Maschinelles Lernen zur Genom-Sequenzanalyse

(GI-Kolloquium, Dagstuhl, 19. Mai 2009)

Sören Sonnenburg

Fraunhofer FIRST
Kekulestr. 7
12489 Berlin

TU Berlin
Franklinstr. 28/29
10587 Berlin

Friedrich-Miescher-Laboratorium
Spemannstr. 39
72076 Tübingen

Bedeutung der Informatik für die Lebenswissenschaften

Riesige Genom-Datenbanken

- 2006: Menschliches Genom sequenziert: Größe $\approx$ 3,3 Mrd.
  (vgl. Amöbe $\approx 1,4$ Billionen; Mikrosporid $\approx$ 2,2 Mio.)
- Mai 2009: 994 Organismen vollständig sequenziert;
  weltweit 4835 Genomprojekte

Bedeutung der Informatik für die Lebenswissenschaften

Riesige Genom-Datenbanken

- 2006: Menschliches Genom sequenziert: Größe $\approx$ 3,3 Mrd. (vgl. Amöbe $\approx 1,4$ Billionen; Mikrosporid $\approx$ 2,2 Mio.)
- Mai 2009: 994 Organismen vollständig sequenziert; weltweit 4835 Genomprojekte

Der Weg zum $1000-Genom

- 1990-2006 Sequenzierungskosten: $\approx$ 3 Mrd. US Dollar
- 2007 $\approx$ 100 Mio. US Dollar
- Anfang 2008 $\approx$ 1 Mio. US Dollar
- Ende 2008 $\approx$ 100.000 US Dollar

Datenanalyse erfordert intelligente und höchsteffiziente Informatikmethoden!

Interdisziplinäre Beiträge

Modellierung

- Maschinelles Lernen
- Ähnlichkeitsmaße für Strings
- Interpretierbarkeit von Lernmaschinen

Interdisziplinäre Beiträge

Modellierung

- Maschinelles Lernen
- Ähnlichkeitsmaße für Strings
- Interpretierbarkeit von Lernmaschinen

Algorithmen

- Large-Scale-Lernen (Lernen mit **10 Millionen** Beispielen)
- Effiziente Datenstrukturen
- Parallelisierte, effiziente Implementierung (**Beschleunigung** um **Faktor 100** beim Trainieren; **100.000** beim Anwenden)

Interdisziplinäre Beiträge

Modellierung

- Maschinelles Lernen
- Ähnlichkeitsmaße für Strings
- Interpretierbarkeit von Lernmaschinen

Algorithmen

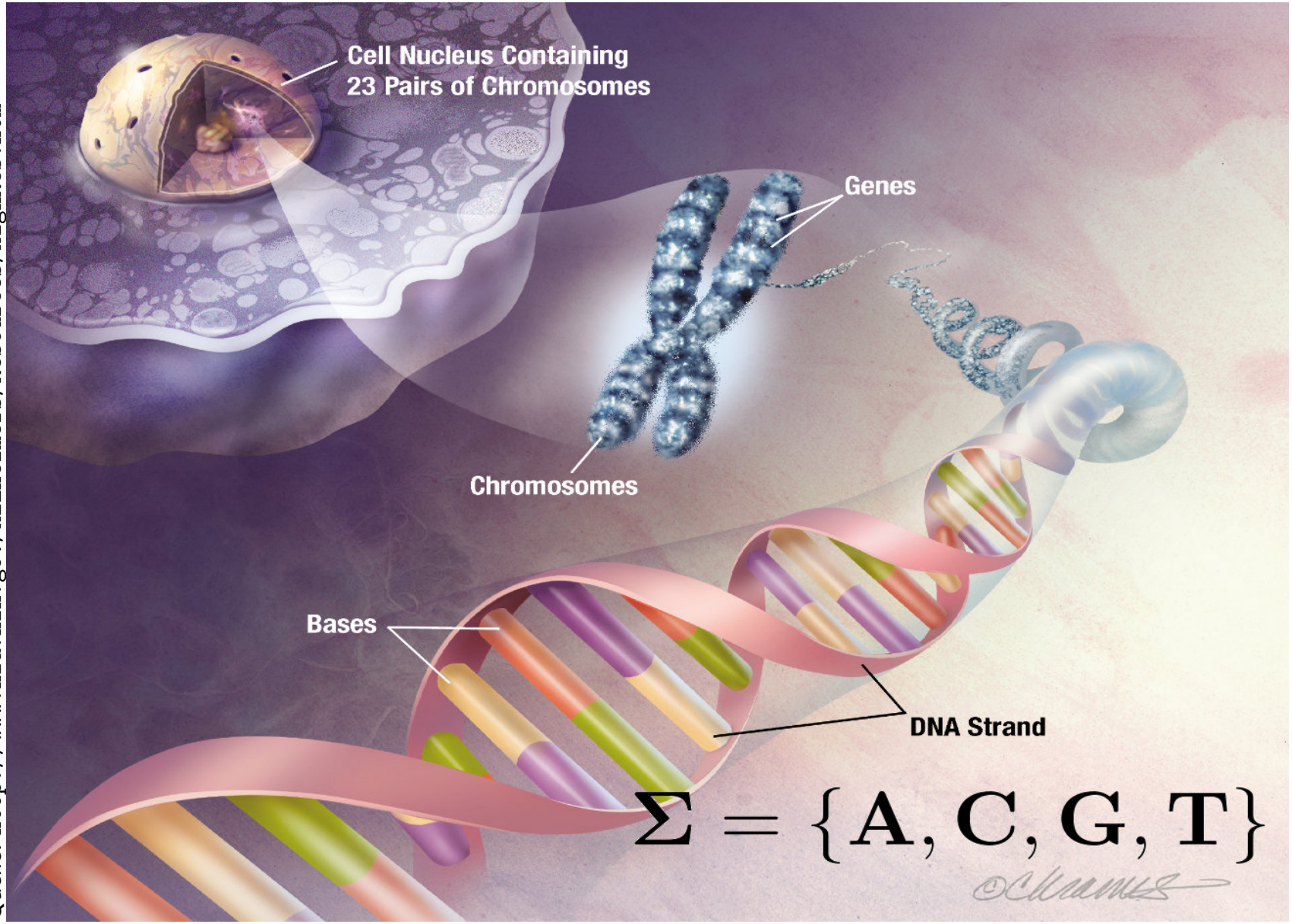
- Large-Scale-Lernen (Lernen mit **10 Millionen** Beispielen)
- Effiziente Datenstrukturen
- Parallelisierte, effiziente Implementierung (**Beschleunigung** um **Faktor 100** beim Trainieren; **100.000** beim Anwenden)

Anwendungen

- Genomanalyse (z.B. Signale, Gene)
- Computersicherheit (z.B. Hackerangriffe)
- Texterkennung (z.B. Spam)

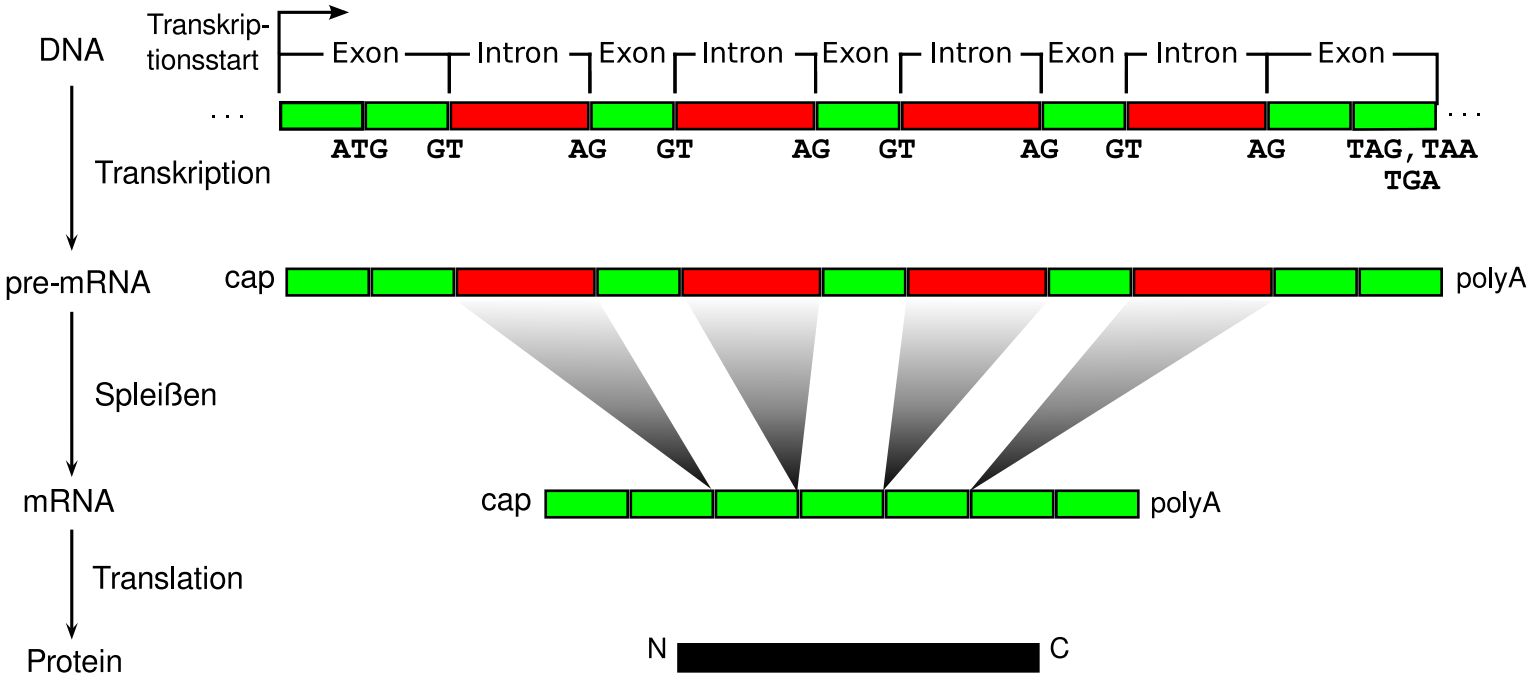
Deutlich genauere Vorhersagen $\Rightarrow$ **Halbierung der Fehlerraten**

Genom



Quelle: http://www.nia.nih.gov/Alzheimers/Resources/HighRes.htm

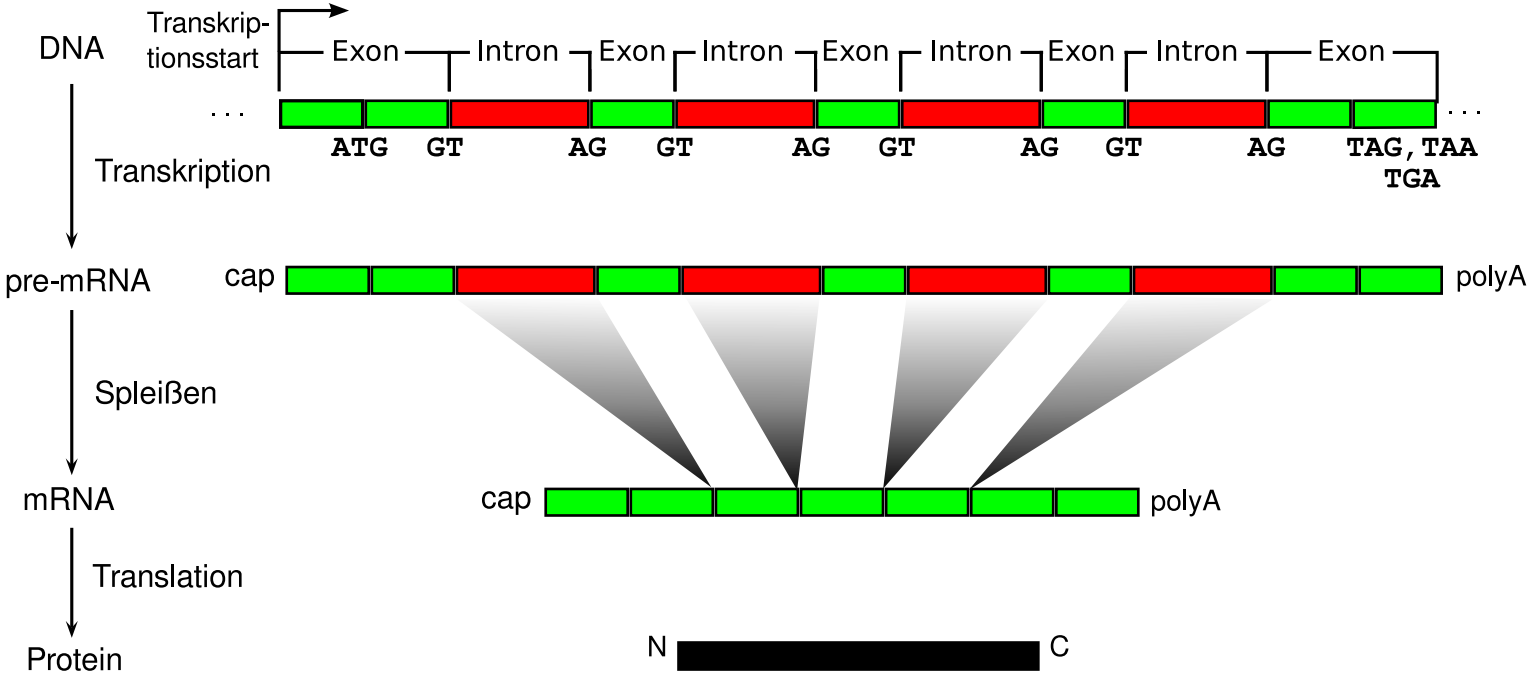
Genom-Signale



Genom-Signal-Erkennungsprobleme

- Start/Ende von Genen
- Donor-Spleißstellen (Exon-Intron-Grenzen)
- Akzeptor-Spleißstellen (Intron-Exon-Grenzen)

Genom-Signale



Genom-Signal-Erkennungsprobleme

- Akzeptor-Spleißstellen (Intron-Exon-Grenzen)

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

Klassifikation - Lernen anhand von Beispielen I

Gegeben:

Trainingsbeispiele $(\mathbf{x}_i, y_i)_{i=1}^N \in (\{A, C, G, T\}^L, \{-1, +1\})^N$

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

($\approx$ 1 Mrd. negative Sequenzen; $<$ 200.000 positive Sequenzen)

Klassifikation - Lernen anhand von Beispielen I

Gegeben:

Trainingsbeispiele $(\mathbf{x}_i, y_i)_{i=1}^N \in (\{A, C, G, T\}^L, \{-1, +1\})^N$

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

($\approx$ 1 Mrd. negative Sequenzen; $<$ 200.000 positive Sequenzen)

Gesucht:

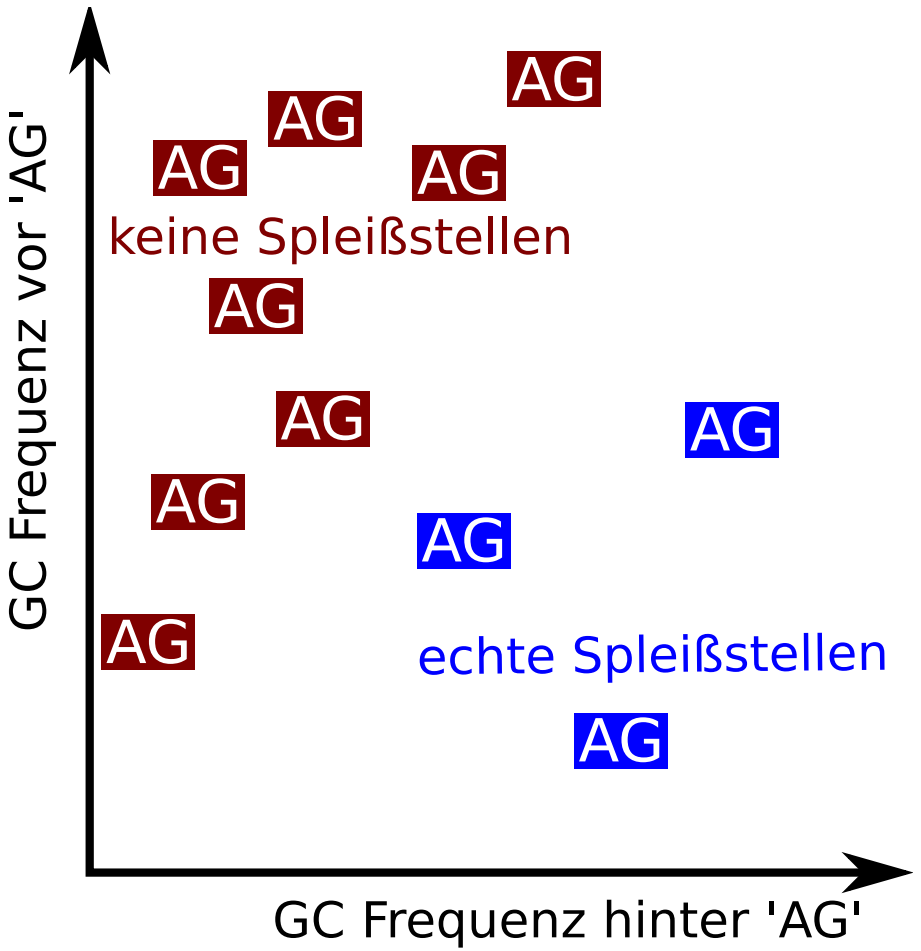
Funktion (Klassifikator) $f(\mathbf{x}) : \{A, C, G, T\}^L \mapsto \{-1, +1\}$

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

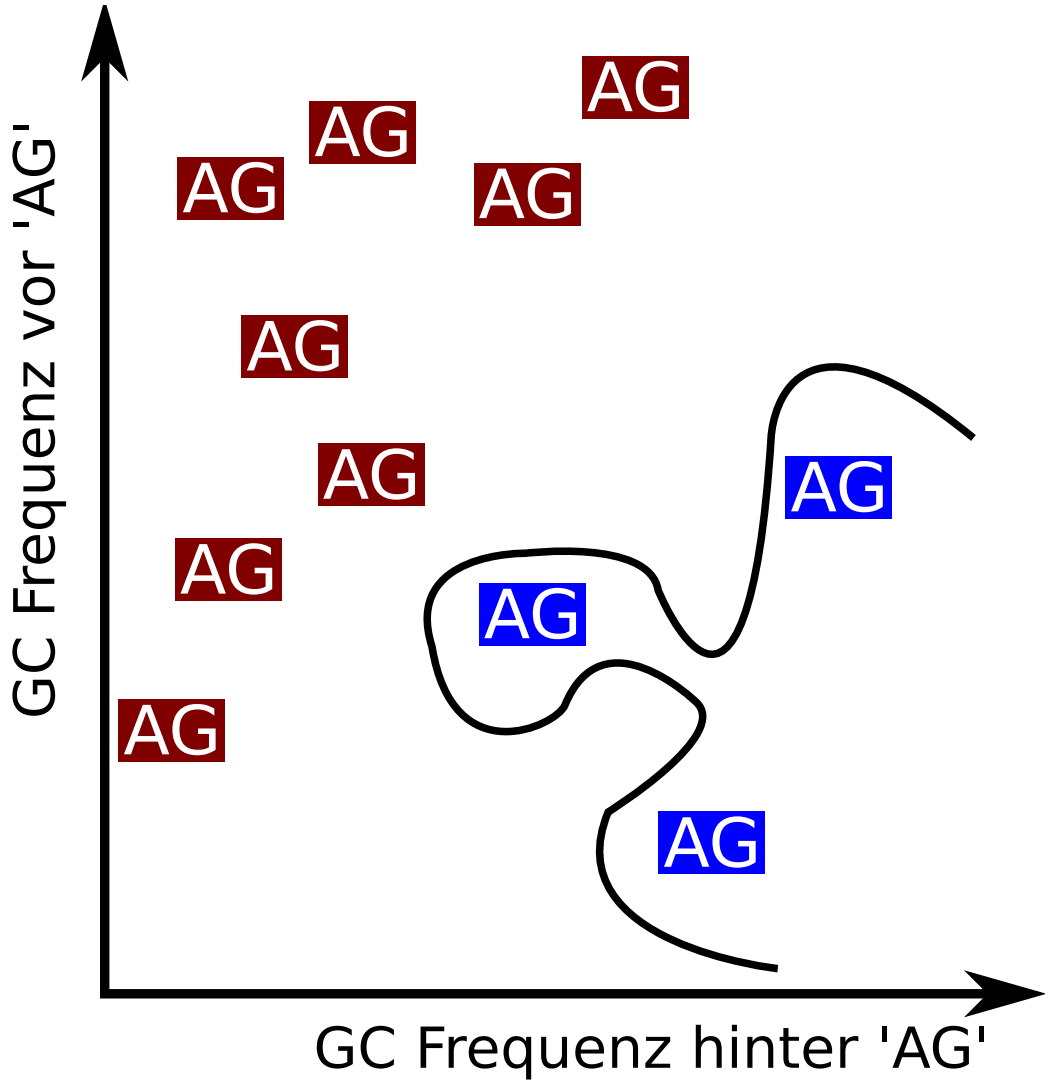
Ziel: Genaue Vorhersagen für das gesamte Genom

Klassifikation - Lernen anhand von Beispielen II

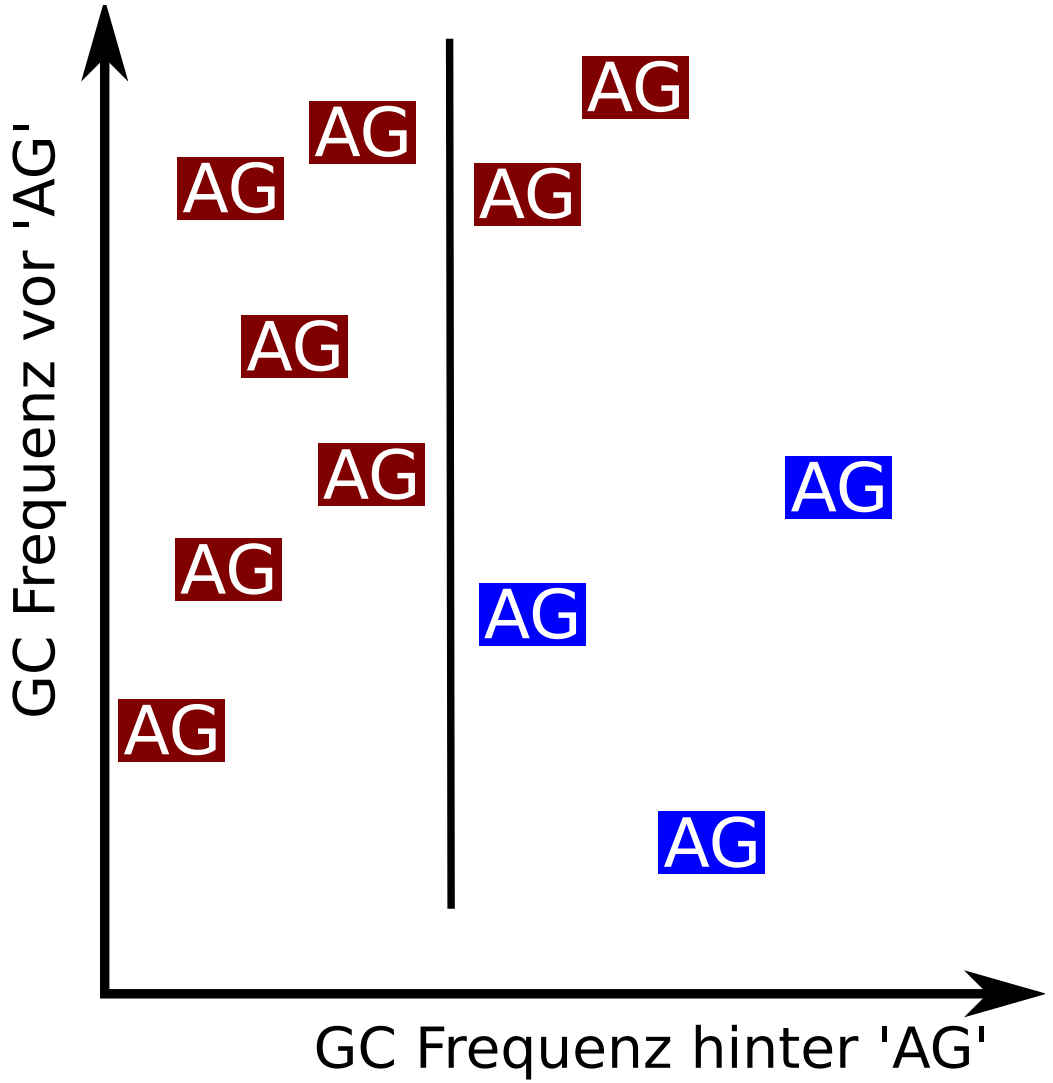
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG



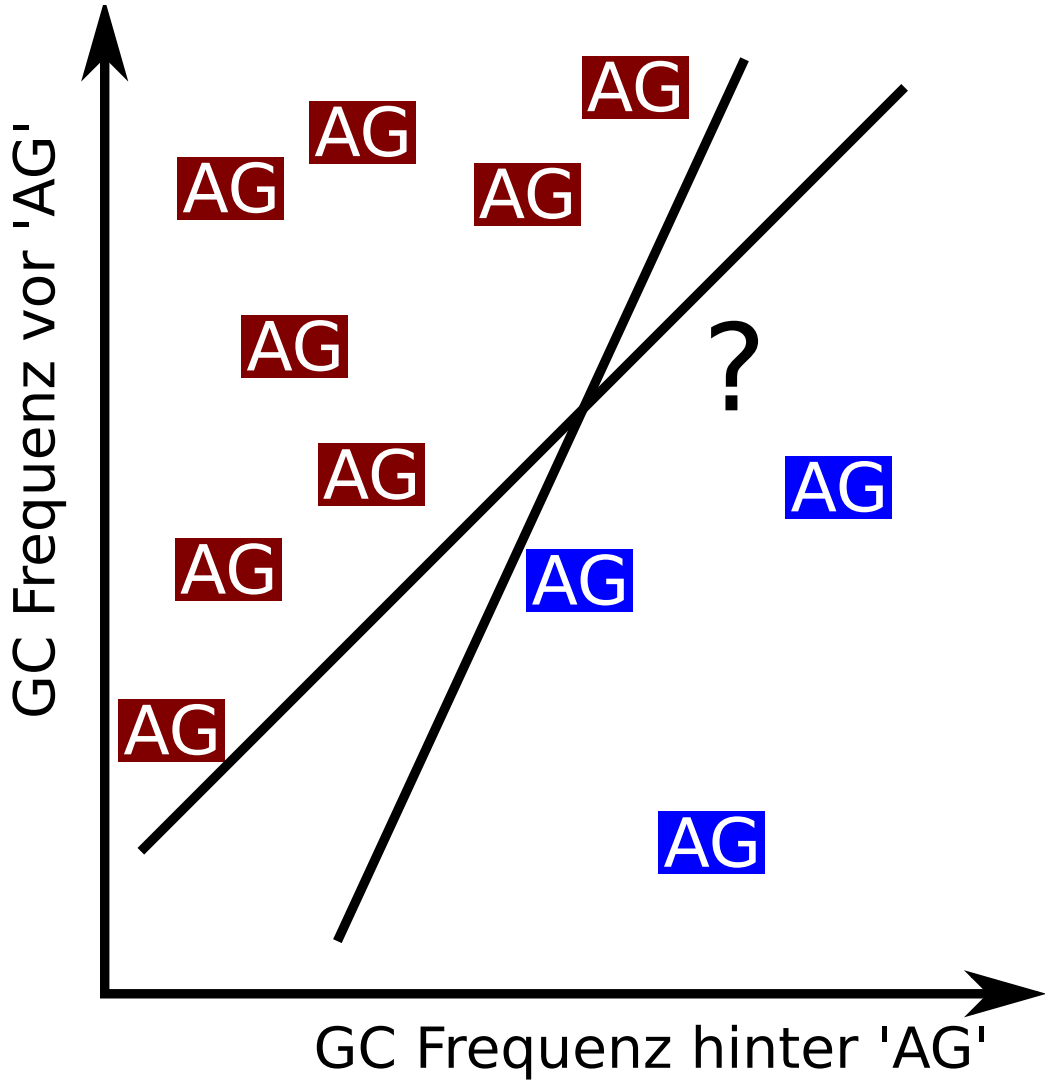
Klassifikation - Lernen anhand von Beispielen III



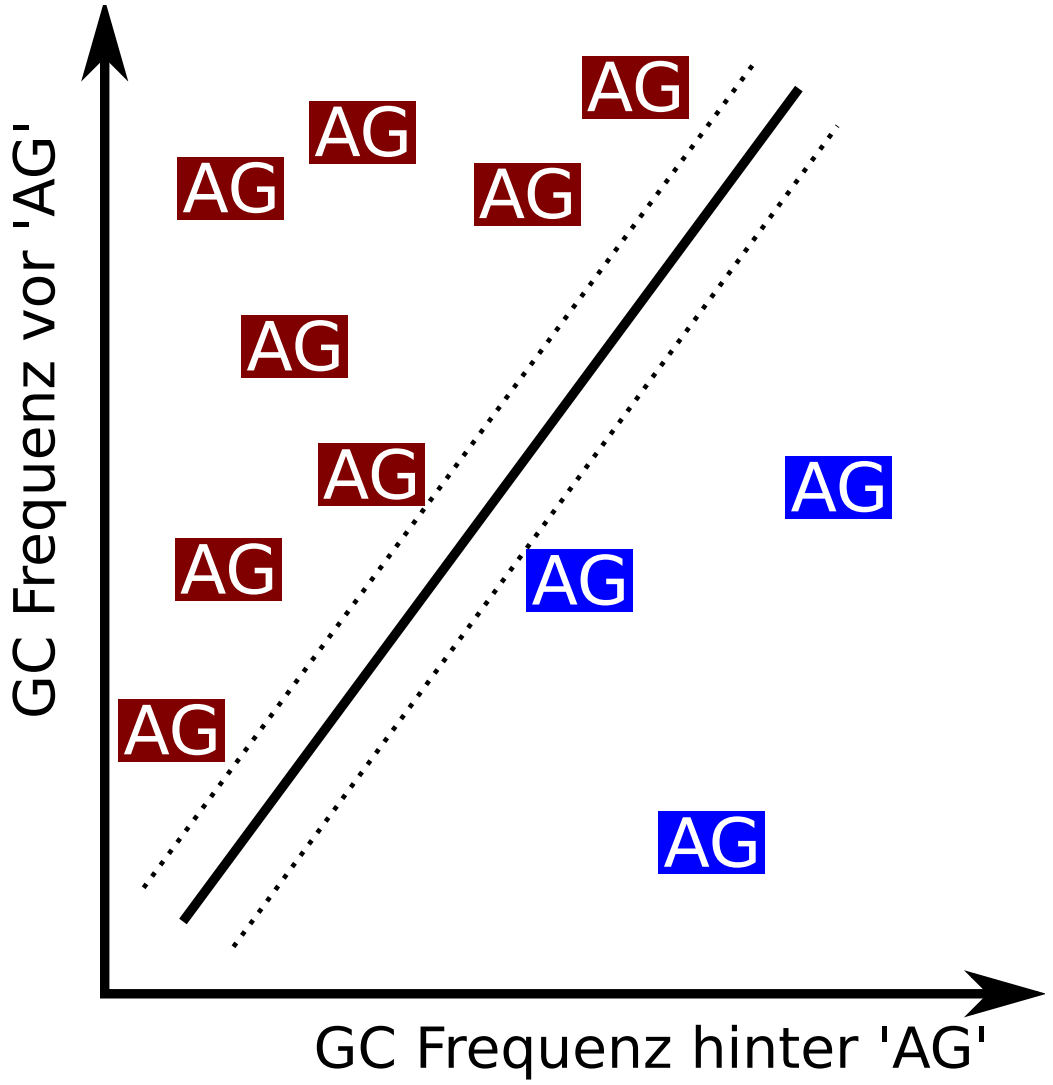
Klassifikation - Lernen anhand von Beispielen IV



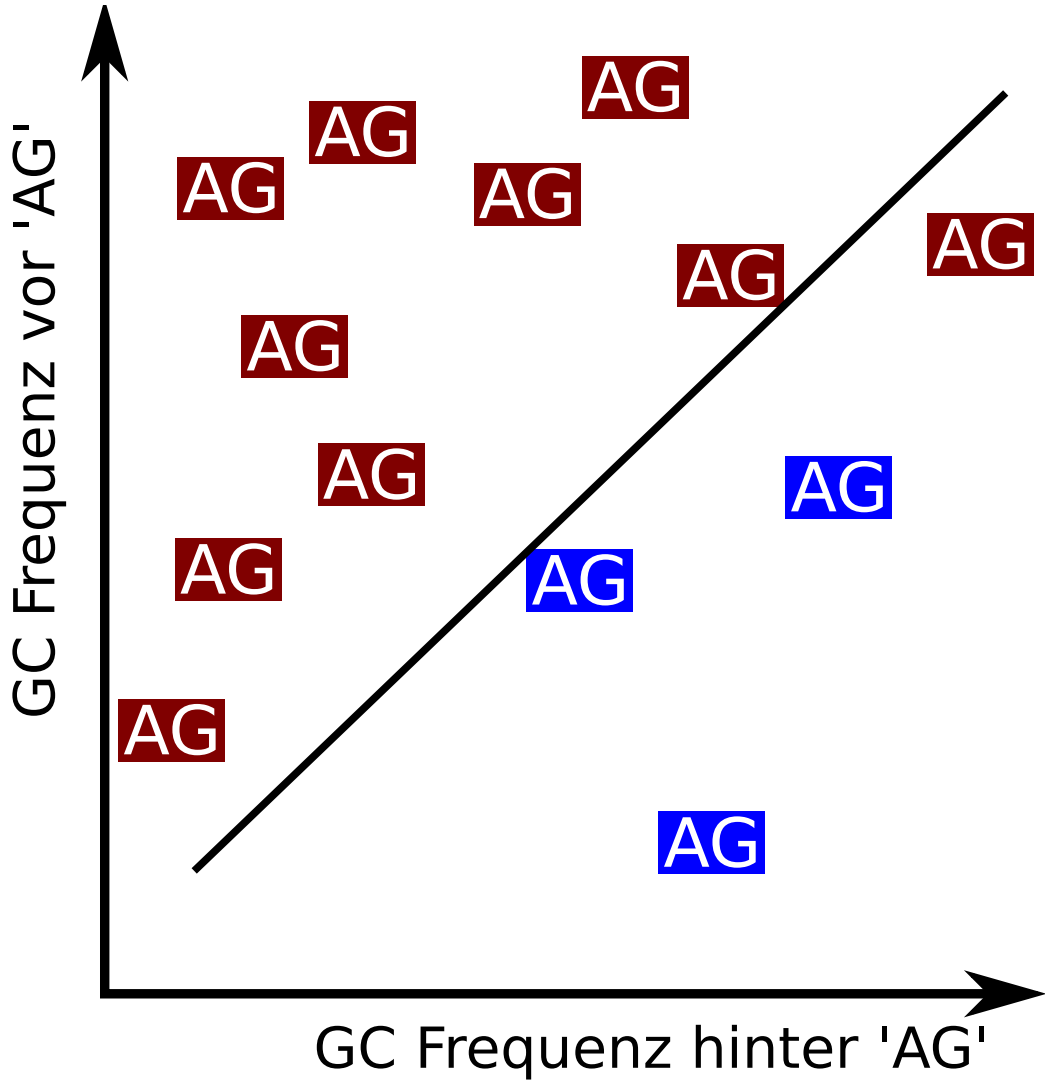
Support Vector Machines (SVMs) I



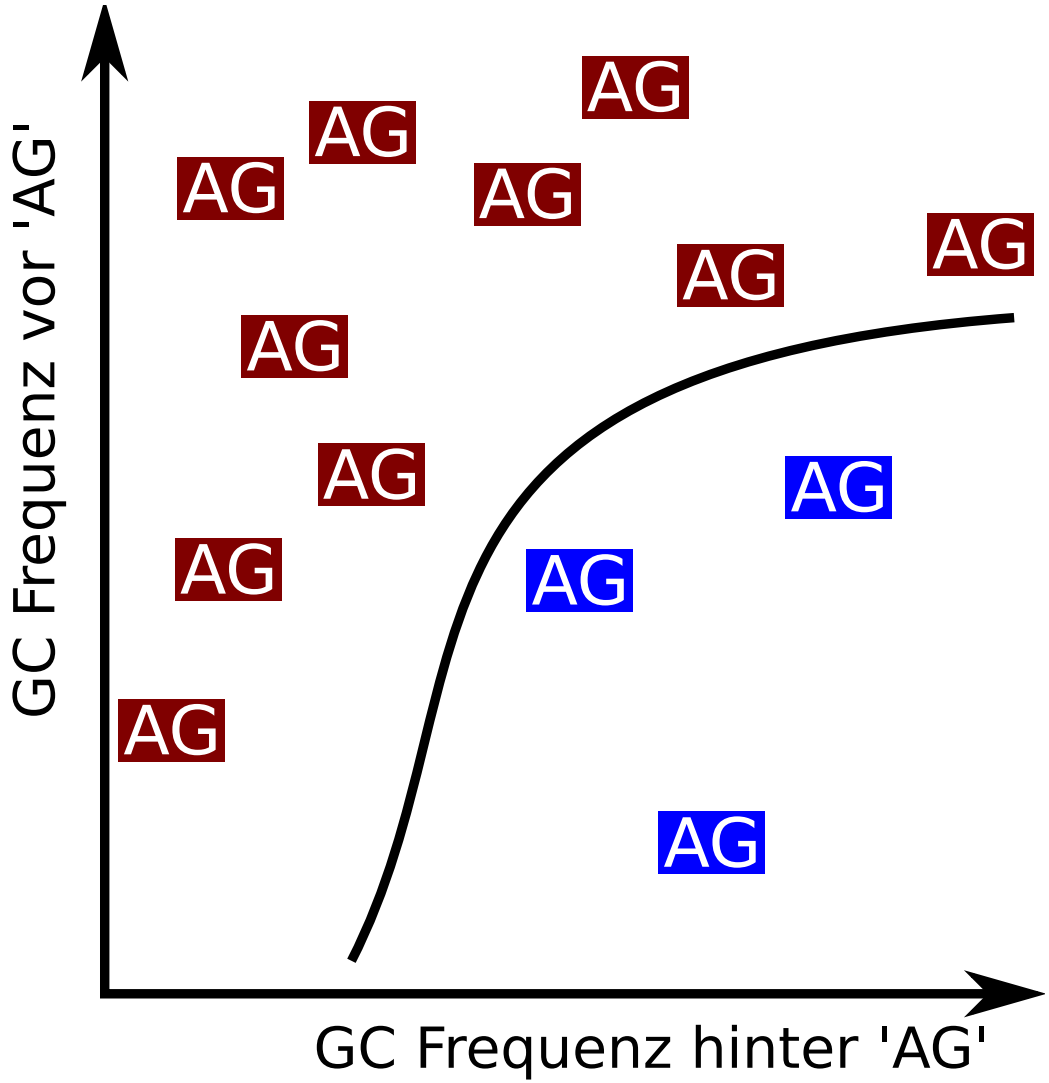
Support Vector Machines (SVMs) II



Support Vector Machines (SVMs) III



SVMs und Kerne



SVMs mit String-Kernen

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b\right)$$

SVMs mit String-Kernen

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

String-Kerne

- Spectrum Kernel (Leslie et al. 2002)

```
AAACAAAAACGTAACTAATCTTTTAGAGAGAACGTTTCAACCATTTTGAG
AAGATTAACTCATCACAGATTTCATTACATACAGATATAATTCAAAAATT
CACTCCCCAAATCAACGATATTTAAAAATCACTAACACATCCGTCTGTGC
```

SVMs mit String-Kernen

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right)$$

String-Kerne

- Spectrum Kernel (Leslie et al. 2002)

```
AAACAAAAACGTAACTAATCTTTTAGAGAGAACGTTTCAACCATTTTGAG
AAGATTAACTCATCACAGATTTCATTACATACAGATATAATTCAAAAATT
CACTCCCCAAATCAACGATATTTAAAAATCACTAACACATCCGTCTGTGC
```

- Weighted Degree Kernel (Sonnenburg et al. 2005)

```
AAACAAATAAGTAACTAATCTTTTAAGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAACATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAATTCACTAACACATCCGTCTGTGCC
```

SVMs mit String-Kernen

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b\right)$$

String-Kerne

- Spectrum Kernel (Leslie et al. 2002)

```
AAACAAAAACGTAACTAATCTTTTAGAGAGAACGTTTCAACCATTTTGAG
AAGATTAACTCATCACAGATTTCATTACATACAGATATAATTCAAAAATT
CACTCCCCAAATCAACGATATTTAAAAATCACTAACACATCCGTCTGTGC
```

- Weighted Degree Kernel (Sonnenburg et al. 2005)

```
AAACAAATAAGTAACTAATCTTTTAAGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAACATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAATTCACTAACACATCCGTCTGTGCC
```

- Weighted Degree Kernel *mit shift* (Rätsch, Sonnenburg et al. 2005)

```
AAACAAATAAGTAACTAATCTTTTAAAGAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTCATTAAATACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTAAATTTCACTAACACATCCGTCTGTGC
```

Geeignete Kernwahl erlaubt es, Vorwissen zu integrieren.

SVMs mit String-Kernen

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b\right)$$

String-Kern-Merkmalsraum

- z.B. alle Substrings der Länge k an jeder Position
- In typischen Anwendungen 10^{14} dimensional!
- Effizient im Eingaberaum berechenbar

SVM selbst bei linearer Kern-Komplexität zu teuer!

Beschleunigung von SVMs mit String-Kernen

Beschleunigung von Linearkombinationen
(Sonnenburg et al., 2005, Sonnenburg et al., 2007):

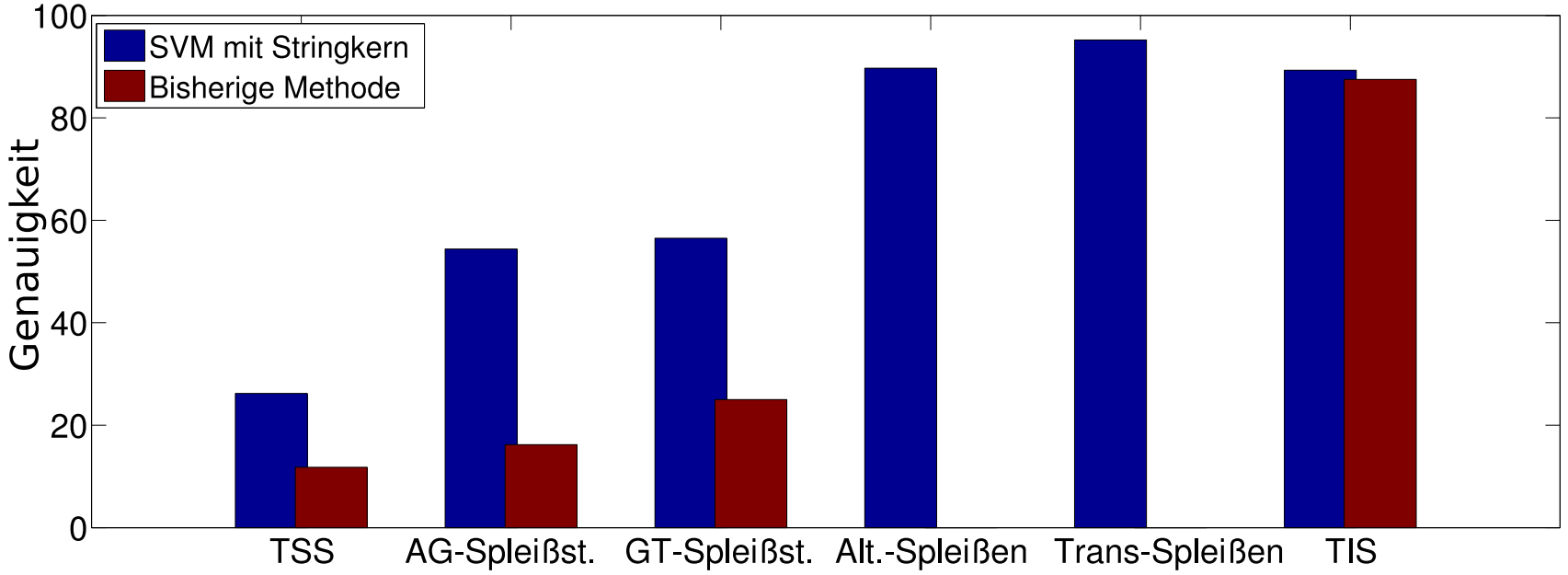
$$f(x_i) = \sum_{j \in W} \alpha_j y_j \, \mathrm{k}(x_i, x_j) \quad \forall i = 1, \ldots, 1.000.000.000$$

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG

1. Konstruktion einer effizienten Datenstruktur (z.B. Suffix-Trie)
2. Lookup-Operation: Datenstruktur (Baum) durchlaufen
3. Parallelisierung

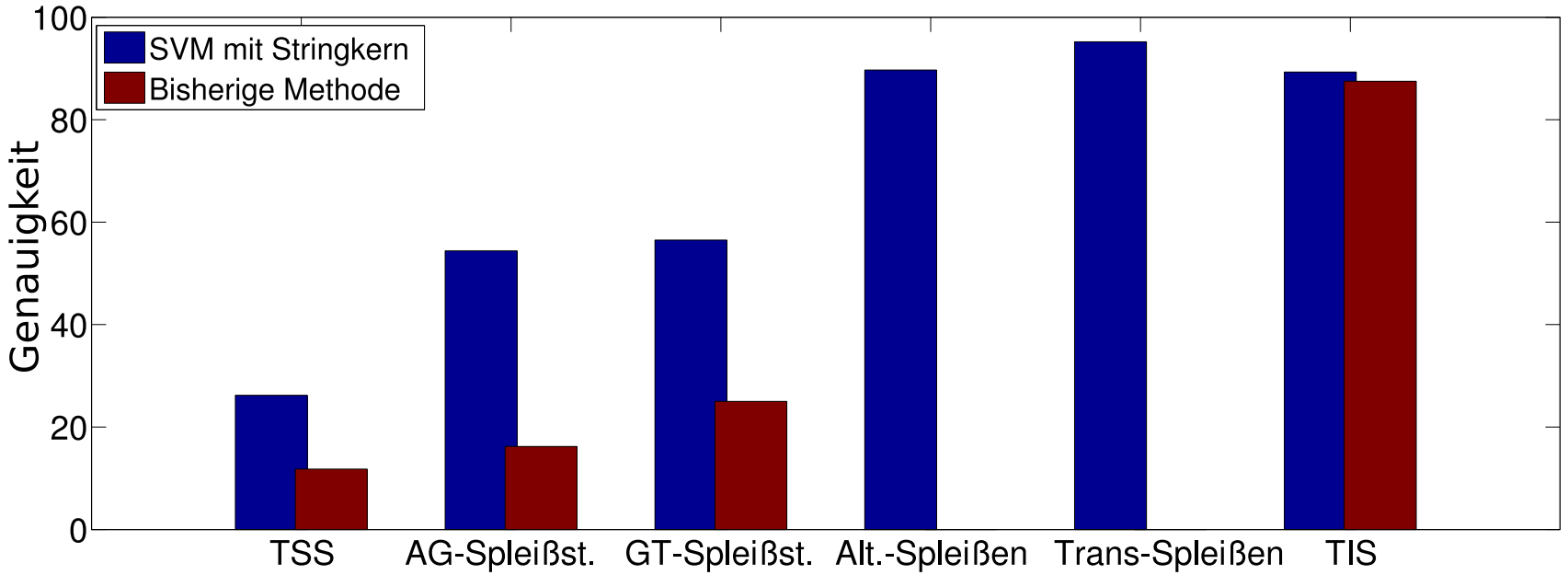
Training von String-Kern-SVMs auf **10 Millionen** Beispielen
Schnell! Faktor 100 beim Training; 100.000 beim Anwenden

Ein vielseitiger Ansatz



- Transkriptionsstart (Sonnenburg et al., 2006/Down et al. 2002)
- Akzeptor-Spleißstelle (Sonnenburg et al., 2007/Baten et al. 2006)
- Donor-Spleißstelle (Sonnenburg et al., 2007/Baten et al. 2006)
- Alternatives Spleißen (Rätsch, Sonnenburg et al., 2005/-)
- Trans-Spleißen (Schweikert, Sonnenburg et al., 2009/-)
- Translation-Initiation (Sonnenburg et al., 2008/Saeys et al., 2007)

Ein vielseitiger Ansatz

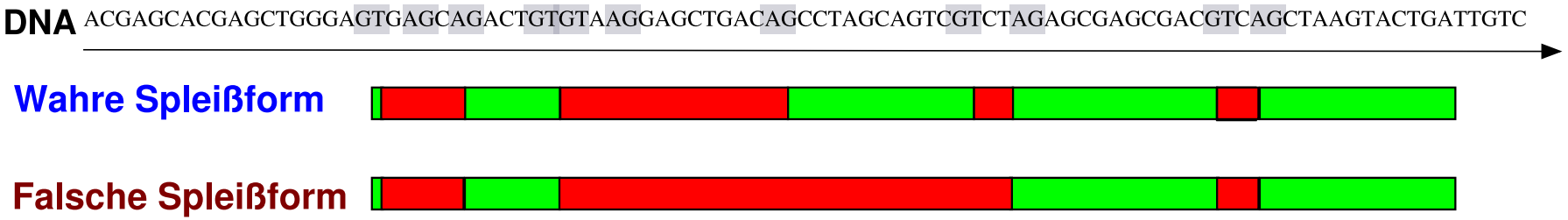


- Transkriptionsstart (Sonnenburg et al., 2006/Down et al. 2002)
- Akzeptor-Spleißstelle (Sonnenburg et al., 2007/Baten et al. 2006)
- Donor-Spleißstelle (Sonnenburg et al., 2007/Baten et al. 2006)
- Alternatives Spleißen (Rätsch, Sonnenburg et al., 2005/-)
- Trans-Spleißen (Schweikert, Sonnenburg et al., 2009/-)
- Translation-Initiation (Sonnenburg et al., 2008/Saeys et al., 2007)

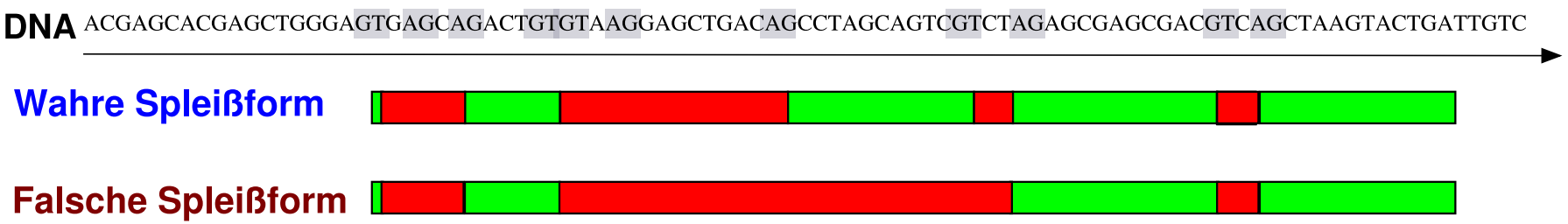
In unabhängiger Evaluierung von Abeel et al. ISMB, 2009

TSS Erkenner (ARTS) Sieger in Evaluation von 17 Methoden

mSplicer – Spleißformvorhersage (Rätsch, Sonnenburg et al., 2007)

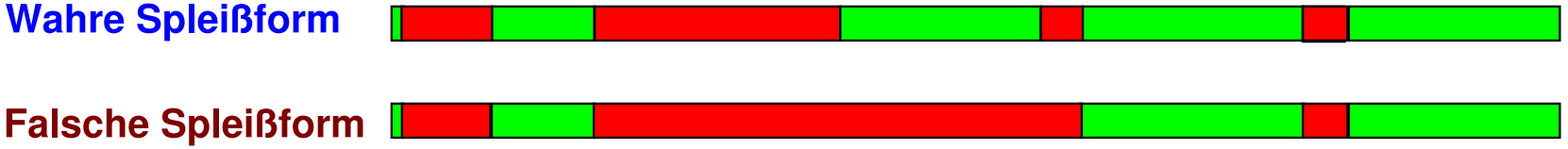


mSplicer – Spleißformvorhersage (Rätsch, Sonnenburg et al., 2007)

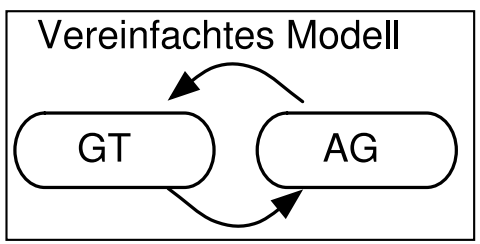


Schritt 1: SVM Signal Vorhersagen

mSplicer – Spleißformvorhersage (Rätsch, Sonnenburg et al., 2007)

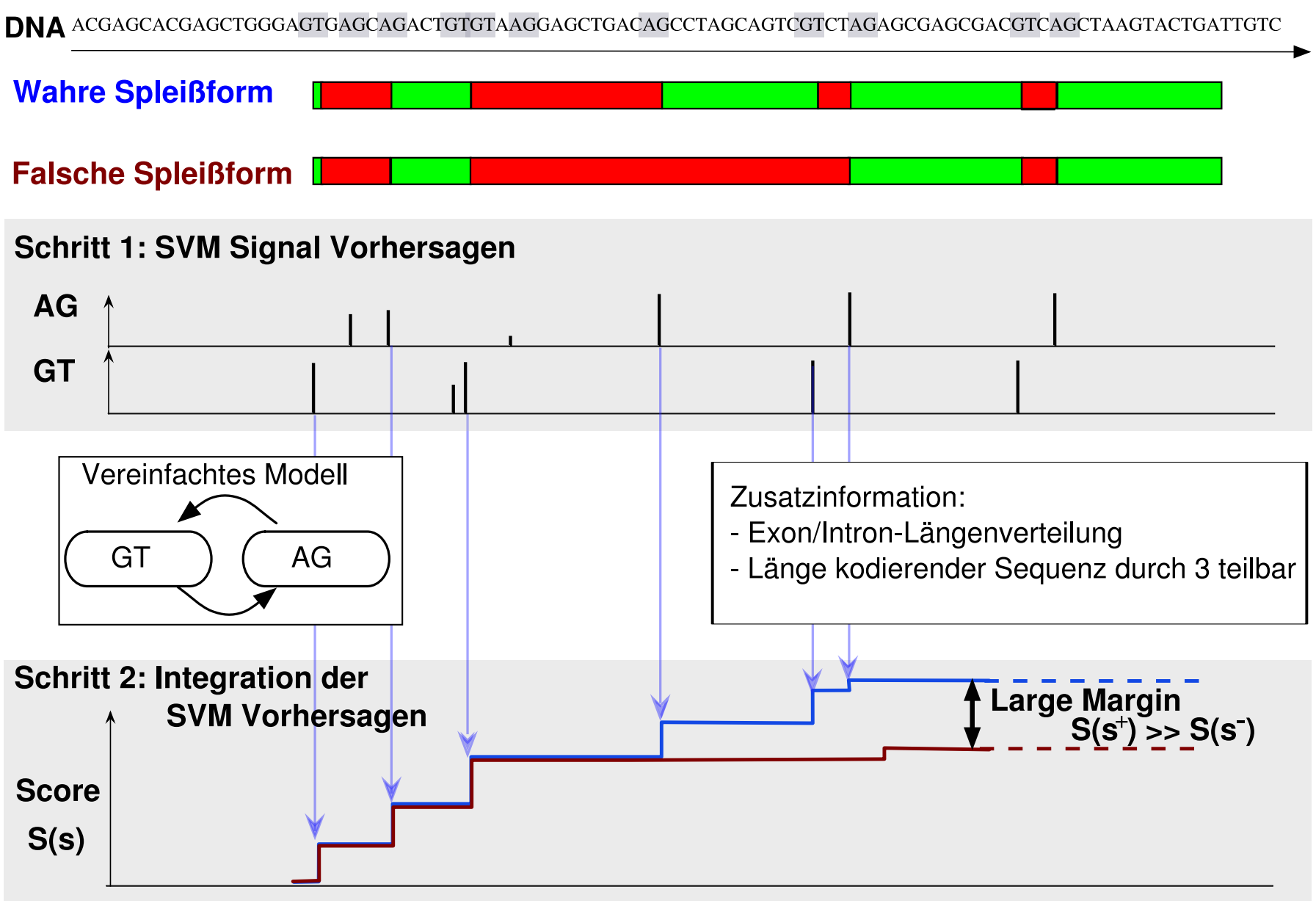


Schritt 1: SVM Signal Vorhersagen

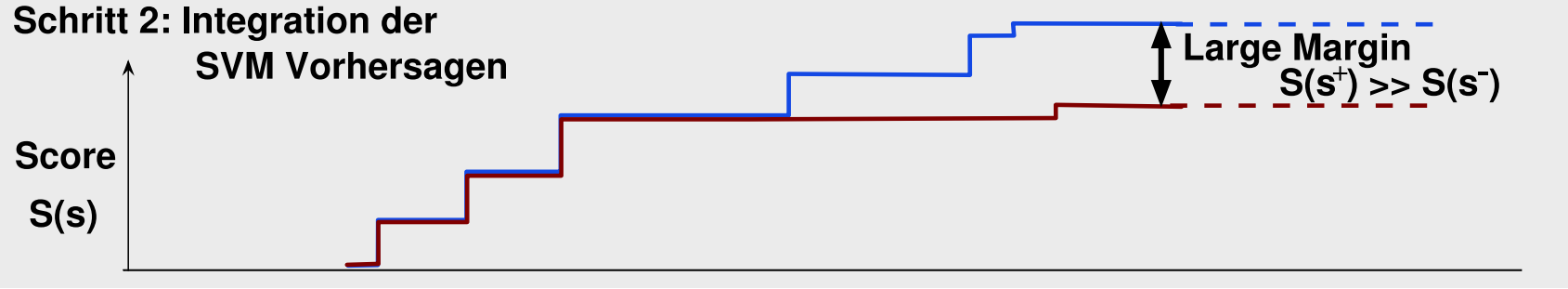


Zusatzinformation:
- Exon/Intron-Längenverteilung
- Länge kodierender Sequenz durch 3 teilbar

mSplicer – Spleißformvorhersage (Rätsch, Sonnenburg et al., 2007)



Diskriminatives Strukturlernen

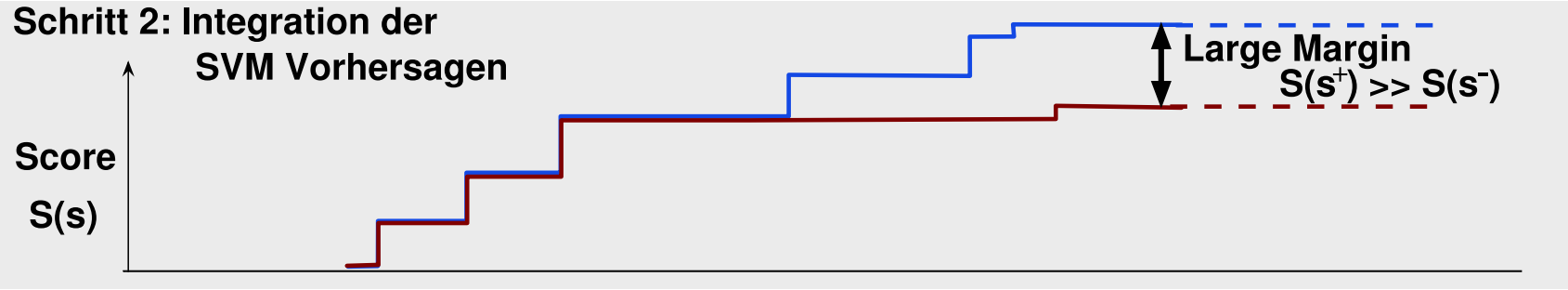


Vereinfachtes Optimierungs-Problem (Rätsch, Sonnenburg, 2007)

$$\text{Minimiere}_{\theta \in \mathbb{R}^d} \quad \mathbf{P}(\theta)$$

$$\text{bzgl.} \quad S_\theta(\mathbf{s}_i^+) - S_\theta(\mathbf{s}_i^-) \geq 1, \quad \forall \mathbf{s}_i^- \neq \mathbf{s}_i^+, \; i = 1, \ldots, \ell$$

Diskriminatives Strukturlernen



Vereinfachtes Optimierungs-Problem (Rätsch, Sonnenburg, 2007)

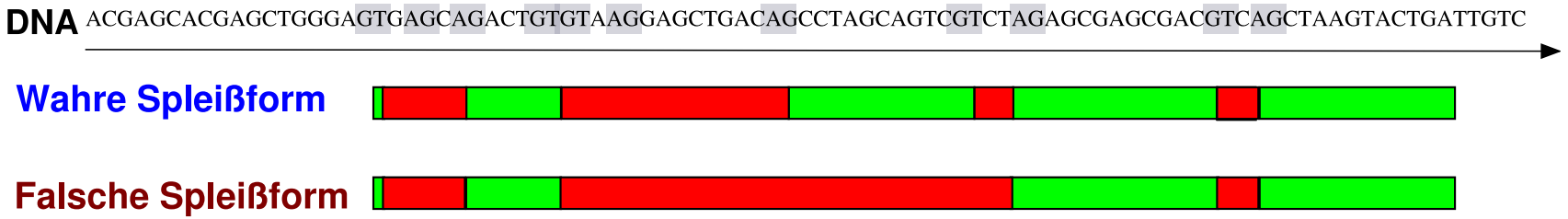
$$\underset{\theta \in \mathbb{R}^d}{\text{Minimiere}} \quad \mathbf{P}(\theta)$$

$$\text{bzgl.} \quad S_\theta(\mathbf{s}_i^+) - S_\theta(\mathbf{s}_i^-) \geq 1, \quad \forall \mathbf{s}_i^- \neq \mathbf{s}_i^+, \; i = 1, \ldots, \ell$$

Lösung durch Column Generation

1. Finde maximal verletzte Nebenbedingung oder terminiere
2. Löse Optimierungsproblem für kleine Anzahl Nebenbedingungen
3. Wiederhole bis terminiert

Rekord-Ergebnisse auf Modellorganismus

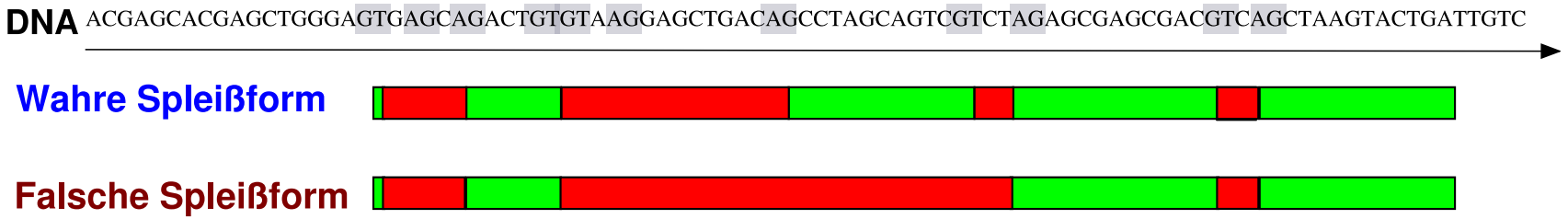


Drastische Reduktion der Fehlerraten

- Halbierung des Fehlers zur Spleißform-Vorhersage auf 4.8%
- 98.9% Exons erkannt mit 0.8% Falschpositivrate

$\Rightarrow$ **Jetzt in der offiziellen Annotation**

Rekord-Ergebnisse auf Modellorganismus



Drastische Reduktion der Fehlerraten

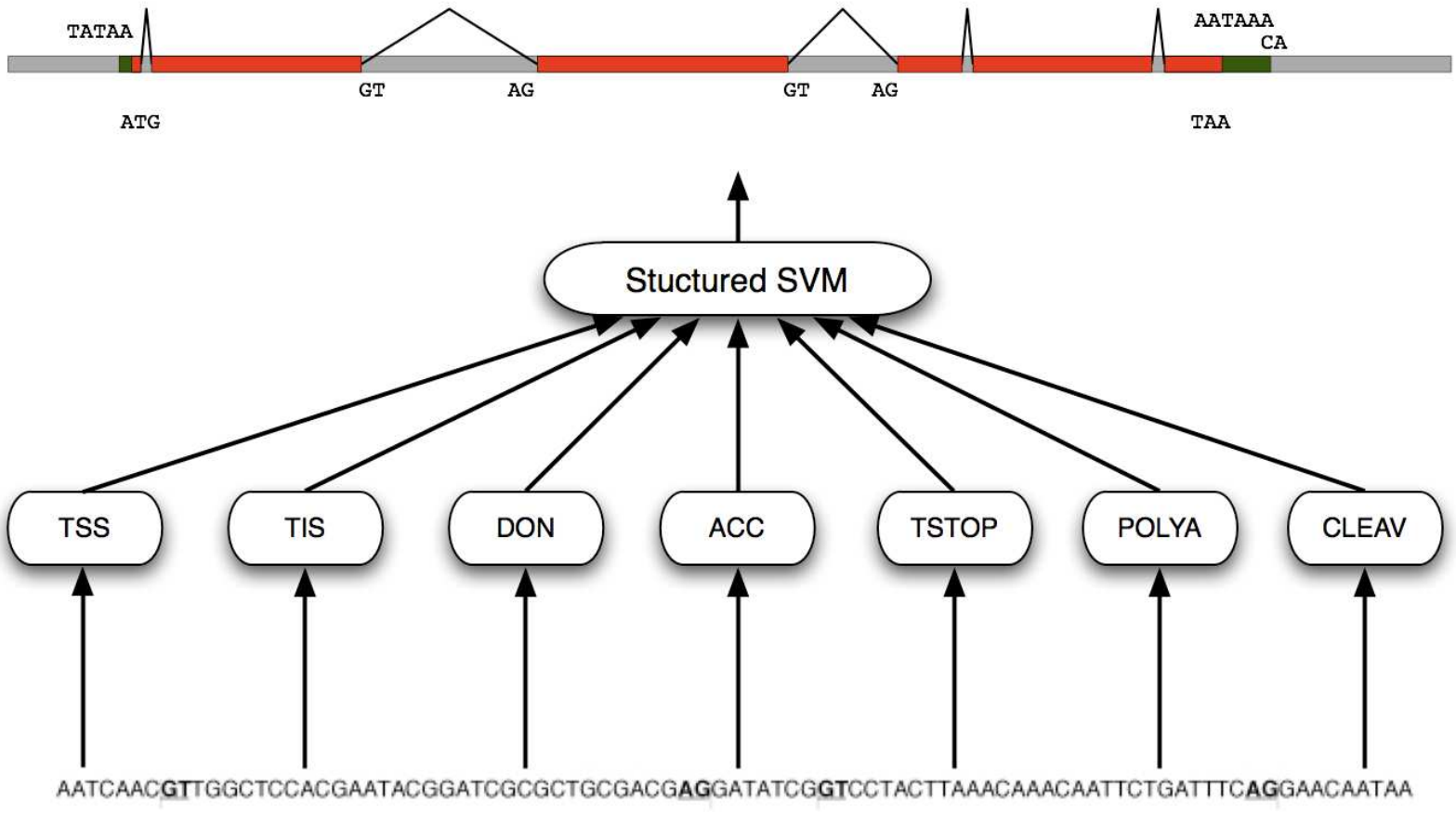
- Halbierung des Fehlers zur Spleißform-Vorhersage auf 4.8%
- 98.9% Exons erkannt mit 0.8% Falschpositivrate

$\Rightarrow$ **Jetzt in der offiziellen Annotation**

Effiziente Implementierung der Algorithmen frei verfügbar

- `http://www.shogun-toolbox.org`
- `http://www.msplicer.org`
- `http://mloss.org`

Gensuchmaschine mGene



⇒ Gewinner in *ab initio* Kategorie nGASP Wettbewerb
⇒ In der offiziellen Annotation für Nematoden
⇒ Web-Interface `http://www.mgene.org`

Biologischer Erkenntnissgewinn?

Multiple Kernel Learning (Sonnenburg et al., 2006)

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{j=1}^{M} \beta_j \sum_{i=1}^{N} y_i \alpha_i \mathrm{k}_j(\mathbf{x}, \mathbf{x}_i) + b\right), \quad \beta_j \geq 0$$

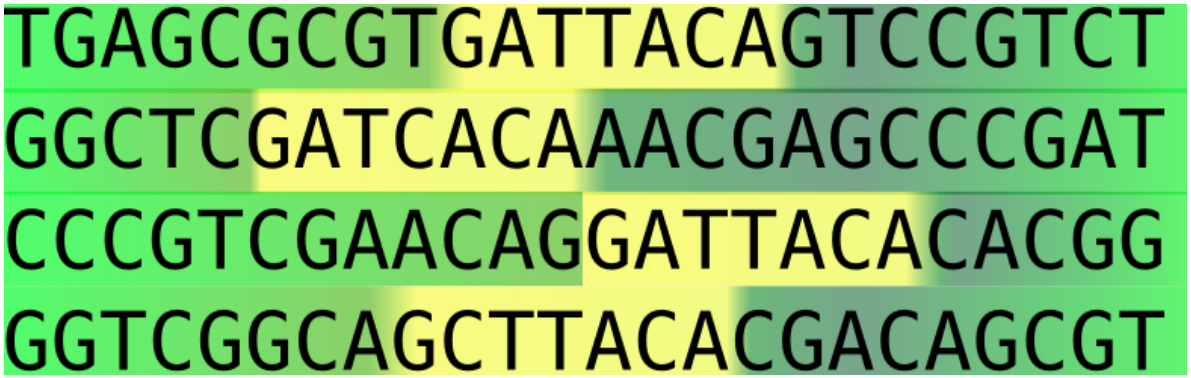
- Ein Kern je Position in der Sequenz
- Entwicklung eines schnellen Trainingsalgorithmus
- **Training auf 1 Million Datenpunkten (vorher 1000)**

Positional Oligomer Importance Matrices (Sonnenburg et al, 2008)

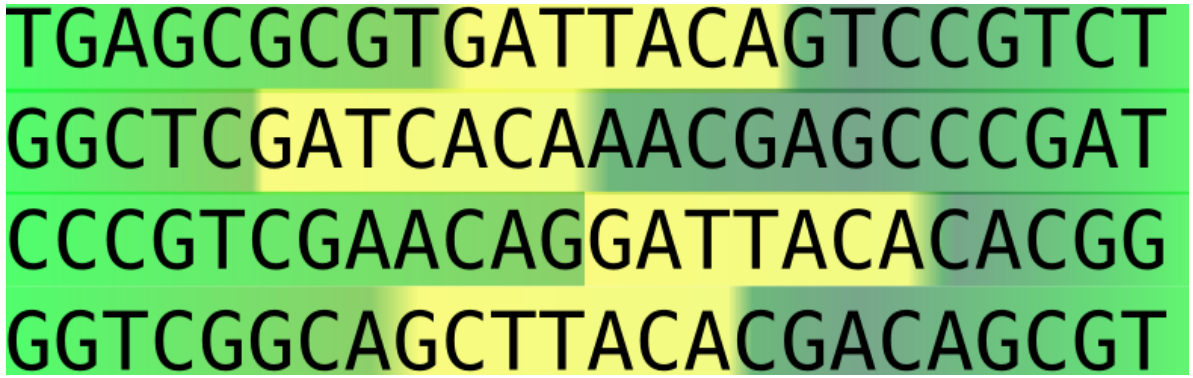
$$Q(\mathbf{z}, j) := \mathbb{E}\left[\, s(\mathbf{x}) \,|\, \mathbf{x}[j] = \mathbf{z} \,\right] - \mathbb{E}\left[\, s(\mathbf{x}) \,\right]$$

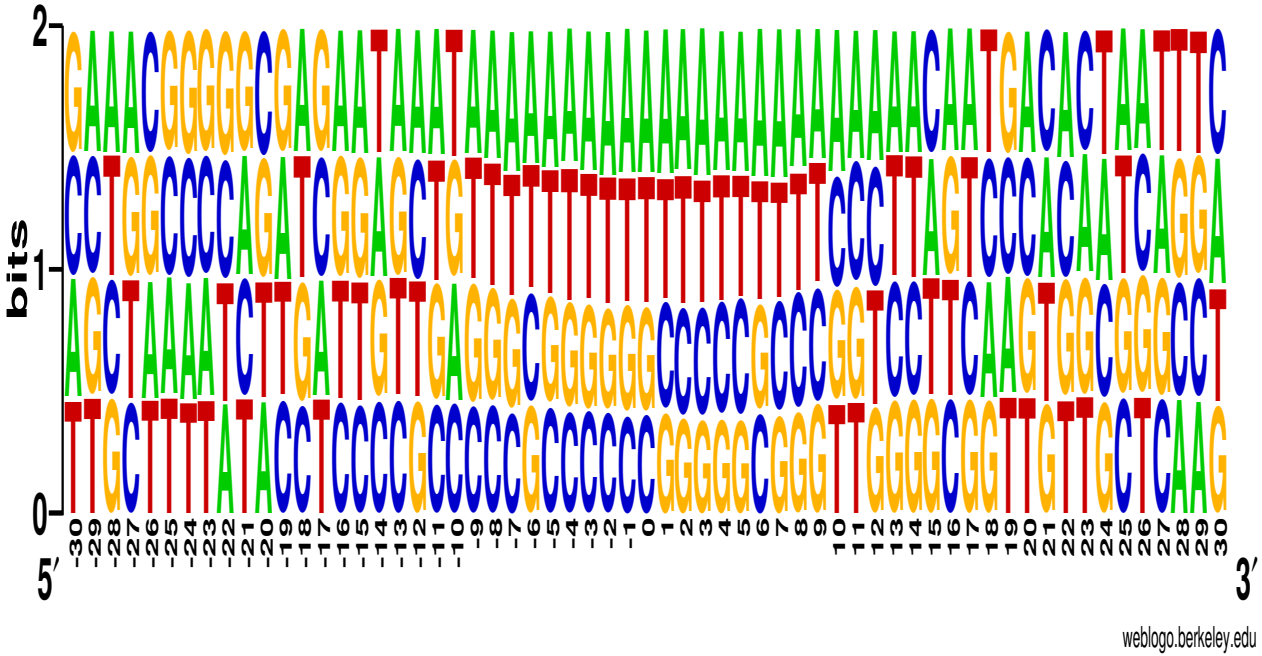
- Effizienter rekursiver Algorithmus: (statt exponentiell) Aufwand linear in der Länge der Eingabe
- **Gewann Best Student Paper Award auf der ISMB'08**

Beispiel GATTACA an variabler Position

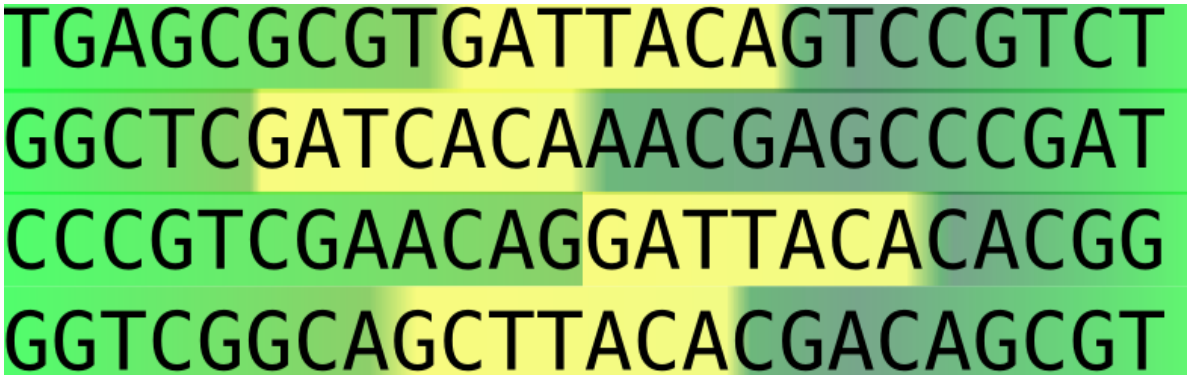


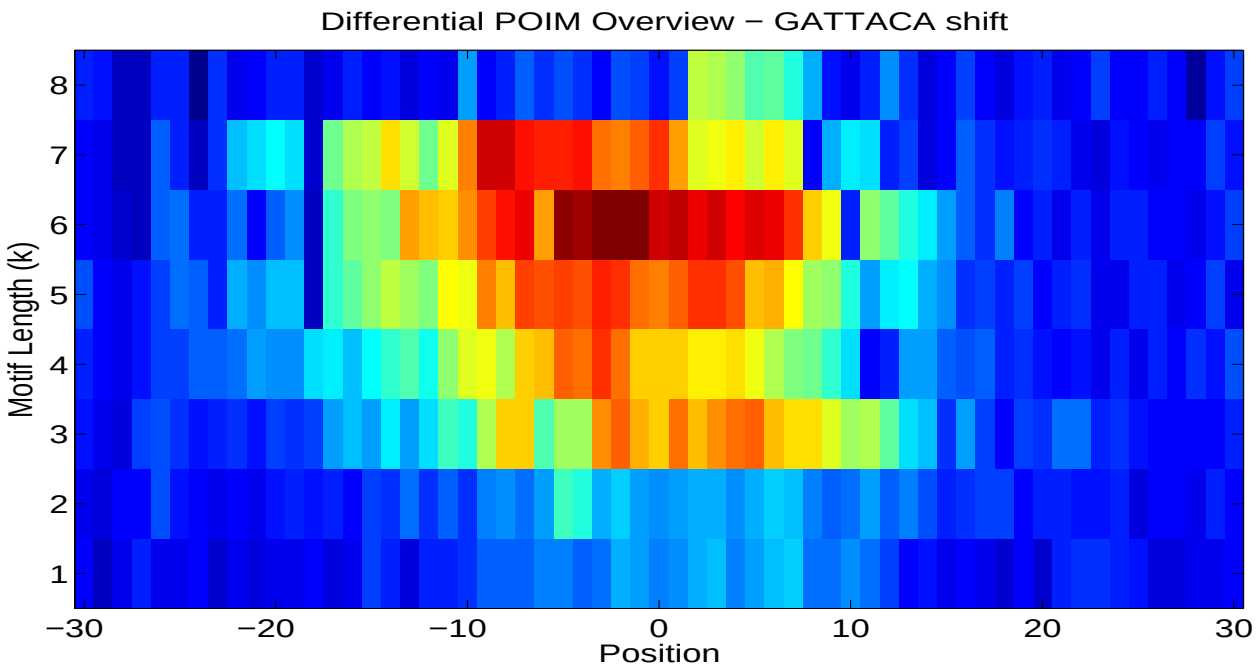
Beispiel GATTACA an variabler Position





Beispiel GATTACA an variabler Position





Beiträge der Arbeit

Maschinelles Lernen

- String-Kerne für Genom-Signale
- Large-Scale-Lernalgorithmen für String-Kerne und SVMs
- Erklärung des gelernten SVM-Klassifikators (MKL, POIMs)

Beiträge der Arbeit

Maschinelles Lernen

- String-Kerne für Genom-Signale
- Large-Scale-Lernalgorithmen für String-Kerne und SVMs
- Erklärung des gelernten SVM-Klassifikators (MKL, POIMs)

Informatik

- effiziente Datenstrukturen und Algorithmen
- Parallelisierung, effiziente Implementierung

$\Rightarrow$ **Bis zu Faktor 100.000 schneller!**

Beiträge der Arbeit

Maschinelles Lernen

- String-Kerne für Genom-Signale
- Large-Scale-Lernalgorithmen für String-Kerne und SVMs
- Erklärung des gelernten SVM-Klassifikators (MKL, POIMs)

Informatik

- effiziente Datenstrukturen und Algorithmen
- Parallelisierung, effiziente Implementierung

⇒ **Bis zu Faktor 100.000 schneller!**

Anwendungen in der Biologie

- Rekord-Erkennungsraten bei der Erkennung von Signalen
- Rekord-Erkennungsraten bei der Genstrukturvorhersage
- Identifikation von Sequenzmustern

⇒ **Halbierung der Fehlerraten!**

Danksagung

- Dr. Gunnar Rätsch und Prof. Dr. Klaus-Robert Müller
- Arbeitsgruppen Fraunhofer Institut FIRST.IDA und TU.IDA in Berlin, Friedrich-Miescher-Laboratorium in Tübingen
- Fabio De Bona, Lydia Bothe, Vojtech Franc, Sebastian Henschel, Motoaki Kawanabe, Cheng Soon Ong, Petra Philips, Konrad Rieck, Reiner Schulz, Gabriele Schweikert, Christin Schäfer, Christian Widmer und Alexander Zien
- Finanzielle Unterstützung durch *IST Programme of the European Community, under the PASCAL2 Network of Excellence* und durch die *Learning and Inference Platform* der Max-Planck- und Fraunhofer-Gesellschaften.