Machine Learning for Genomic Sequence Analysis

(Wissenschaftliche Aussprache)

Sören Sonnenburg

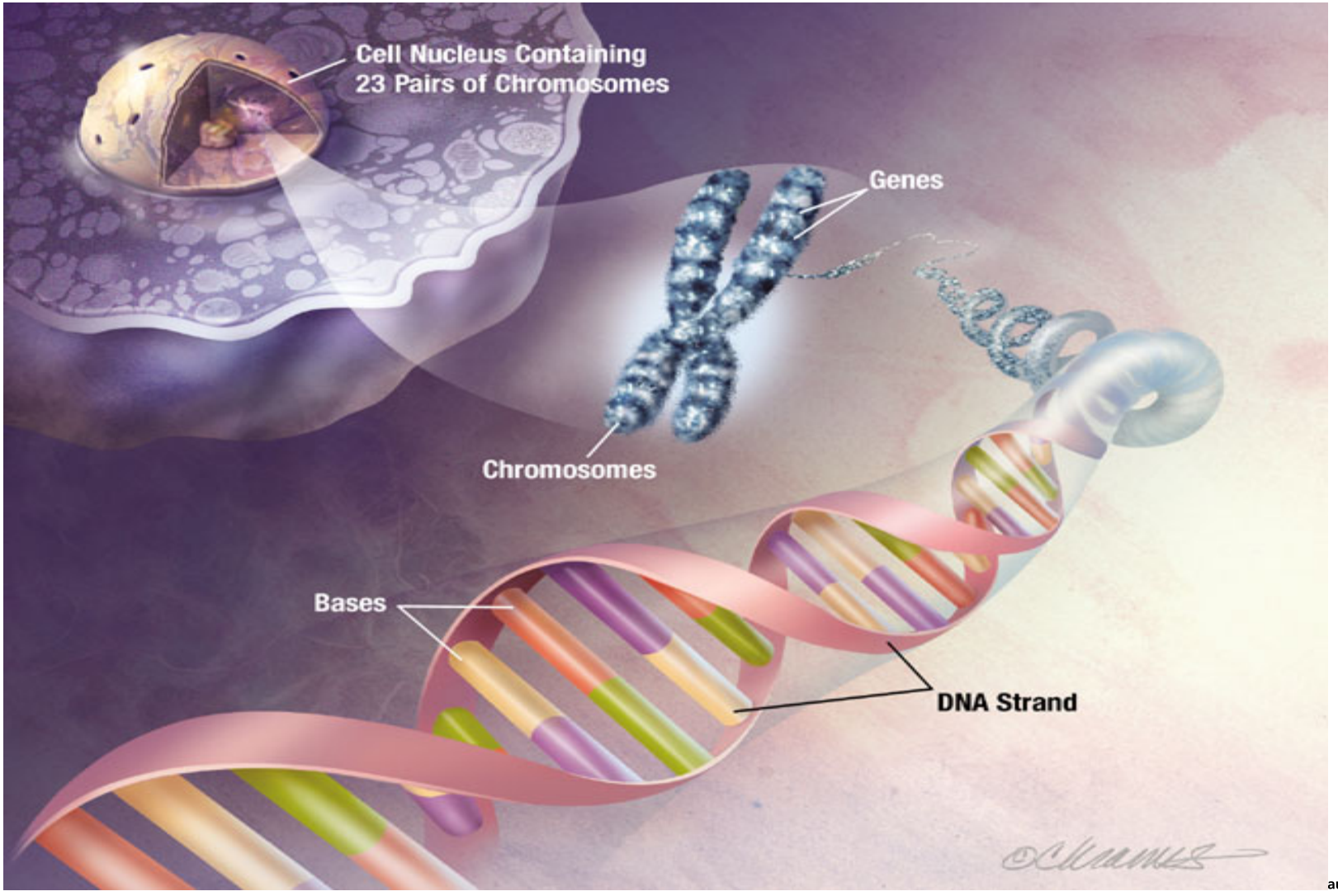
Fraunhofer Institut FIRST.IDA, Berlin

Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Inhalt

Inhalt

Genom


Cell Nucleus Containing
23 Pairs of Chromosomes
Genes
Chromosomes
Bases
DNA Strand

Sequenzanalyse

Genom: Lange Zeichenketten von `A,C,G,T`

Riesige Genom-Datenbanken

- 2006: menschliche Genom sequenziert, 3 Mrd. Basenpaare
- 2008: $>$ 250 Organismen sequenziert, $>$ 4000 Genomprojekte

Aber: Lage und Bedeutung der funktionalen Elemente unklar

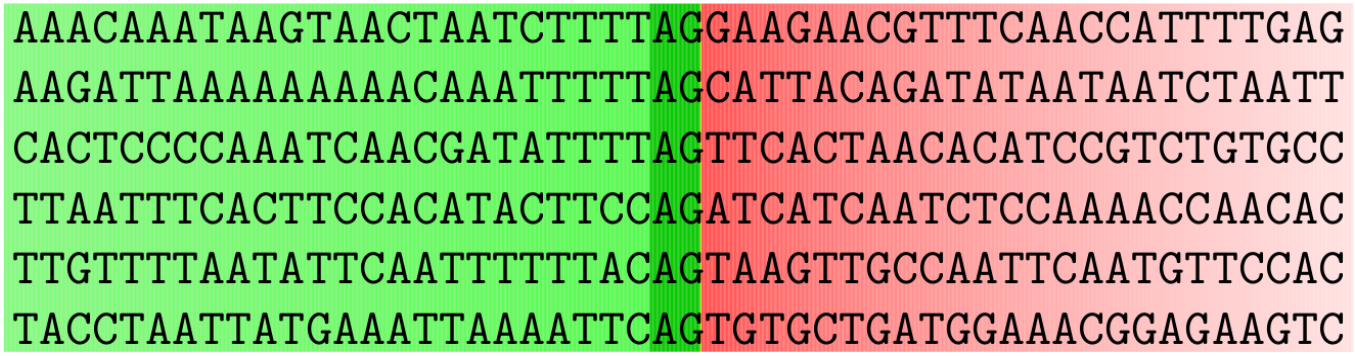
- Wo liegen proteinkodierende Gene (Anfang, Ende)?
- Welche Teile des Gens sind proteinkodierend (Exons)?

Intelligente, computergestützte Methoden notwendig!

Klassifikation - Lernen anhand von Beispielen

Gegeben:

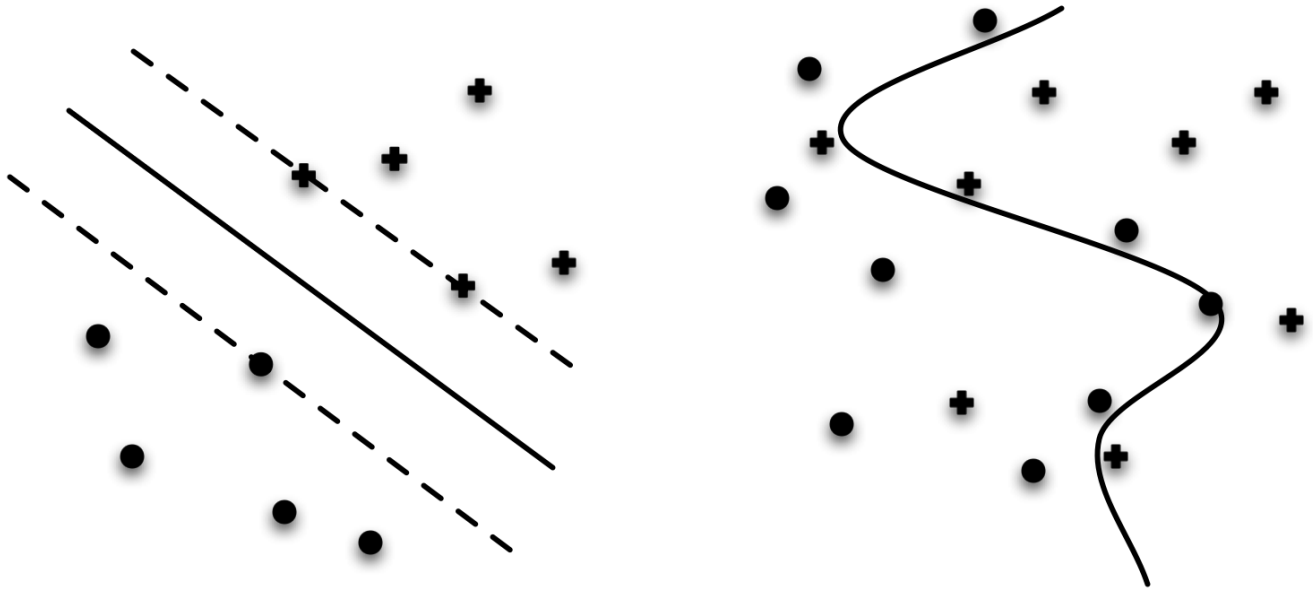
Trainingsbeispiele $(\mathbf{x}_i, y_i)_{i=1}^N \in (\{A, C, G, T\}^L, \{-1, +1\})^N$



Gesucht:

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

Support Vector Machines (SVMs)



- **Support Vector Machines** lernen Gewichte $\alpha \in \mathbb{R}^N$ auf Trainingsbeispielen im Kern-Merkmalsraum $\Phi : \mathbf{x} \mapsto \mathbb{R}^D$,

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

mit Kern $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$

Beiträge

Maschinelles Lernen

- String-Kerne für Genom-Signale
- Large-Scale-Lernalgorithmen für String-Kerne und SVMs
- Erklärung des gelernten SVM-Klassifikators

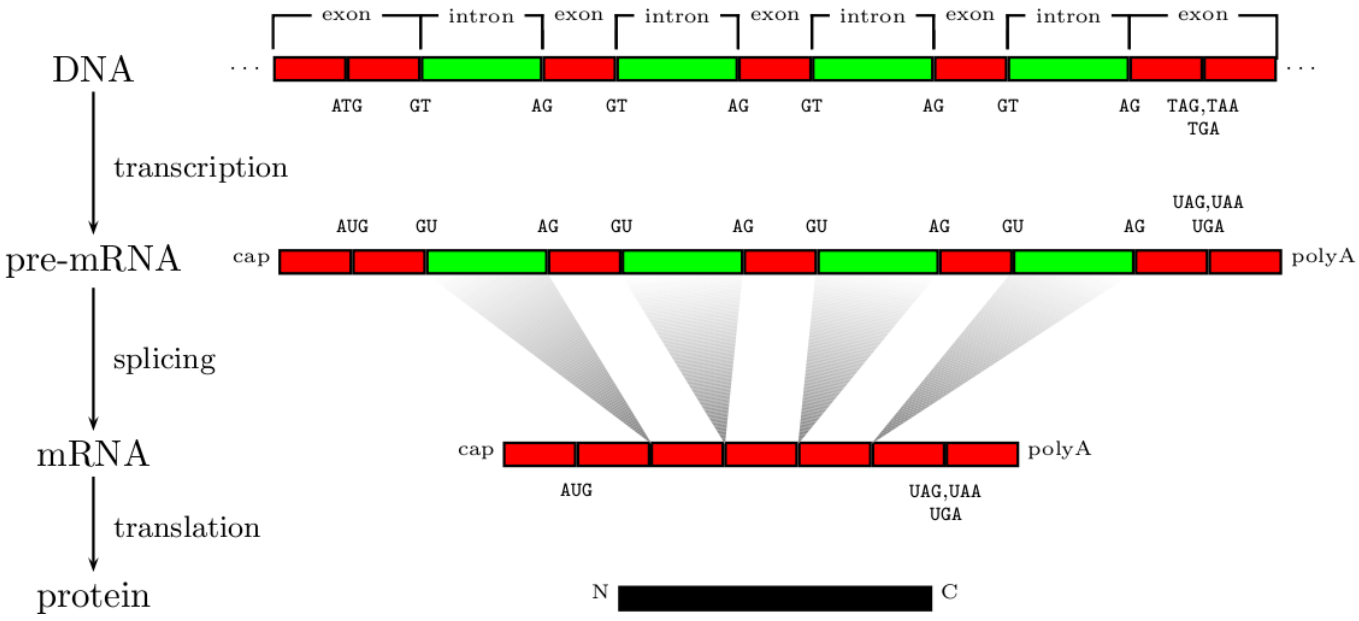
Bioinformatik Anwendungen

- Rekord-Erkennungsraten bei der Erkennung von Signalen (z.B. Transkriptions-Start-Stellen, Spleißstellen)
- Identifikation von Sequenzmustern

Inhalt

Genom-Signale

Lernen von Genom-Signalen durch Maschinelles Lernen

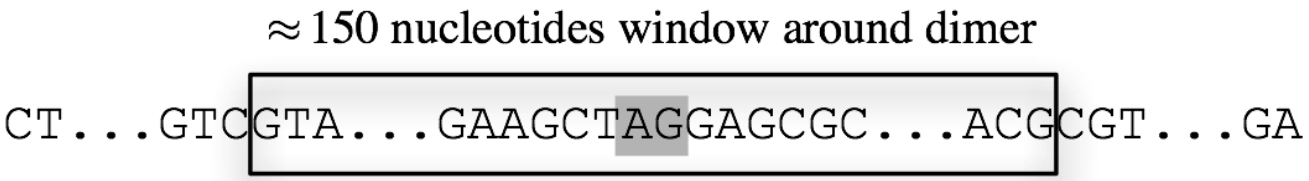


Signale

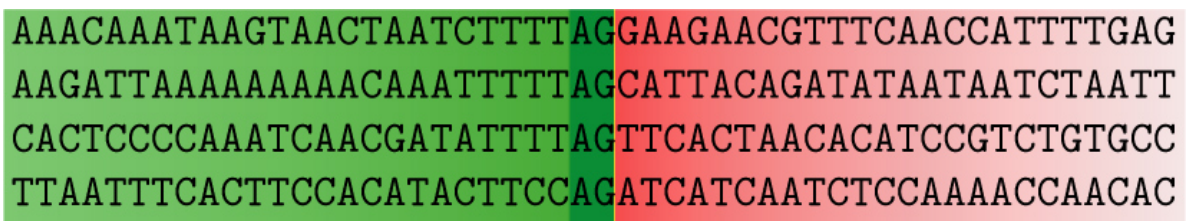
- Start/Ende von Genen
- Spleißstellen (Exon-Intron-Grenzen), etc.

Anwendungsbeispiel Spleißstellen-Erkennung

Diskriminieren von wahren Signalen gegen andere Positionen



- Wahre Stellen: festes Fenster um die wahre Spleißstelle
- Falsche Stellen: alle anderen Konsensus-Stellen

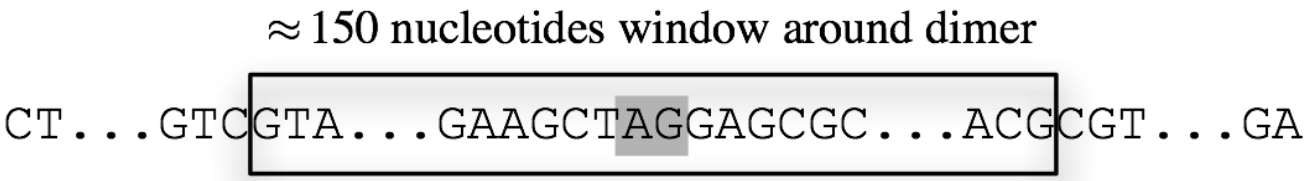


Erkennungsmethode

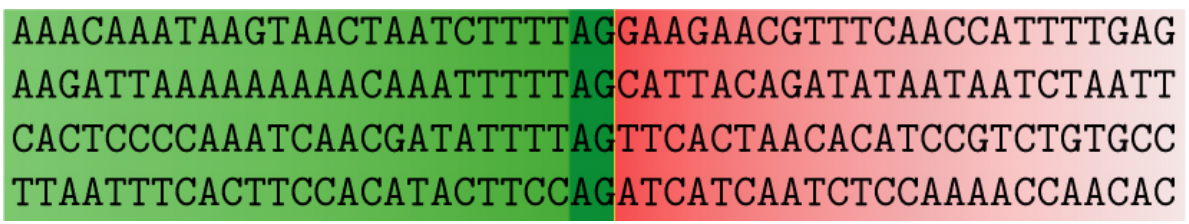
- Verwenden von Support Vector Machines
- Ähnlichkeit von Sequenzen wird mittels String-Kern gemessen

Anwendungsbeispiel Spleißstellen-Erkennung

Diskriminieren von wahren Signalen gegen andere Positionen



- Wahre Stellen: festes Fenster um die wahre Spleißstelle
- Falsche Stellen: alle anderen Konsensus-Stellen



Erkennungsmethode

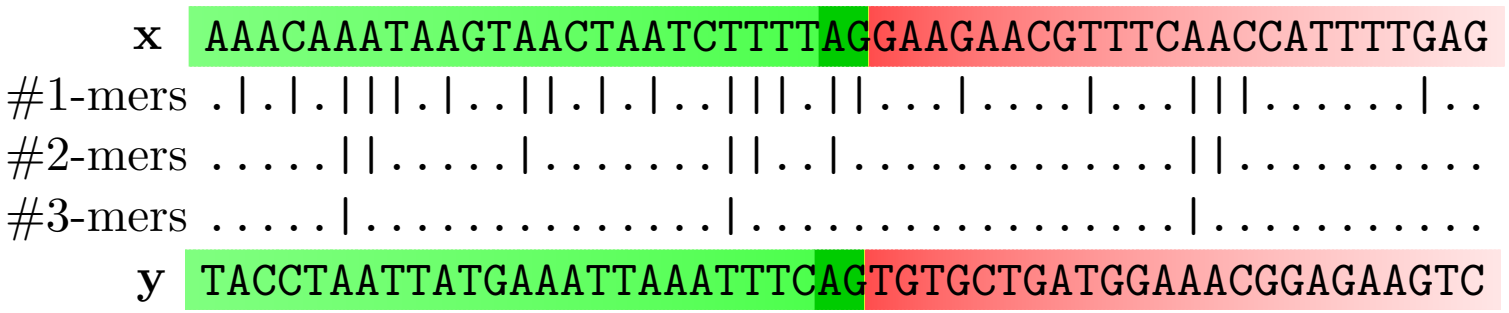
- Verwenden von Support Vector Machines
- Ähnlichkeit von Sequenzen wird mittels String-Kern gemessen

Der Weighted Degree Kern

Support Vector Machine

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

$$\mathsf{k}(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{K} \beta_k \sum_{i=1}^{L-k+1} \mathbb{I}\left\{\mathbf{x}[i]^k = \mathbf{x}'[i]^k\right\}.$$



Beispiel: $K = 3 : \mathsf{k}(\mathbf{x}, \mathbf{x}') = \beta_1 \cdot 21 + \beta_2 \cdot 8 + \beta_3 \cdot 3$

Verschnellerung von String-Kern-SVMs

1. Lineare Laufzeit des Kerns
2. Beschleunigung von Linearkombinationen von Kernen

Idee des `Linadd` Algorithmus:

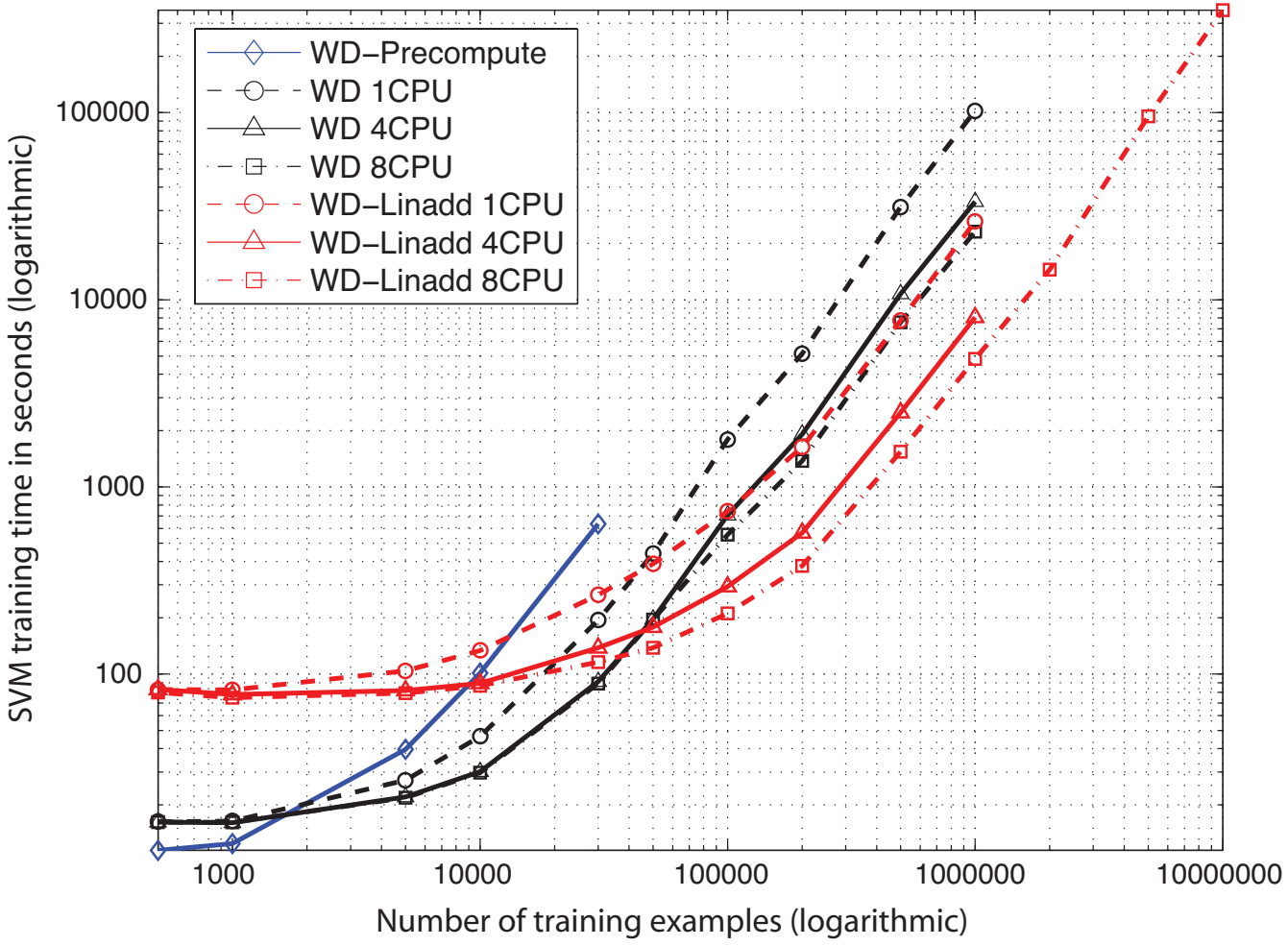
Effizientes Speichern von $\mathbf{w}$ und Berechnen von $\mathbf{w} \cdot \Phi(\mathbf{x})$

$$f(\mathbf{x}_j) = \sum_{i=1}^{N} \alpha_i y_i \, \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \underbrace{\sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i)}_{\mathbf{w}} \cdot \Phi(\mathbf{x}_j) = \mathbf{w} \cdot \Phi(\mathbf{x}_j)$$

Möglich bei niedrigdimensionalem oder sehr spärlichem $\mathbf{w}$

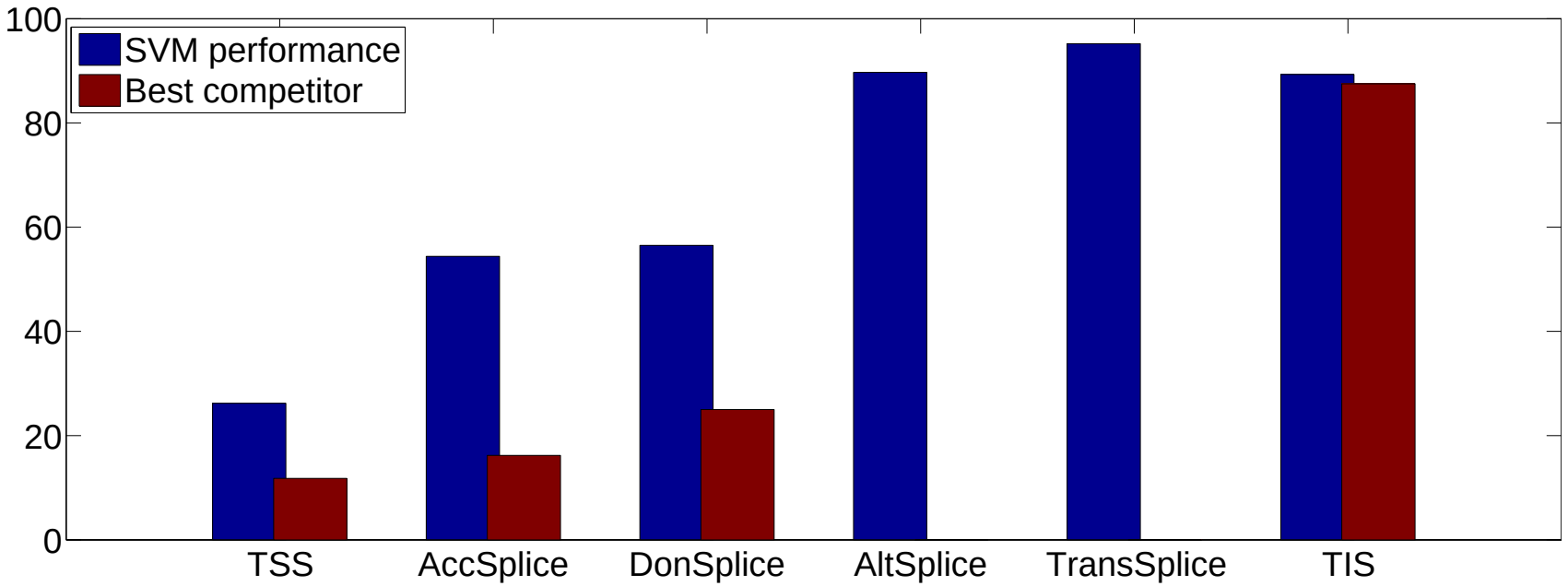
Aufwand: $\mathcal{O}(ML) \Rightarrow$ **Beschleunigung um Faktor** N

Ergebnisse



- Beschleunigung des Trainings: In gleicher Zeit Training auf mehr als doppelt so vielen Daten möglich

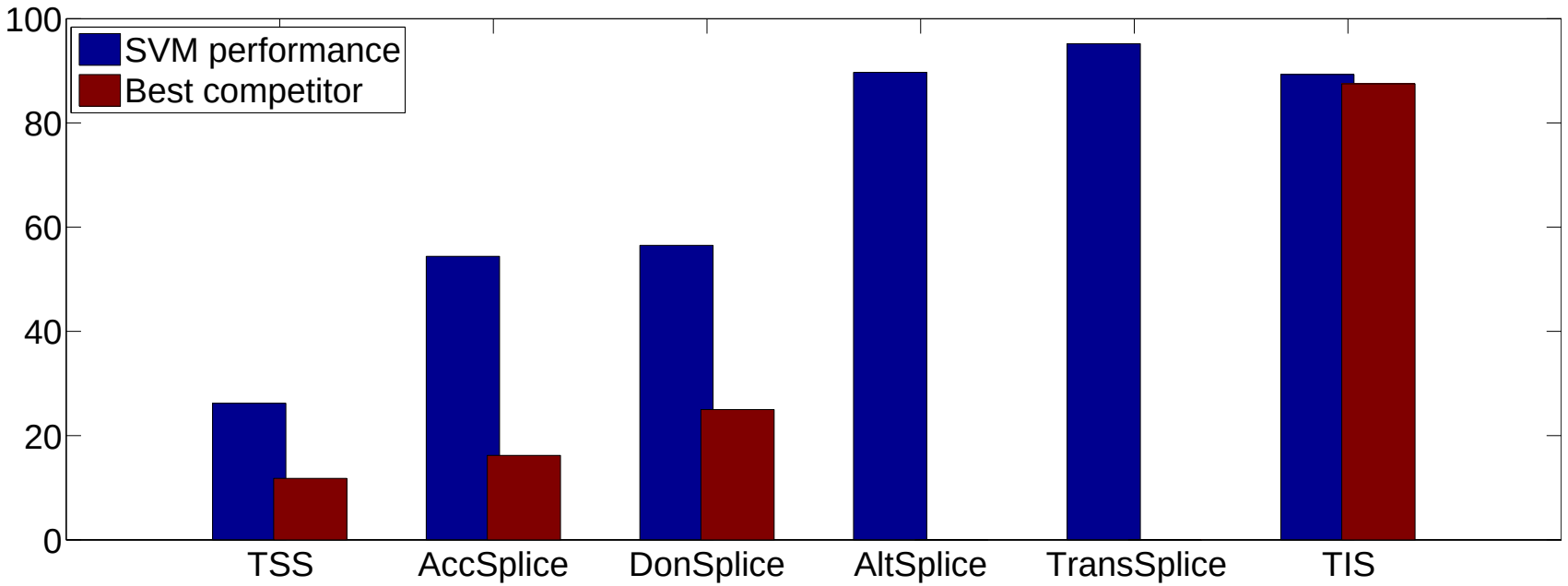
Ein vielseitiger Ansatz



- Transkriptionsstart (Sonnenburg et al., Eponine Down et al.)
- Akzeptor- und Donor-Spleißstelle (Sonnenburg et al.)
- Alternative Splicing (Rätsch et al.), Transsplicing (Schweikert et al.)
- Translation-Initiation (Sonnenburg et al., Saeys et al.)

Nachteil von SVMs: Lösung schwer zu verstehen

Ein vielseitiger Ansatz



- Transkriptionsstart (Sonnenburg et al., Eponine Down et al.)
- Akzeptor- und Donor-Spleißstelle (Sonnenburg et al.)
- Alternative Splicing (Rätsch et al.), Transsplicing (Schweikert et al.)
- Translation-Initiation (Sonnenburg et al., Saeys et al.)

Nachteil von SVMs: Lösung schwer zu verstehen

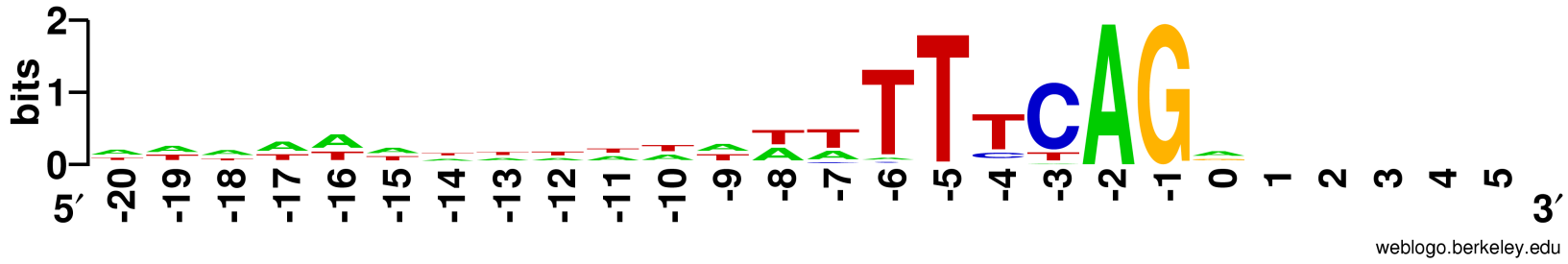
Inhalt

Support Vector Machines verstehen

Für Positional Weight Matrices (PWMs)

k-mer	pos. 1	pos. 2	pos. 3	pos. 4	···
A	0.1	0.3	0.2	0.2	···
C	0.0	0.0	0.4	0.2	···
G	0.1	0.7	0.3	0.5	···
T	0.8	0.0	0.1	0.1	···

gibt es sequence logos:



Ziel:

Ähnliche Mittel, um Support Vector Machines zu verstehen.

Wieso sind SVMs schwer zu verstehen?

SVM Entscheidungsfunktion ist Gewichtung α von Trainingsbeispielen.

$$s(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \, \mathrm{k}(\mathbf{x}_i, \mathbf{x}) + b$$

$\alpha_1 \cdot$ AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
$\alpha_2 \cdot$ AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
$\alpha_3 \cdot$ CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
$\vdots$ $\vdots$
$\alpha_N \cdot$ TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC

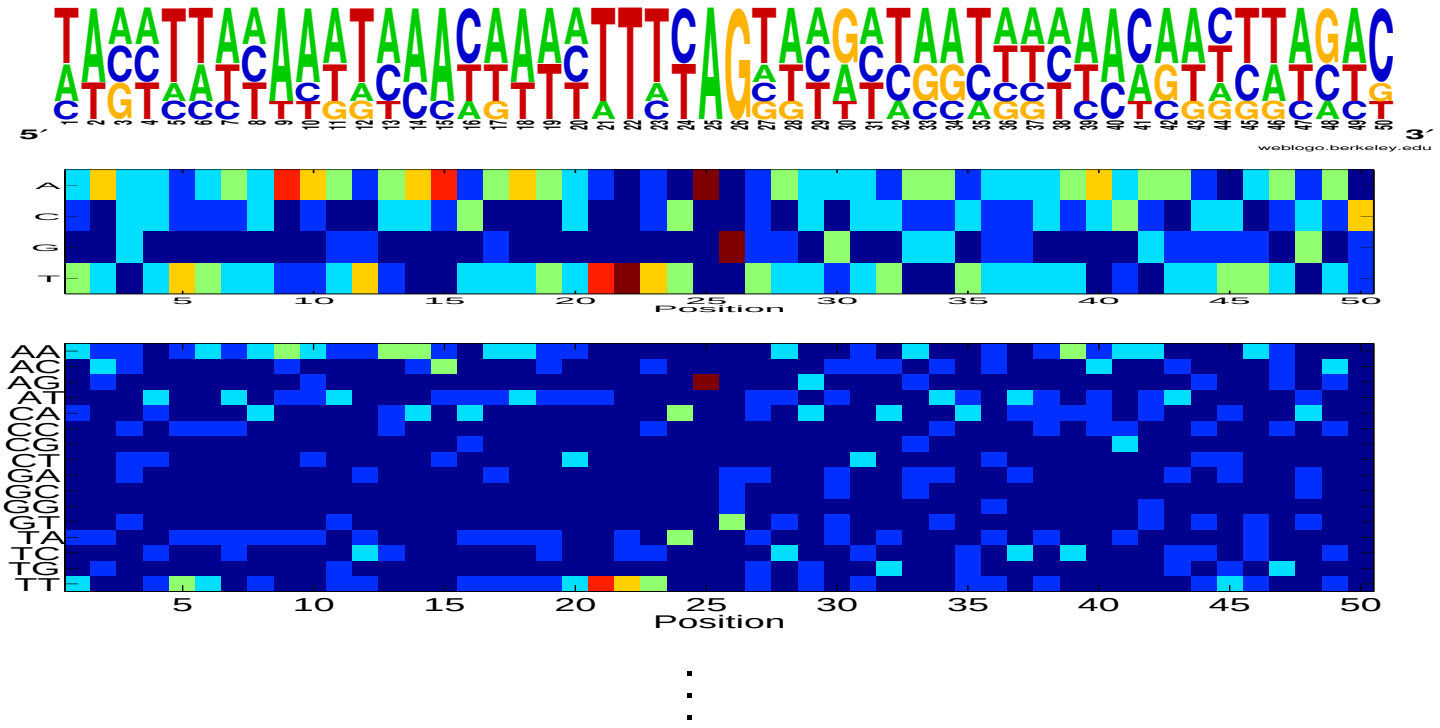
Aber wir sind interessiert an einer Gewichtung der Merkmale.

SVM Bewertungsfunktion

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i) \qquad s(\mathbf{x}) := \sum_{k=1}^{K} \sum_{i=1}^{L-k+1} w\left(\mathbf{x}[i]^k, i\right) + b$$

k-mer	pos. 1	pos. 2	pos. 3	pos. 4	$\cdots$
A	+0.1	-0.3	-0.2	+0.2	$\cdots$
C	0.0	-0.1	+2.4	-0.2	$\cdots$
G	+0.1	-0.7	0.0	-0.5	$\cdots$
T	-0.2	-0.2	0.1	+0.5	$\cdots$
AA	+0.1	-0.3	+0.1	0.0	$\cdots$
AC	+0.2	0.0	-0.2	+0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
TT	0.0	-0.1	+1.7	-0.2	$\cdots$
AAA	+0.1	0.0	0.0	+0.1	$\cdots$
AAC	0.0	-0.1	+1.2	-0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
TTT	+0.2	-0.7	0.0	0.0	$\cdots$

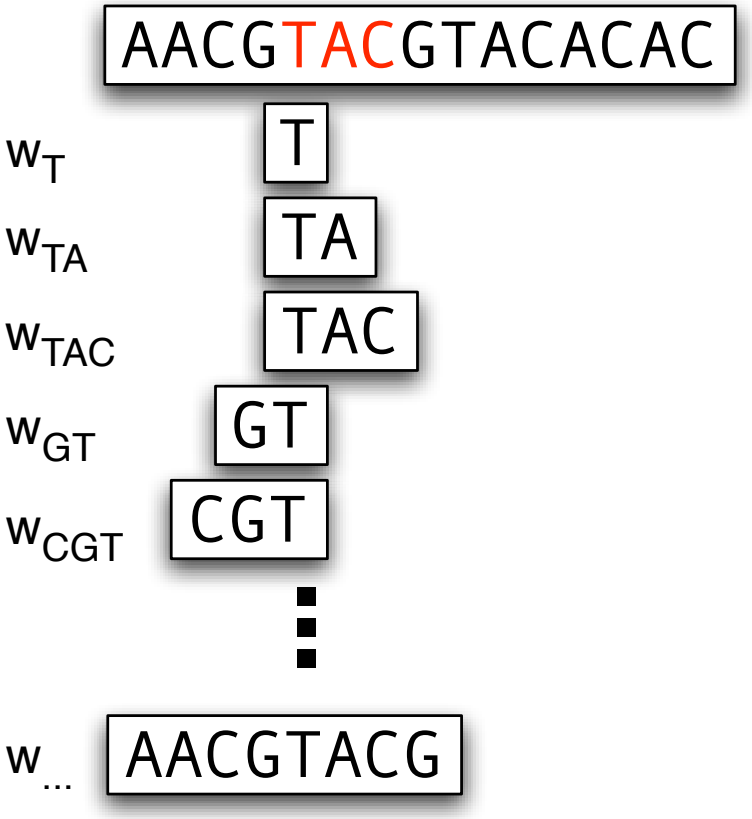
Der SVM Gewichtsvektor **w**



- Explizite Repräsentation von **w** erlaubt (teilweise) Interpretation!
- String-Kern-SVMs sind effizient in der Handhabung langer k-mere ($k > 10$)

Aber: Gewichte der Sub-Strings sind nicht unabhängig!

Gegenseitige Abhängigkeit von $k-$mer Gewichten



Wie wichtig ist TAC?

- Wert w_{TAC} nehmen?
- Aber Sub-Strings und überlappende Strings tragen auch zu der Bewertung bei!

Problem

Das SVM-w spiegelt NICHT die Wichtigkeit eines Motivs wider.

Positional Oligomer Importance Matrices (POIMs)

Idee:

- Gegeben $k-$mer **z** an Position j in der Sequenz, berechne erwartete Score $\mathbb{E}[s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z}]$ (**für kleine k**)

```
AAAAAAAAAATACAAAAAAAAAA
AAAAAAAAAATACAAAAAAAAAC
AAAAAAAAAATACAAAAAAAAAG
          ⋮
TTTTTTTTTTTACTTTTTTTTTT
```

- Vergleichen mit *erwarteter Score* über **alle Sequenzen**

POIMs

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

$\Rightarrow$ **Benötigt effizienten Algorithmus zur Berechnung**

Effiziente Berechnung

Aufwand des naiven Ansatzes exponentiell $\mathcal{O}(|\Sigma|^L + L|\Sigma|^k)$
(z.B. Spleißstellen 10^{120})

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

- Anzahl der k-mere wächst linear mit der Eingabe
- Nur Merkmale abhängig von $(\mathbf{z}, j)$ sind relevant

Hauptbeitrag

Effizienter rekursiver Algorithmus:
Aufwand linear in der Länge der Eingabe: $\mathcal{O}(LN + L|\Sigma|^k)$

Idee des Algorithmus

Zerlegung der Berechnung in 4 Teile

$$\begin{aligned} Q(\mathbf{z}, j) &:= \mathbb{E}\left[s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z} \right] - \mathbb{E}\left[s(\mathbf{x}) \right]. \\ &= u(\mathbf{z}, j) + \sum_{\mathbf{y} \in \Sigma^{|\mathbf{z}|}} u(\mathbf{z}, j) \end{aligned}$$

$$\begin{aligned} u(\mathbf{z}, j) &:= \sum_{(\mathbf{y}, i) \in \mathcal{I}(\mathbf{z}, j)} Pr\left(\mathbf{x}[i] = \mathbf{y} \mid \mathbf{x}[j] = \mathbf{z} \right) w(\mathbf{y}, i) \\ &= u^{\vee}(\mathbf{z}, j) + u^{\wedge}(\mathbf{z}, j) + u^{<}(\mathbf{z}, j) + u^{>}(\mathbf{z}, j) - w(\mathbf{z}, j) \ , \end{aligned}$$

Für `AATACGTAC`: Substring, Superstring, Links- und Rechts-Überlappung

Effizienter rekursiver Algorithmus

$$
\begin{aligned}
u^{\vee}(\mathbf{z},j) &= w(\mathbf{z},j) + u^{\vee}(\tau\mathbf{z}',j) + u^{\vee}(\mathbf{z}'\tau',j+1) - u^{\vee}(\mathbf{z}',j+1) \\
u^{\wedge}(\mathbf{z},j) &= w(\mathbf{z},j) - \sum_{(\sigma,\sigma')\in\Sigma^2} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma'\right) Pr\left(\mathbf{x}[j-1]=\sigma\right) u^{\wedge}(\sigma\mathbf{z}\sigma',j-1) \\
&+ \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j-1]=\sigma\right) u^{\wedge}(\sigma\mathbf{z},j-1) + \sum_{\sigma'\in\Sigma} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma'\right) u^{\wedge}(\mathbf{z}\sigma',j) \\
u^{<}(\mathbf{z},j) &= \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j-1]=\sigma\right) \sum_{k=1}^{|\mathbf{z}|-1} L(\sigma\mathbf{z}[1]^k,j-1) \\
u^{>}(\mathbf{z},j) &= \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma\right) \sum_{k=1}^{|\mathbf{z}|-1} R(\mathbf{z}[|\mathbf{z}|-k+1]^k\sigma,j+|\mathbf{z}|-k) \ ,
\end{aligned}
$$

wobei

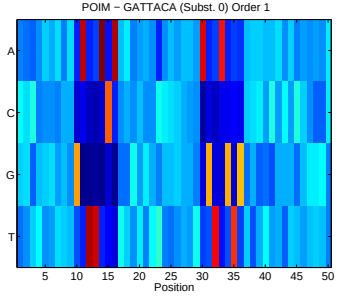
$$
\begin{aligned}
L(\mathbf{t},j) &:= \sum_{(\mathbf{y},i)\in\mathcal{L}(\mathbf{t},j)} Pr\left(\mathbf{x}[i]=\mathbf{y} \mid \mathbf{x}[j]=\mathbf{t}\right) w(\mathbf{y},i) \\
R(\mathbf{t},j) &:= \sum_{(\mathbf{y},i)\in\mathcal{R}(\mathbf{t},j)} Pr\left(\mathbf{x}[i]=\mathbf{y} \mid \mathbf{x}[j]=\mathbf{t}\right) w(\mathbf{y},i) \ .
\end{aligned}
$$

Rangliste von Merkmalen und kompakte Darstellung

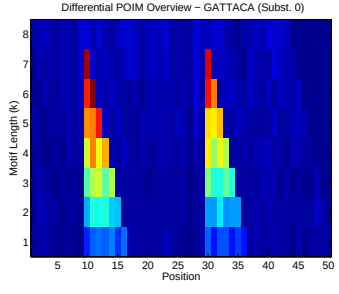
- Höchstbewertete **z** von $|Q(\mathbf{z}, i)|$ sind Enhancer oder Silencer

z	i	$\lvert Q(\mathbf{z}, i)\rvert$
GATTACA	10	+30
AGTAGTG	30	+20
AAAAAAA	10	-10
...	...	...

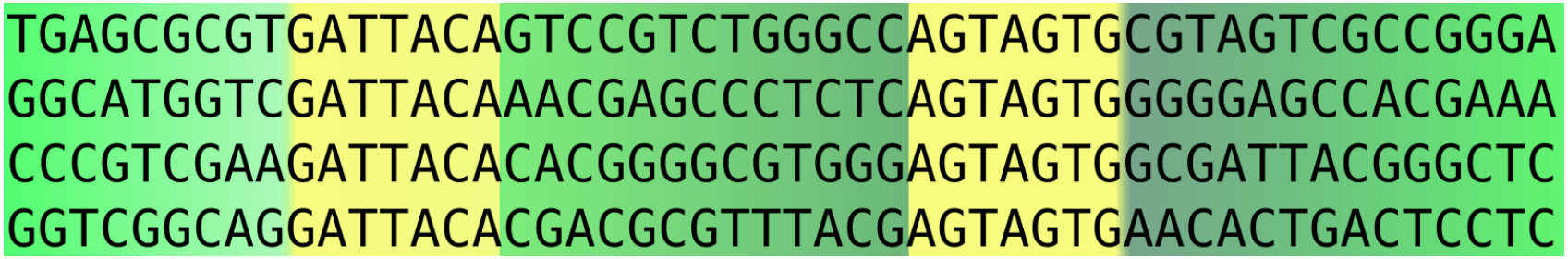
- Visualisierung von POIMs als Wärmebild:
  x-Achse: Position
  y-Achse: k-mer
  Farbe: Wichtigkeit



- Für große k: Differential POIMs;
  x-Achse: Position
  y-Achse: k-mer Länge
  Farbe: Wichtigkeit



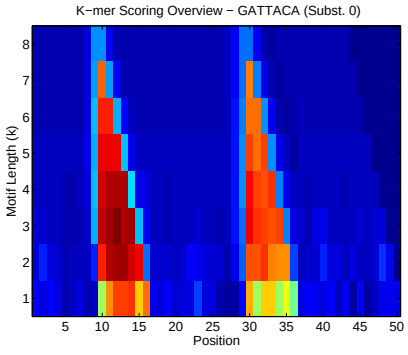
GATTACA und AGTAGTG an festen Positionen 10 and 30



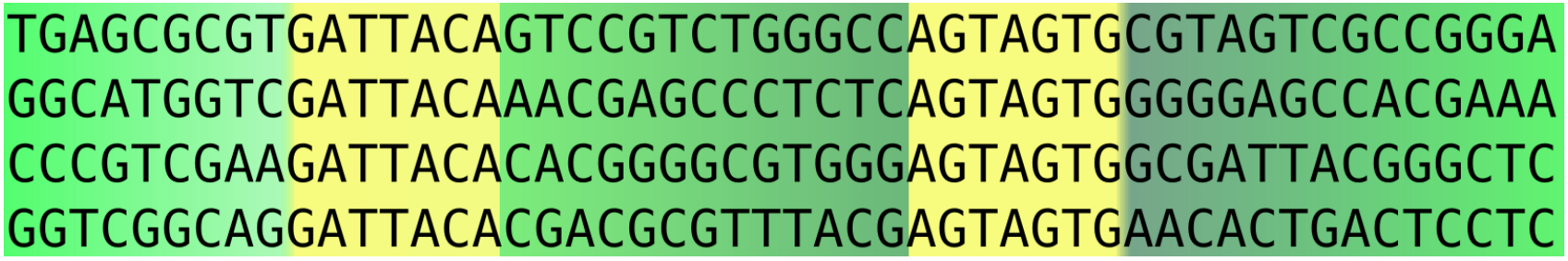
GATTACA und AGTAGTG an festen Positionen 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

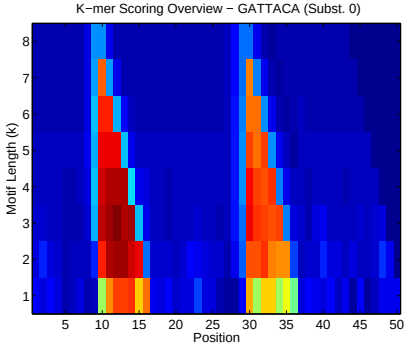
w



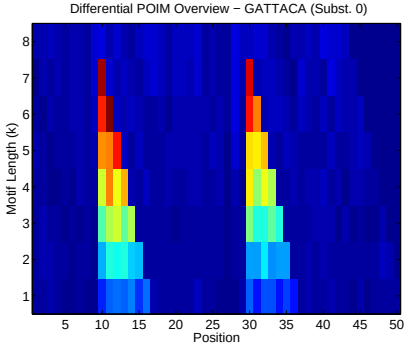
GATTACA und AGTAGTG an festen Positionen 10 and 30



w



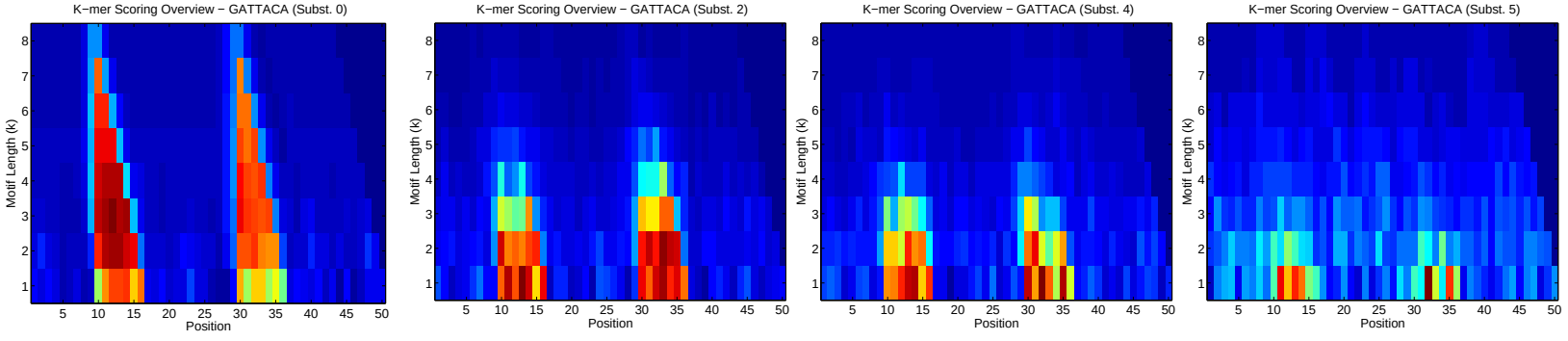
Q



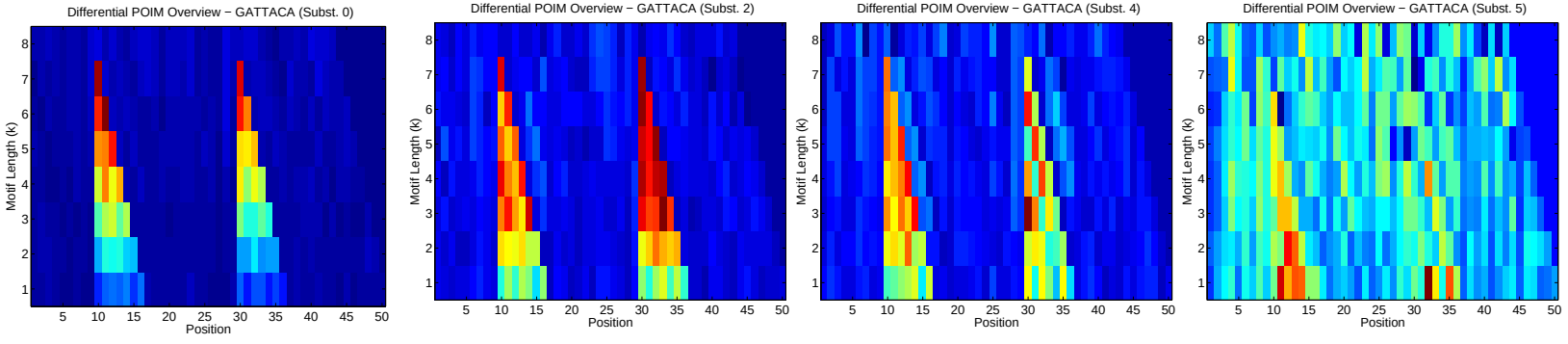
GATTACA und AGTAGTG an festen Positionen 10 und 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

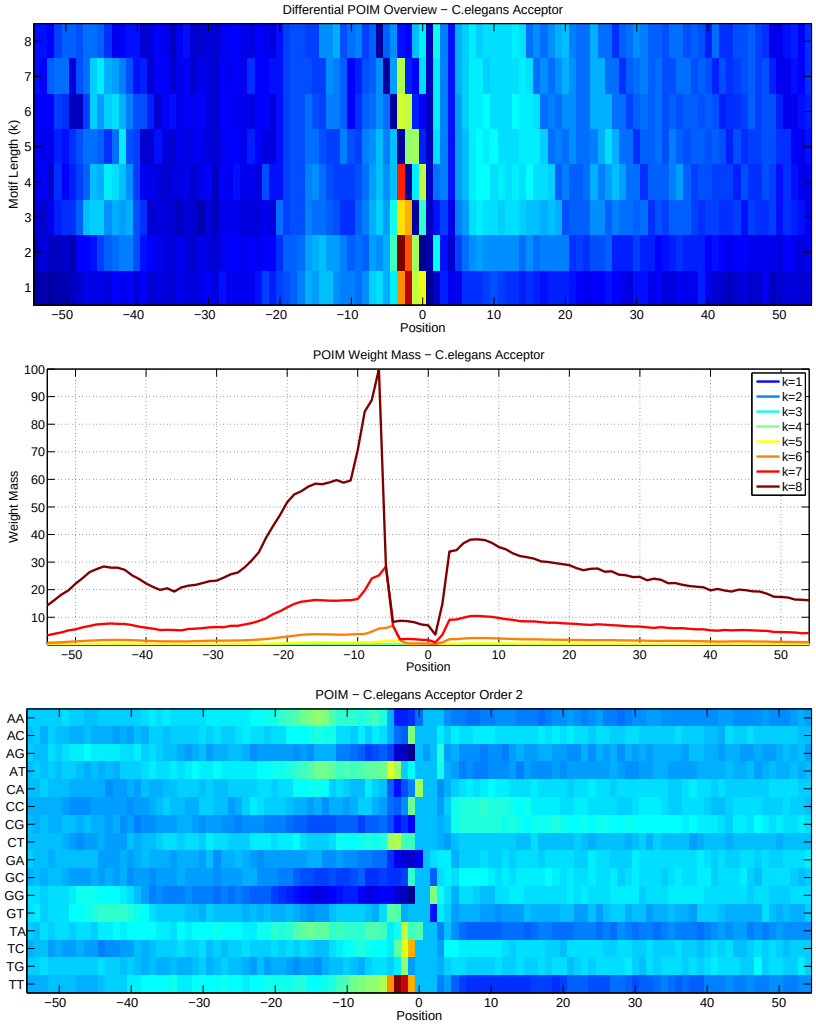
w



Q



C.elegans-Akzeptor-Spleißstellen-Erkennung



- Upstream
  - AG**GT**AAGT -44/++ Donor
  - GGGGGG -16/- - Silencer?
  - **TAAT**AA -16/++ Branch
- Central
  - TTTTTTC -06/+
  - TTTC**AG**$\frac{A}{G}$ -03/++ Akzeptor
- Downstream
  - TTTTTTTT +07/- -
  - TTTTT +26/- -

Inhalt

Beiträge

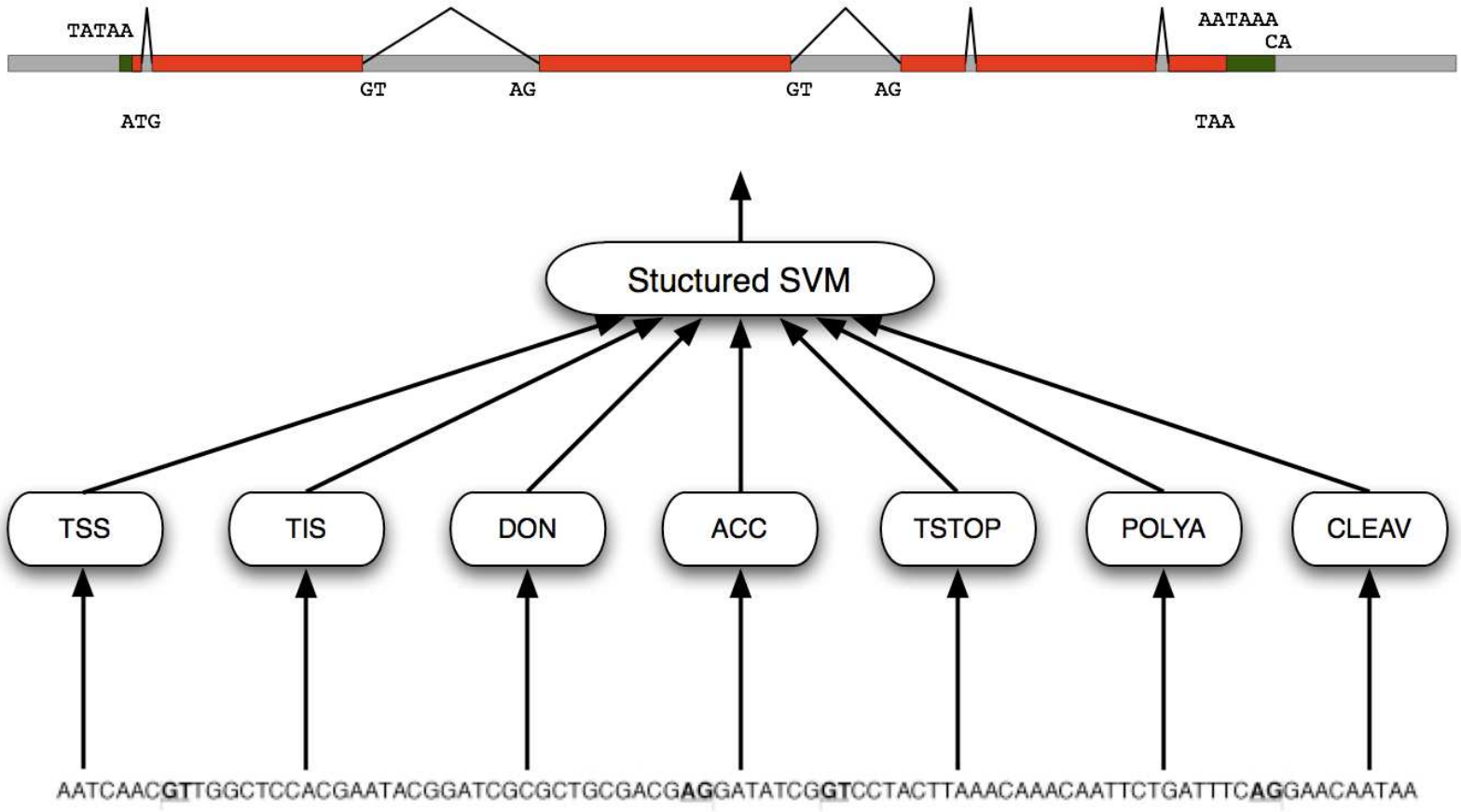
Maschinelles Lernen

- String-Kerne für Genom-Signale
  - Weighted Degree Kern (mit Shift)
- Large-Scale-Lernalgorithmen für String-Kerne und SVMs
  - Beschleunigt Training und Auswertung
- Erklärung des gelernten SVM-Klassifikators
  - POIMs
  - Multiple Kernel Learning

Bioinformatik Anwendungen

- Rekord-Erkennungsraten bei der Erkennung von Signalen
  - Transkriptions-Start
  - Spleißstellen
- Ausblick: Gen-Suche

Ausblick: Gen-Suche



Gensuchmaschine mGene (Schweikert et al. 2009)

Danksagung

- Dr. Gunnar Rätsch und Prof. Dr. Klaus-Robert Müller
- Arbeitsgruppen Fraunhofer Institute FIRST.IDA und TU.IDA in Berlin, Friedrich Miescher Labor 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 PASCAL Network of Excellence (IST-2002-506778)* und durch die *Learning and Inference Platform* der Max-Planck- und Fraunhofer-Gesellschaft.

Publikationen I


A. Ben-Hur, C. S. Ong, **S. Sonnenburg**, B. Schölkopf, and G. Rätsch.
Support vector machines and kernels for computational biology.
PLoS Comput Biology, 4(10):e1000173, Oct 2008.


V. Franc and **S. Sonnenburg**.
OCAS optimized cutting plane algorithm for support vector machines.
In *Proceedings of the 25nd International Machine Learning Conference*, pages 320–327. ACM Press, 2008.


K.-R. Müller, G. Rätsch, **S. Sonnenburg**, S. Mika, M. Grimm, and N. Heinrich.
Classifying 'drug-likeness' with kernel-based learning methods.
J. Chem. Inf. Model, 45:249-253, 2005


G. Rätsch and **S. Sonnenburg**.
Accurate Splice Site Prediction for Caenorhabditis Elegans, pages 277–298.
MIT Press series on Computational Molecular Biology. MIT Press, 2004.


G. Rätsch and **S. Sonnenburg**.
Large scale hidden semi-markov svms.
In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1161–1168. MIT Press, Cambridge, MA, 2007.


G. Rätsch, **S. Sonnenburg**, and B. Schölkopf.
RASE: Recognition of alternatively spliced exons in C. elegans.
Bioinformatics, 21:i369–i377, 2005.


G. Rätsch, **S. Sonnenburg**, and C. Schäfer.
Learning interpretable svms for biological sequence classification.
BMC Bioinformatics, 7((Suppl 1)):S9, Mar. 2006.

Publikationen II

G. Rätsch, **S. Sonnenburg**, J. Srinivasan, H. Witte, K.-R. Müller, R. Sommer, and B. Schölkopf.
Improving the c. elegans genome annotation using machine learning.
PLoS Computational Biology, 3(2):e20, 2007.

K. Rieck, P. Laskov, and **S. Sonnenburg**.
Computation of similarity measures for sequential data using generalized suffix trees.
In *Advances in Neural Information Processing Systems 19*, pages 1177–1184, Cambridge, MA, 2007. MIT Press.

G. Schweikert, G. Zeller, A. Zien, J. Behr, C.-S. Ong, P. Philips, A. Bohlen, **S. Sonnenburg**, and G. Rätsch.
mGene: A novel discriminative gene finding system.
In preparation, 2009.

S. Sonnenburg.
New methods for splice site recognition.
Master's thesis, Humboldt University, 2002.
supervised by K.-R. Müller H.-D. Burkhard and G. Rätsch.

S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller.
New methods for splice-site recognition.
In *Proceedings of the International Conference on Artifical Neural Networks.*, pages 329–336, 2002.
Copyright by Springer.

S. Sonnenburg, G. Rätsch, and C. Schäfer.
Learning interpretable SVMs for biological sequence classification.
In S. Miyano, J. P. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. Waterman, editors, *Research in Computational Molecular Biology, 9th Annual International Conference, RECOMB 2005*, volume 3500, pages 389–407. Springer-Verlag Berlin Heidelberg, 2005a.

Publikationen III

S. Sonnenburg, G. Rätsch, and B. Schölkopf.
Large scale genomic sequence SVM classifiers.
In L. D. Raedt and S. Wrobel, editors, *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 849–856, New York, NY, USA, 2005b. ACM Press.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf.
Large Scale Multiple Kernel Learning.
Journal of Machine Learning Research, 7:1531–1565, July 2006a.

S. Sonnenburg, G. Rätsch and C. Schäfer.
A General and Efficient Multiple Kernel Learning Algorithm.
Advances in Neural Information Processing Systems 18, pages 1273–1280, Cambridge, MA 2006, MIT Press.

S. Sonnenburg, A. Zien, and G. Rätsch.
ARTS: Accurate Recognition of Transcription Starts in Human.
Bioinformatics, 22(14):e472–480, 2006b.

S. Sonnenburg, G. Rätsch, and K. Rieck.
Large scale learning with string kernels.
In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 73–103. MIT Press, 2007a.

S. Sonnenburg, M. Braun, C.S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K.-R. Müller, F. Pereira, C. E. Rasmussen, G. Rätsch, B. Schölkopf, A. Smola, P. Vincent, J. Weston, and R. Williamson.
The Need for Open Source Software in Machine Learning.
Journal of Machine Learning Research, 8:2443-2466, September 2007.

Publikationen IV

S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch.

Accurate Splice Site Prediction.

BMC Bioinformatics, Special Issue from NIPS workshop on New Problems and Methods in Computational Biology Whistler, Canada, 18 December 2006, 8:(Suppl. 10):S7, December 2007b.

S. Sonnenburg, A. Zien, P. Philips, and G. Rätsch.

POIMs: positional oligomer importance matrices — understanding support vector machine based signal detectors.

Bioinformatics, 2008.

(received the Best Student Paper Award at ISMB08).

K. Tsuda, M. Kawanabe, G. Rätsch, **S. Sonnenburg**, and K. Müller.

A new discriminative kernel from probabilistic models.

In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural information processings systems*, volume 14, pages 977–984, Cambridge, MA, 2002a. MIT Press.

K. Tsuda, M. Kawanabe, G. Rätsch, **S. Sonnenburg**, and K. Müller.

A new discriminative kernel from probabilistic models.

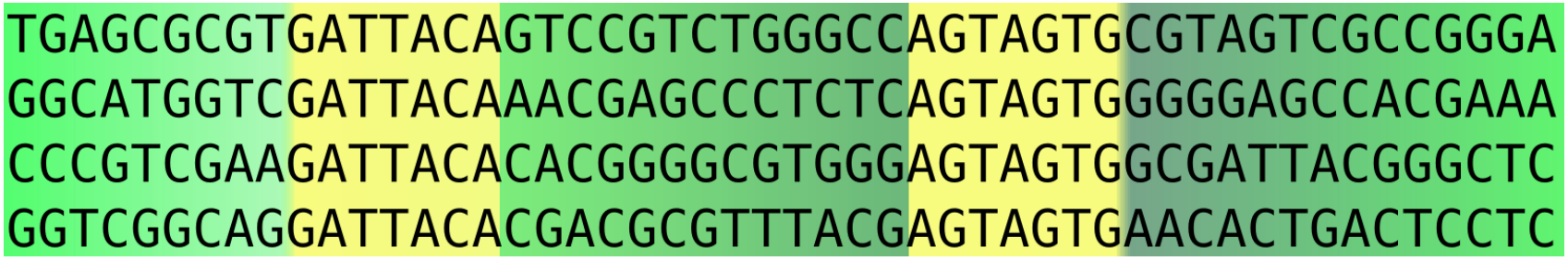
Neural Computation, 14:2397–2414, 2002b.

A. Zien, P. Philips, and **S. Sonnenburg**.

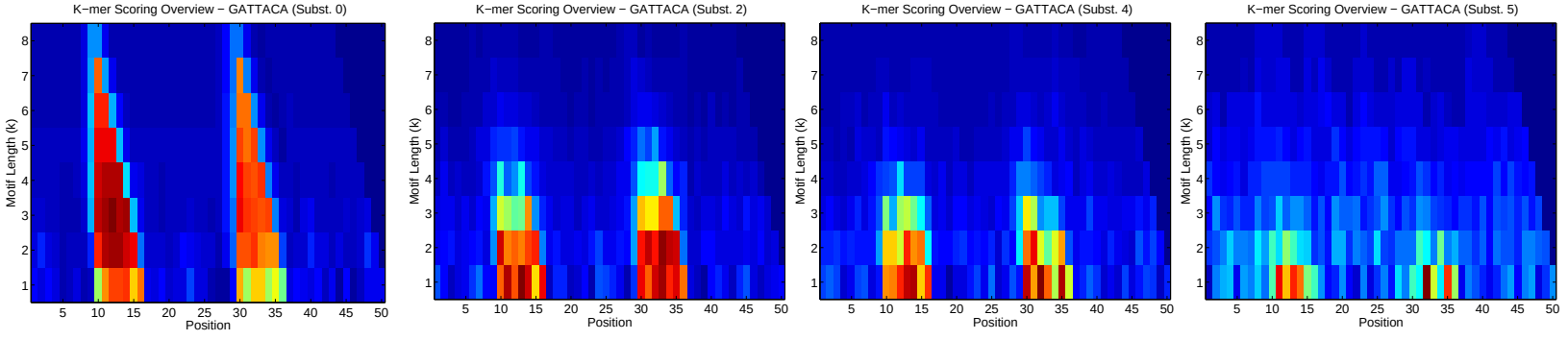
Computing Positional Oligomer Importance Matrices (POIMs).

Research Report; Electronic Publication 2, Fraunhofer FIRST, Dec. 2007.

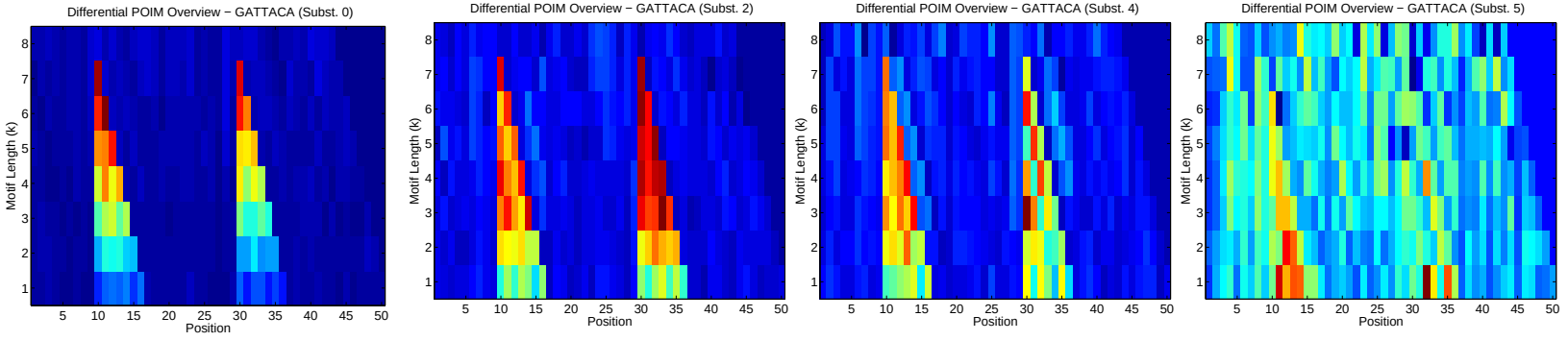
GATTACA und AGTAGTG an festen Positionen: 1000 Bsp.



w



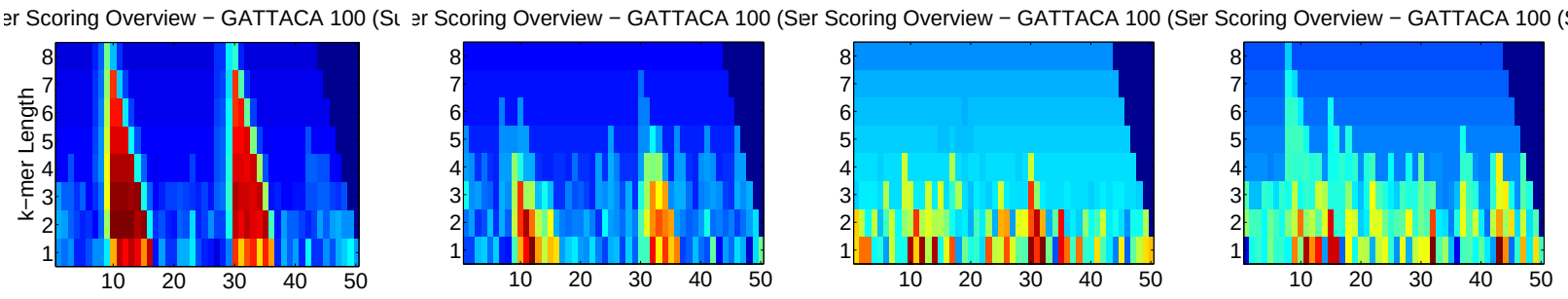
Q



GATTACA und AGTAGTG an festen Positionen: 100 Beispiele

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

w



Q