, 2008)
- Bioinformatics approaches can provide useful alternative or complementary information

Identifying metalloproteins and binding sites

- High throughput technologies can identify metalloproteins but not binding sites (Shi & Chance, 2008)
- Bioinformatics approaches can provide useful alternative or complementary information
- PROSITE motifs, e.g. 4Fe-4S Ferredoxin `C-x(2)-C-x(2)-C-x(3)-C-[PEG]`
  high precision, low recall

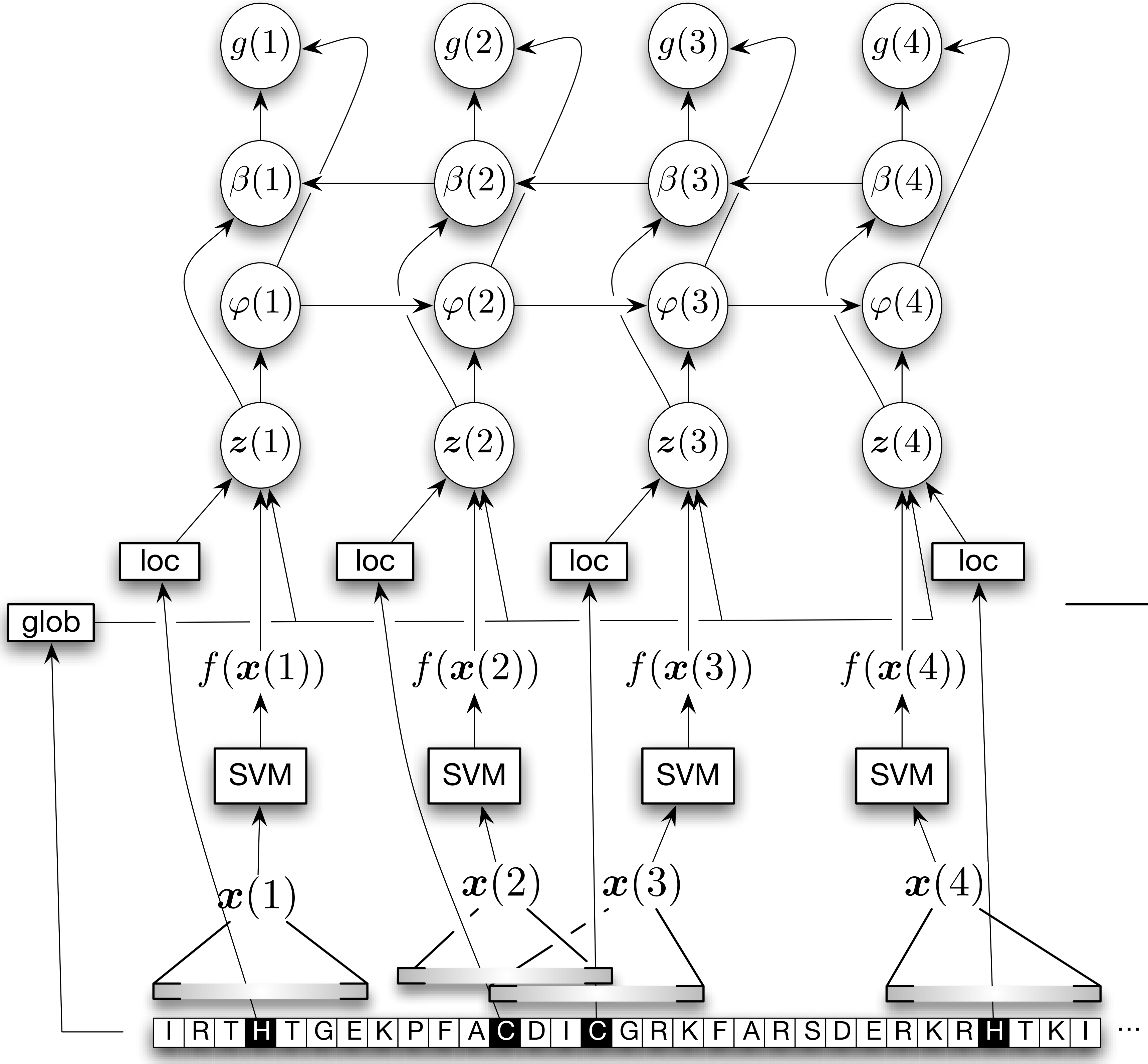
Identifying metalloproteins and binding sites

- High throughput technologies can identify metalloproteins but not binding sites (Shi & Chance, 2008)
- Bioinformatics approaches can provide useful alternative or complementary information
- PROSITE motifs, e.g. 4Fe-4S Ferredoxin `C-x(2)-C-x(2)-C-x(3)-C-[PEG]`
  high precision, low recall
- Binding sites very compact in 3D: prediction from known structures easy (Sodhi et al. 2004; Ebert & Altman 2007)

Bonding state determination

- For each candidate ligand (in {C,H,D,E}), predict the bonding state as free vs. metal-bound (binary classification)
- In the (special but important) case of cysteines, a third class is associated with disulfide bridges
- The most relevant features for learning are multiple alignment profiles in a window of residues flanking each candidate ligand

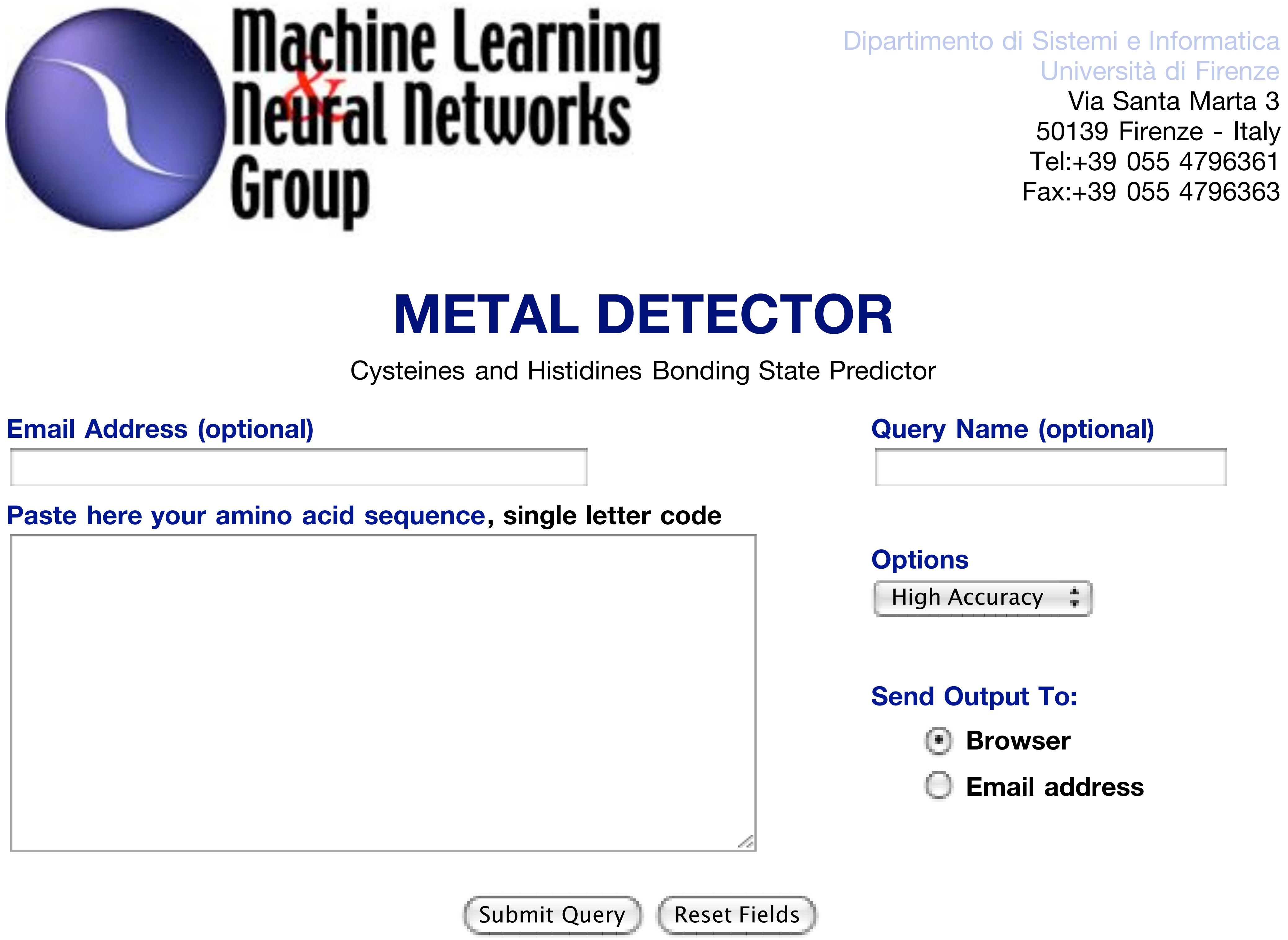
Metal ligand predictor
(Passerini, Ceroni, Punta, Rost & Frasconi 2006)



Second stage: BRNN

First stage: SVM

MetalDetector (Lippi, Passerini, Punta, Rost & Frasconi 2008)



Machine Learning & Neural Networks Group

Dipartimento di Sistemi e Informatica
Università di Firenze
Via Santa Marta 3
50139 Firenze - Italy
Tel:+39 055 4796361
Fax:+39 055 4796363

METAL DETECTOR

Cysteines and Histidines Bonding State Predictor

Email Address (optional)

Query Name (optional)

Paste here your amino acid sequence, single letter code

Options
High Accuracy

Send Output To:
● Browser
○ Email address

METAL DETECTOR

Cysteines and Histidines Bonding State Predictor

Results for **0phB3c**

```
            ........10........20........30........40........50........60........70.........
AA          AFVVTDNCIKCKYTDCVEVSPVDCFYEGPNFLVIHPDECIDCALCEPECPAQAIFSEDEVPEDMQEFIQLNAELAEVWP
State              M  F    M       M          M   M  M  M   M

            80........90........100....
AA          NITEKKDPLPDAEDWDGVKGKLQHLER
State                              F
```

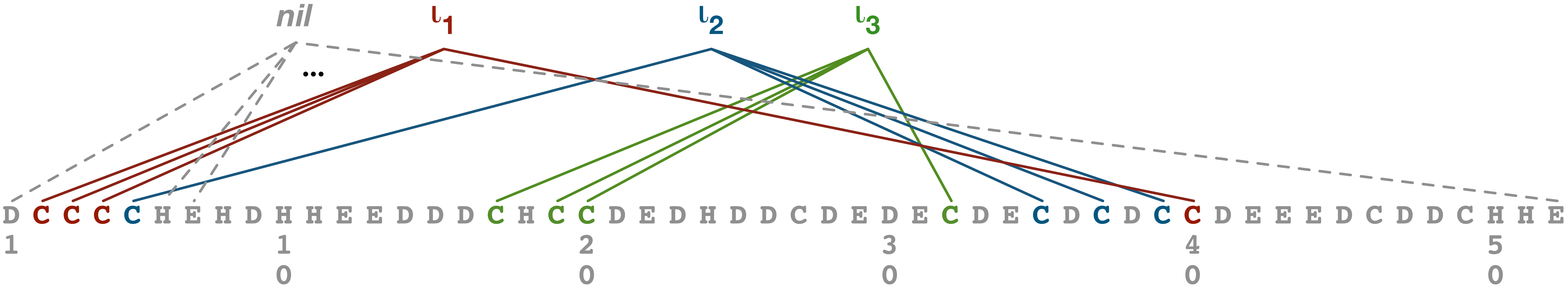
Position	**Residue**	**Prediction**	**Metal**	**Free**	**Disul**
7	C	M	**0.64**	0.18	0.18
10	C	F	0*	**0.76**	0.24
15	C	M	**0.94**	0.03	0.03
23	C	M	**0.76**	0.21	0.02
34	H	M	**0.56**	0.44	
38	C	M	**0.99**	0.01	0
41	C	M	**0.98**	0.01	0.01
44	C	M	**0.99**	0.01	0.01
48	C	M	**0.99**	0.01	0
102	H	F	0.1	**0.9**	

Metal binding geometry: Formalization

- Protein sequence: a string s in the AA alphabet
- Candidate ligands: CYS and HIS
- Most ions are coordinated by few residues. Using a maximum of m=4 ligands covers 93% known proteins
- Include a special nil symbol for “free” amino acids

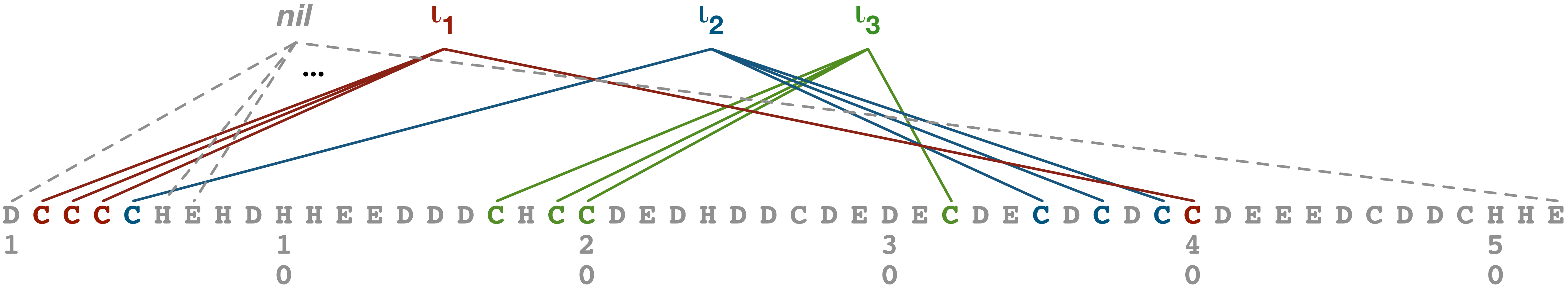
Metal binding geometry as a structured output learning problem (Frasconi & Passerini 2009)

Metal binding structure of PDB entry 1H0Hb


nil
ι_1
ι_2
ι_3
...
D C C C C H E H D H H E E D D D C H C C D E D H D D C D E D E C D E C D C D C D E E E D C D D C H H E
1
10
20
30
40
50

Metal binding geometry as a structured output learning problem (Frasconi & Passerini 2009)

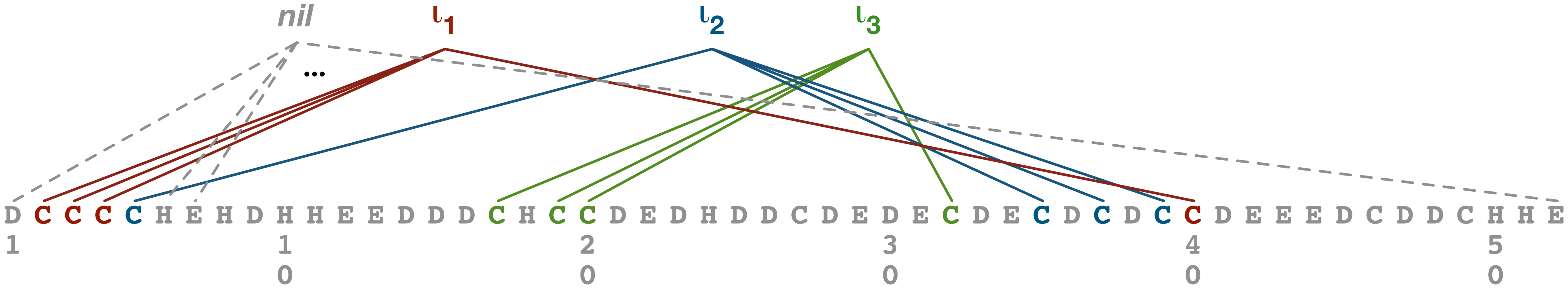
Metal binding structure of PDB entry 1H0Hb


nil
ι_1
ι_2
ι_3
...
D C C C C H E H D H H E E D D D C H C C D E D H D D C D E D E C D E C D C D C C D E E E D C D D C H H E
1
10
20
30
40
50

Goal: Predict edges y in a bipartite graph $(x \cup \mathcal{I}, y)$ where x is the subsequence of s after removing non-candidate residues and $\mathcal{I}$ a set of (anonymous) ion identifiers

Metal binding geometry as a structured output learning problem (Frasconi & Passerini 2009)

Metal binding structure of PDB entry 1H0Hb



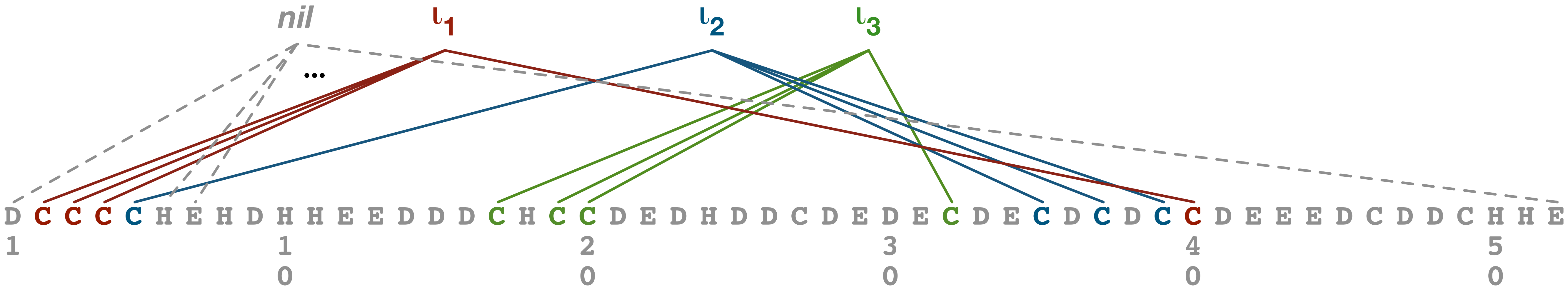
nil
...
ι_1
ι_2
ι_3
D C C C C H E H D H H E E D D D C H C C D E D H D D C D E D E C D E C D C D C C D E E E D C D D C H H E
1
10
20
30
40
50

Goal: Predict edges y in a bipartite graph $(x \cup \mathcal{I}, y)$ where x is the subsequence of s after removing non-candidate residues and $\mathcal{I}$ a set of (anonymous) ion identifiers

MBG Property: A bipartite edge set $y \subset x \times \mathcal{I}$ satisfies the metal binding geometry (MBG) property if the degree of each vertex in x in the graph $(x \cup \mathcal{I}, y)$ is at most 1.

Metal binding geometry as a structured output learning problem

Metal binding structure of PDB entry 1H0Hb


nil
ι_1
ι_2
ι_3
...
D C C C C H E H D H H E E D D D C H C C D E D H D D C D E D E C D E C D C D C C D E E E D C D D C H H E
1
10
20
30
40
50

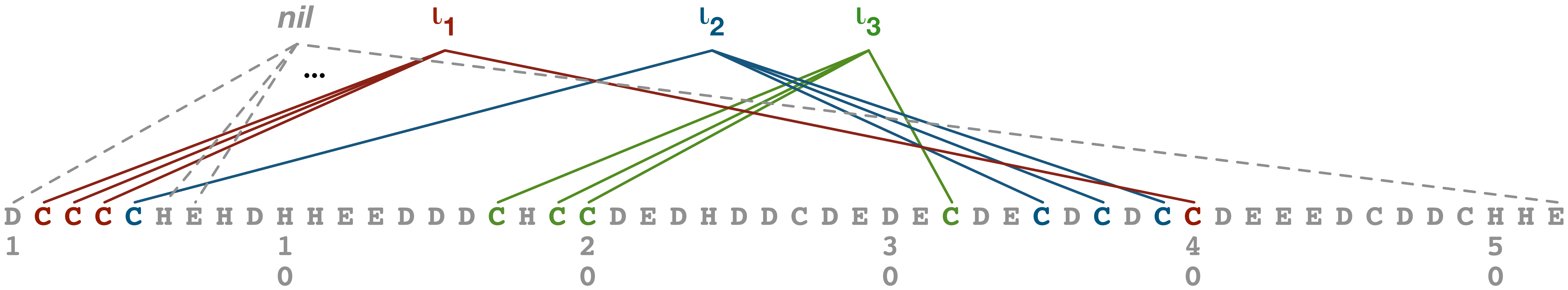
How many alternatives? Let $|x| = n$ the # of candidate ligands, m the number of ions, and k_i the number of ligands for ion ι_i. Then the number of possible geometries is the multinomial coefficient

$$\frac{n!}{k_1!k_2!\cdots k_m!(n-k_1-\cdots-k_m)!}$$

In the example $n = 52$ and $m = 3$ ions coordinated by 4 residues each, yielding $\approx 7 \cdot 10^{15}$ admissible conformations.

Metal binding geometry as a structured output learning problem

Metal binding structure of PDB entry 1H0Hb


nil
ι_1
ι_2
ι_3
...
D C C C C H E H D H H E E D D D C H C C D E D H D D C D E D E C D E C D C D C C D E E E D C D D C H H E
1
10
20
30
40
50

- If we interpret the bipartite graph as a sort of “parse tree”, it is immediately apparent is that the underlying grammar needs to be context sensitive in order to capture the crossing-dependencies between bound amino acids.

The metal binding geometry problem

Find:

$$\arg\max_{y \in \mathcal{Y}_x} F_x(y)$$

where $\mathcal{Y}_x$ is the set of y that satisfy the MBG property and $F_x : \mathcal{Y}_x \mapsto I\!R^+$ a function that assigns a positive score to each bipartite edge set in $\mathcal{Y}_x$.

Not a matching problem as in (Taskar *et al.* 2005): more than one edge can be incident on vertices belonging to $\mathcal{I}$.

Matroids

Algebraic structure $\mathcal{M} = (S, \mathcal{Y})$ where S is a finite set and $\mathcal{Y}$ a family of so-called *independent* subsets of S such that:

i) $\emptyset \subseteq \mathcal{Y}$;

ii) all proper subsets of a set y in $\mathcal{Y}$ are in $\mathcal{Y}$;

iii) if y and y' are in $\mathcal{Y}$ and $|y| < |y'|$ then there exists $e \in y' \setminus y$ such that $y \cup \{e\} \in \mathcal{Y}$.

Matroids

Algebraic structure $\mathcal{M} = (S, \mathcal{Y})$ where S is a finite set and $\mathcal{Y}$ a family of so-called *independent* subsets of S such that:

i) $\emptyset \subseteq \mathcal{Y}$;

ii) all proper subsets of a set y in $\mathcal{Y}$ are in $\mathcal{Y}$;

iii) if y and y' are in $\mathcal{Y}$ and $|y| < |y'|$ then there exists $e \in y' \setminus y$ such that $y \cup \{e\} \in \mathcal{Y}$.

If y is an independent set, then $\text{ext}(y) = \{e \in S : y \cup \{e\} \in \mathcal{Y}\}$ is called the **extension set** of y. A maximal (having an empty extension set) independent set is called a **base**.

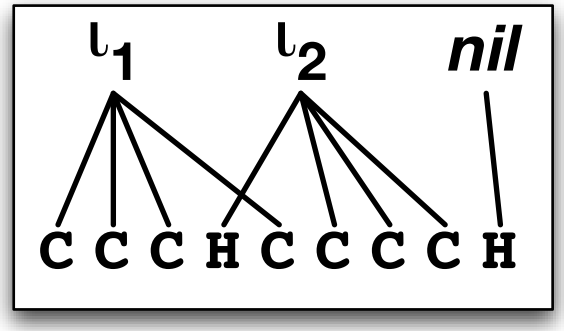
Metal binding and matroids

Theorem If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.

Metal binding and matroids

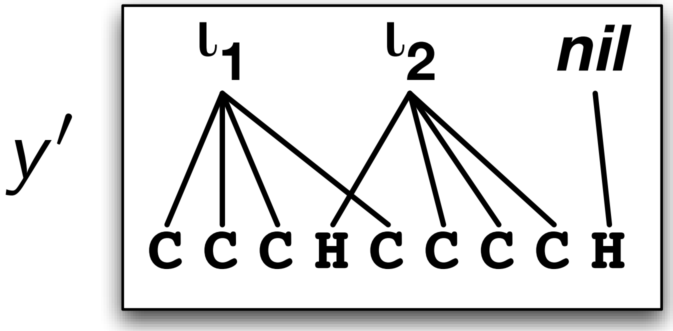
Theorem If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.

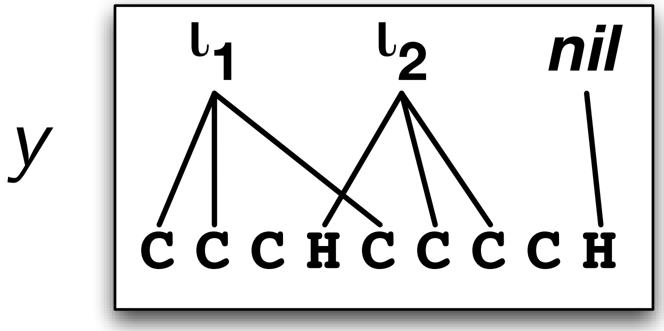
y'



Metal binding and matroids

Theorem If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.

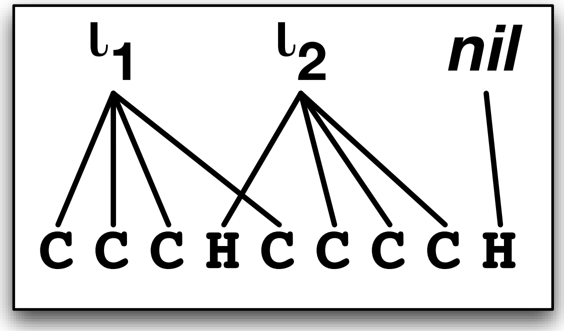




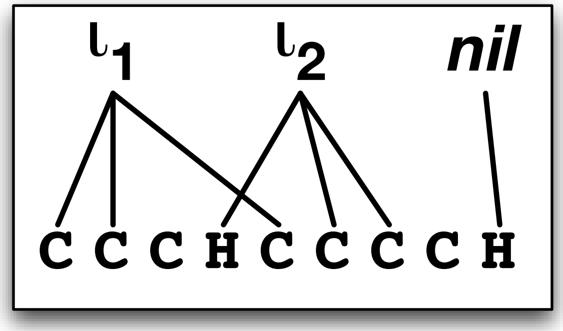
Metal binding and matroids

Theorem If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.

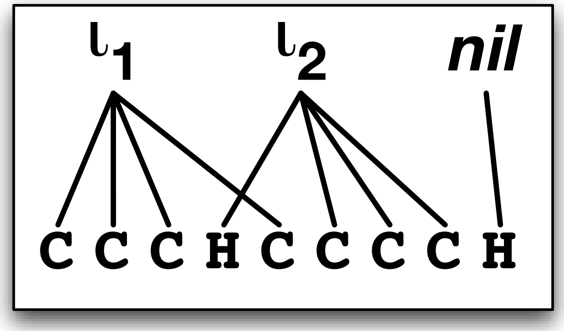
y'



y



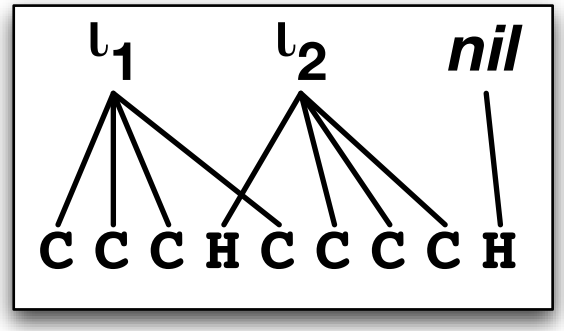
y'



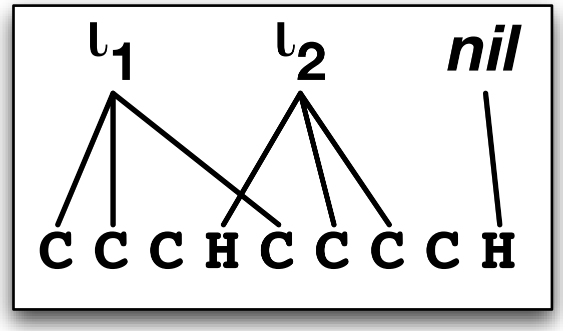
Metal binding and matroids

Theorem If each $y \in \mathcal{Y}_x$ satisfies the MBG property, then $\mathcal{M}_x = (S_x, \mathcal{Y}_x)$ is a matroid.

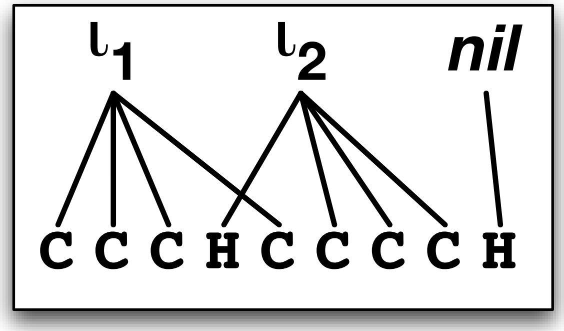
y'



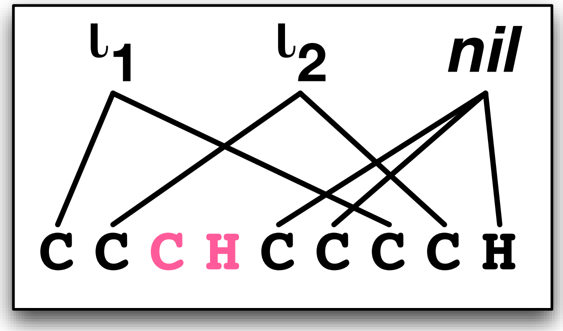
y



y'



y



Matroids and greedy algorithms

The following classic result is the support for many greedy algorithms:

Theorem (Edmonds 1967). For any nonnegative $v(\cdot)$, a lexicographically maximum base in $\mathcal{Y}$ maximizes the global objective function

$$F(y) = \sum_{e \in y} v(e)$$

As a result, the following algorithm finds an optimal structure on a weighted matroid:

```
GreedyConstruct($\mathcal{M}$, $F$)
    $y \leftarrow \emptyset$
    while $\mathrm{ext}(y) \neq \emptyset$:
        $y \leftarrow y \cup \left\{ \arg \max_{e \in \mathrm{ext}(y)} F(y \cup \{e\}) \right\}$
    return $y$
```

Additive objective functions?

If F is a sum of edge weights (as in the classic theorem) then each weight contributes independently while the whole point of structured output learning is to **collectively** decide which parts should be present in the output structure

Additive objective functions?

If F is a sum of edge weights (as in the classic theorem) then each weight contributes independently while the whole point of structured output learning is to **collectively** decide which parts should be present in the output structure

Theorem (Helman *et al.* 1993).
If F is *consistent*, i.e. for any $y \subset y' \subset S$ and $e, e' \in S \setminus y'$ satisfies

$$F(y \cup \{e\}) \geq F(y \cup \{e'\}) \Rightarrow F(y' \cup \{e\}) \geq F(y' \cup \{e'\})$$

then, for each matroid on S, all greedy bases are optimal.

Formulation as structured output

Data set: $\mathcal{D} = \{(x_i, y_i)\}$ where x_i is a string in $\mathcal{T}^*$ and y_i a bipartite graph.

Formulation as structured output

Data set: $\mathcal{D} = \{(x_i, y_i)\}$ where x_i is a string in $\mathcal{T}^*$ and y_i a bipartite graph.

Space of objective functions:
Given input string x and (partial) output structure $y \in \mathcal{Y}$, let $F_x(y) = w^T \phi_x(y)$ being w a weight vector and $\phi_x(y)$ a feature vector for (x, y).

Formulation as structured output

Data set: $\mathcal{D} = \{(x_i, y_i)\}$ where x_i is a string in $\mathcal{T}^*$ and y_i a bipartite graph.

Space of objective functions:
Given input string x and (partial) output structure $y \in \mathcal{Y}$, let $F_x(y) = w^T \phi_x(y)$ being w a weight vector and $\phi_x(y)$ a feature vector for (x, y).

Prediction:

$$f(x) = \arg\max_{y \in \mathcal{Y}_x} F_x(y)$$

where the objective function F_x must satisfy the following properties, ensuring that the above argmax can be computed in a greedy fashion:

F_x is consistent (in the sense of Theorem 2)

$\forall i : F_{x_i}(y' \cup \{e\}) > F_{x_i}(y' \cup \{e'\}) \; \forall \; y' \subset y_i, e \in \text{ext}(y') \cap y_i, e' \in \text{ext}(y') \setminus y_i$

Max margin formulation

Ensure that correct extensions receive a higher weight than wrong extensions

$$\min \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to:} \quad w^T\Big(\phi_{x_i}(y' \cup \{e\}) - \phi_{x_i}(y' \cup \{e'\})\Big) \geq 1$$

$$w^T\Big(\phi_{x_i}(y'' \cup \{e\}) - \phi_{x_i}(y'' \cup \{e'\})\Big) \geq 1$$

$$\forall i = 1, \ldots, |\mathcal{D}|,$$

$$\forall e \in \mathrm{ext}(y') \cap y_i,\ \forall e' \in \mathrm{ext}(y') \setminus y_i,$$

$$\forall y' \subset y_i,\ \forall y'' : y' \subset y'' \subset S_x.$$

Force the objective function to obey the consistency constraints so we can be greedy

Learning algorithm: main ideas

- Online algorithm (e.g. LaSVM, Bordes et al. 2005)
- For each example, keep a best current structure (initially the empty set of edges)
- Pick an example and add edges greedily, based on current score F, trying to reconstruct the target structure
- Sample “bad edges” and enforce correctness constraints

Using a kernel function

We represent the objective function F using a kernel

$$k(z, z') = \langle \phi_x(y), \phi_{x'}(y') \rangle$$

between two structured instances $z = (x, y)$ and $z' = (x', y')$, so that

$$F_x(y) = F(z) = \sum_i \alpha_i k(z, z_i).$$

The kernel

binding-site kernel

edges incident on i-th ion

$$k(z,z') = k_{\mathrm{glob}}(z,z') \sum_{i=1}^{n(z)} \sum_{j=1}^{n(z')} \frac{k_{\mathrm{mbs}}(\sigma_i(z), \sigma_j(z'))}{n(z)n(z')}$$

global kernel

# of ions having at least one incident edge

The global kernel

similarity is zero unless the two proteins have the same # of sites

# of candidate ligands should be similar

$$k_{\text{glob}}(z, z') = \delta(n(z), n(z')) \frac{2 \min\{|x|, |x'|\}}{|x| + |x'|}$$

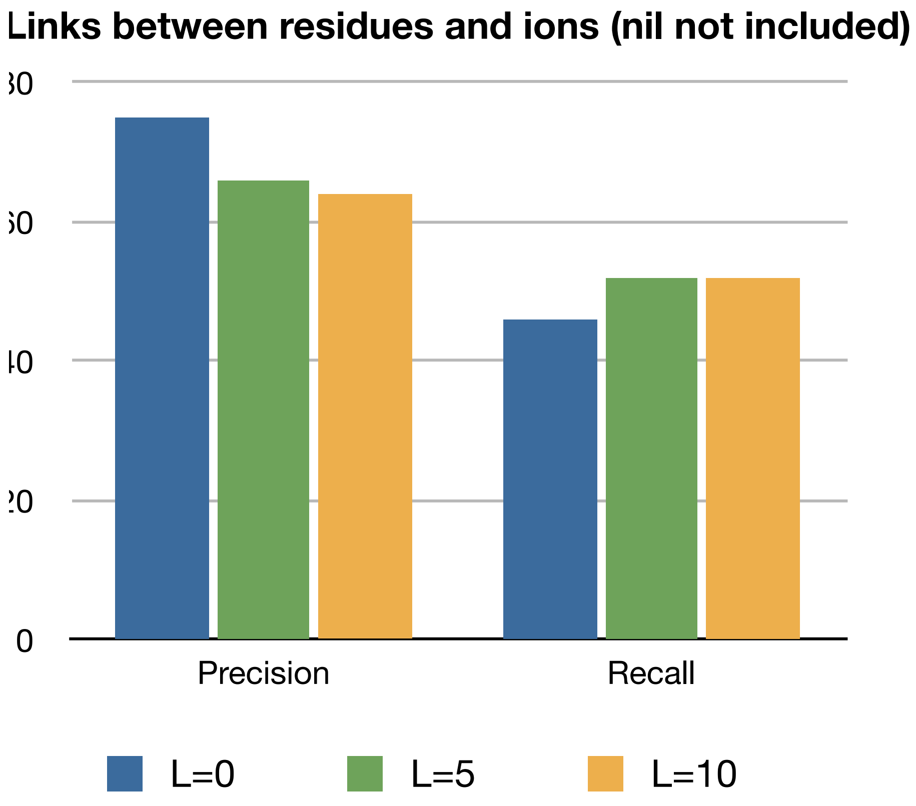
The metal binding site kernel

kernel between ligands, based on multiple alignment profiles

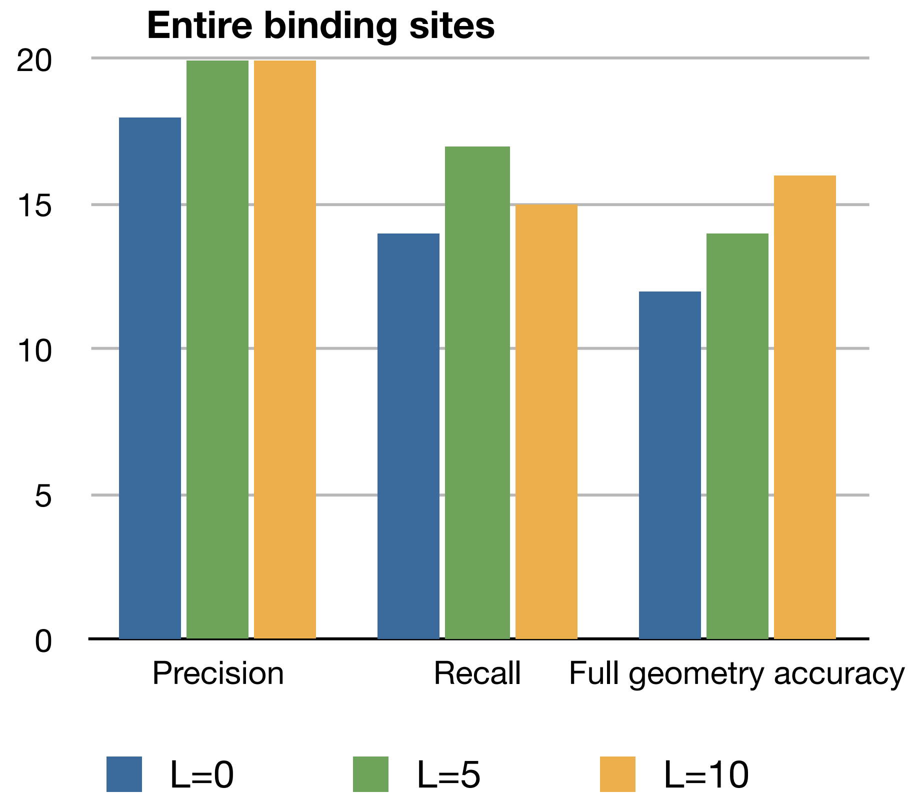
$$k_{\mathrm{mbs}}(\sigma_i(z), \sigma_j(z')) = \delta(|\sigma_i(z)|, |\sigma_j(z')|) \sum_{\ell=1}^{|\sigma_i(z)|} k_{\mathrm{res}}(x_i(\ell), x_j'(\ell))$$

similarity is zero unless the two
sites have the same # of
ligands

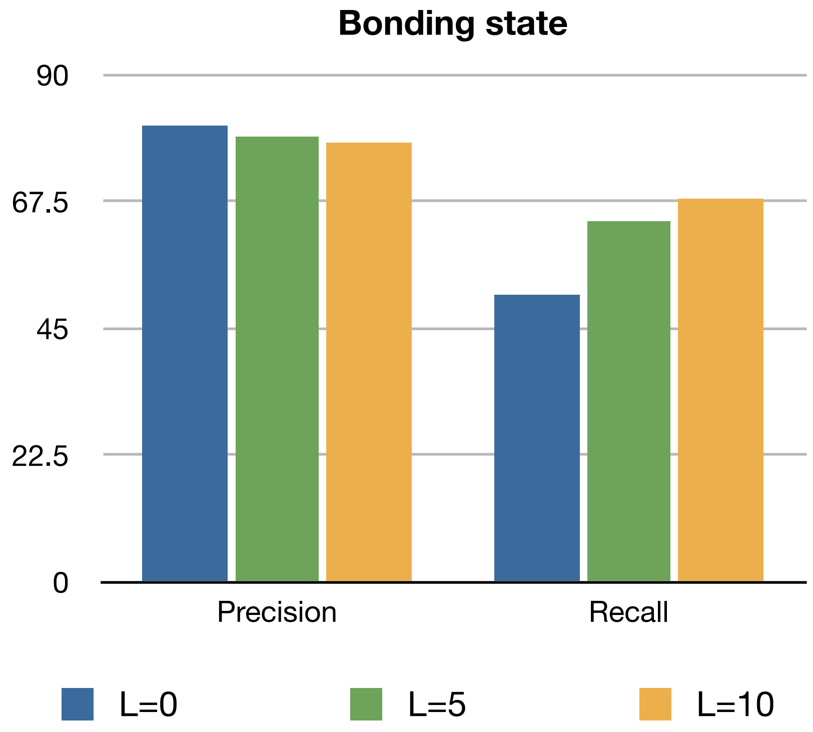
Results: ab initio prediction



Links between residues and ions (nil not included)

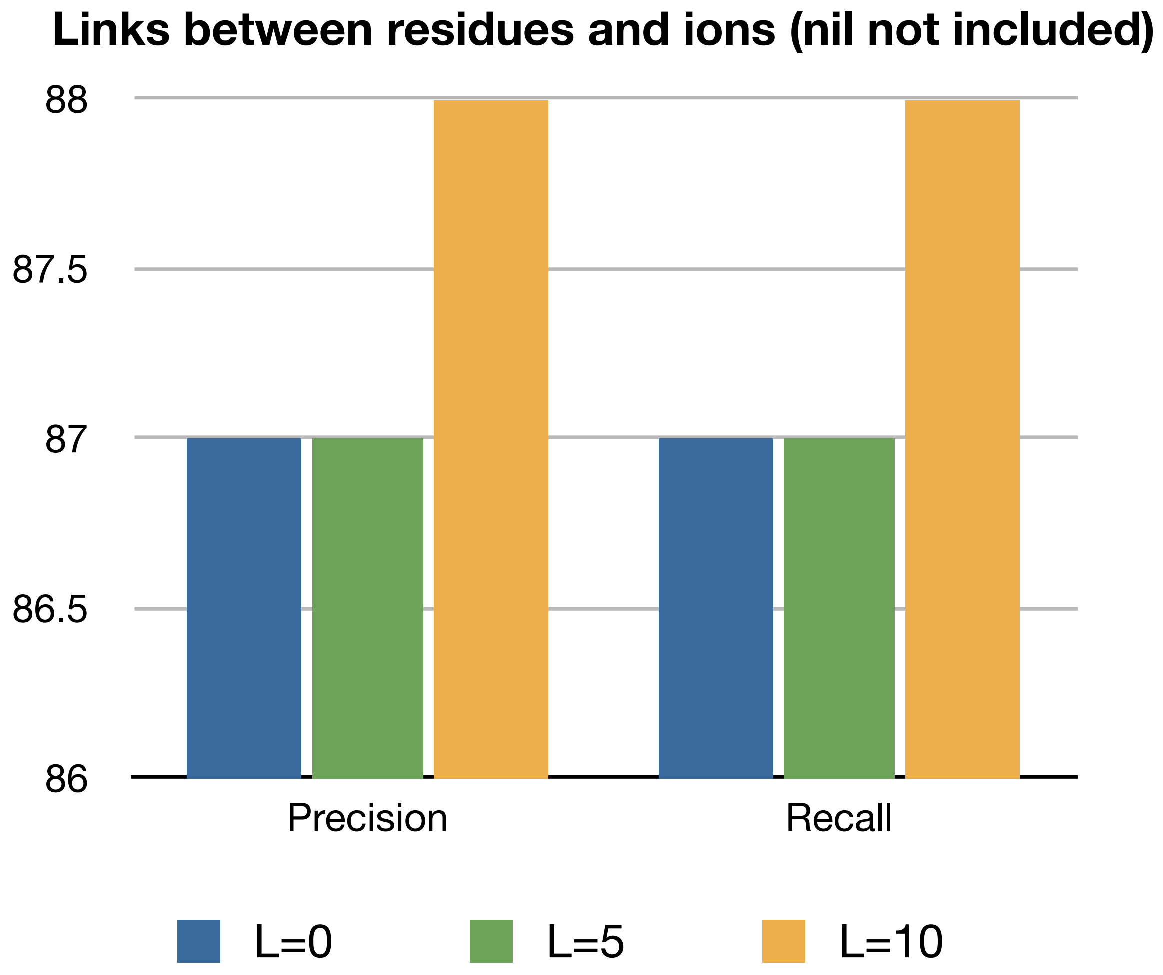


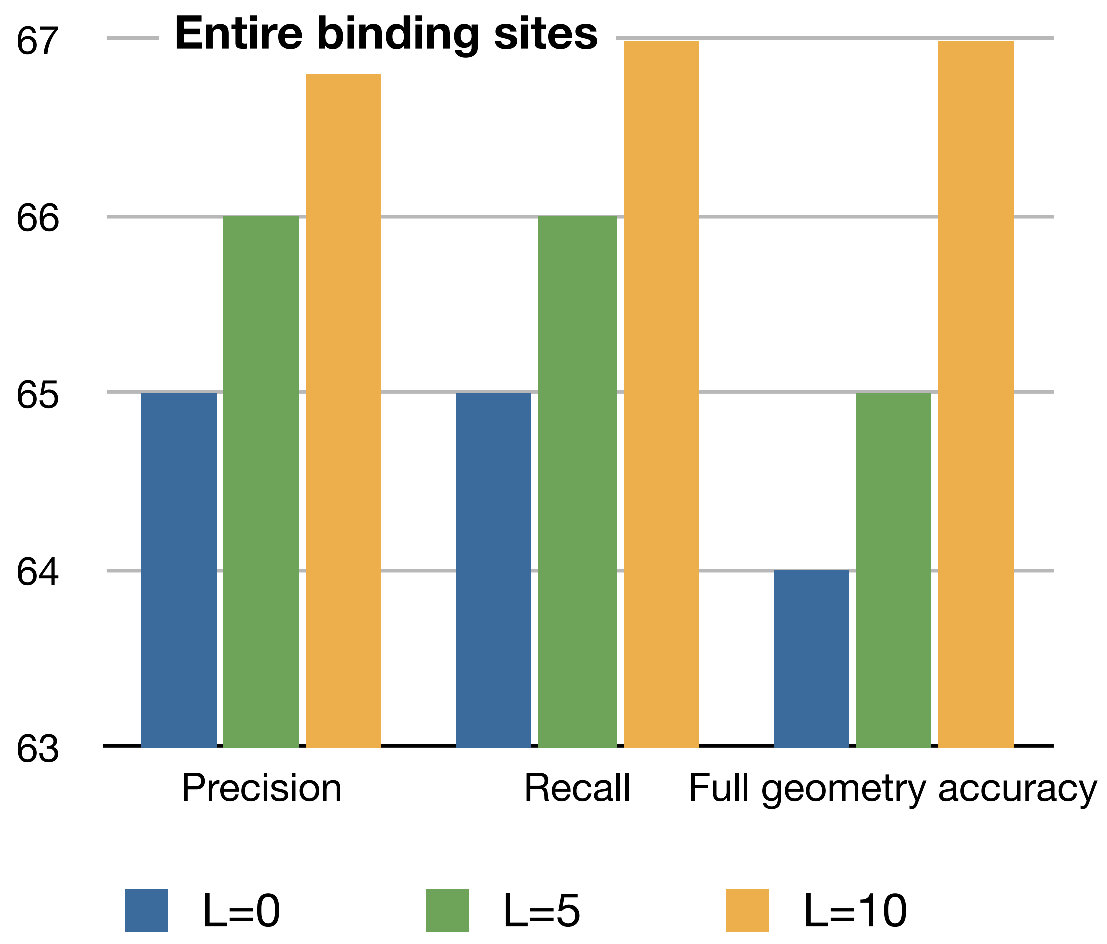
Entire binding sites



Bonding state

Results: bonding state given





Conclusions and ongoing work

- Metal binding can be successfully predicted
- First attempt to solve the binding geometry problem
- Greedy structured output algorithm potentially applicable to other domains
- Work in progress:
  - Improved kernels between metal binding geometries (about 5-7% improvement on precision/recall for site prediction)
  - Prediction of binding sites starting from 3D data

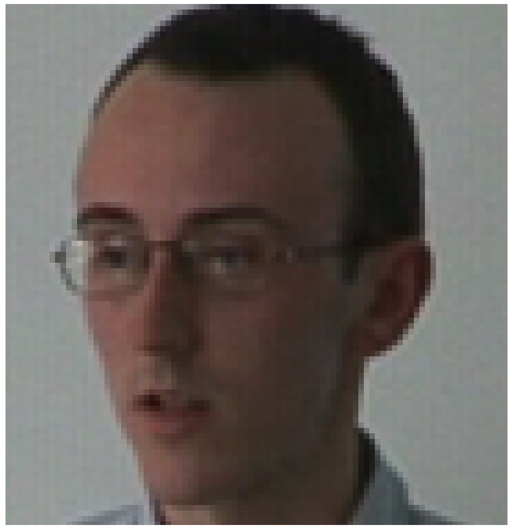
Acknowledgments

Andrea Passerini

(Università di Trento)

Marco Punta

(Columbia University)



Marco Lippi

(Università di Firenze)

Burkhard Rost

(Columbia University)