

Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Alexander Zien, Petra Philips, Sören Sonnenburg

Computing Positional Oligomer Importance Matrices (POIMs)

FIRST Reports 2/2007

FIRST Reports
Herausgegeben von
Prof. Dr.-Ing. Stefan Jähnichen

© Fraunhofer-Institut für Rechnerarchitektur und Softwaretechnik FIRST 2007

ISSN 1613-5024

FIRST Reports 2/2007

Alle Rechte vorbehalten.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Die FIRST Reports können bezogen werden über:

Fraunhofer-Institut für Rechnerarchitektur
und Softwaretechnik FIRST
Kekuléstraße 7
12489 Berlin

Tel.: ++49 (0)30 6392 18 00
Fax: ++49 (0)30 6392 18 05
E-Mail: first@first.fraunhofer.de
Internet: www.first.fraunhofer.de

Computing Positional Oligomer Importance Matrices (POIMs)

Alexander Zien[†,‡], Petra Philips[†], Sören Sonnenburg[‡]

[†] Friedrich Miescher Lab, Spemannstr. 39, 72076 Tübingen, Germany
[‡] Fraunhofer FIRST, Kekuléstr. 7, 12489 Berlin, Germany

December 7, 2007

Abstract

We show how to efficiently compute Positional Oligomer Importance Matrices (POIMs) which are a novel and powerful way to extract, rank, and visualize higher order (i.e. oligo-nucleotide) compositional information for nucleotide sequences. Given a scoring function for nucleotide sequences which is linear w.r.t. positionwise occurrences of oligomers, POIMs quantify the increase (or decrease) of the expected score caused by information about each k-mer at each position. We demonstrate how to obtain a recursive algorithm which enables us to efficiently compute POIMs by using string index data structures. This is especially useful for scoring functions whose linear weighting is sparse, as is the case for the scoring function produced by string kernel classifiers.

1 Introduction

1.1 Motivation

To visualize the nucleotide composition of nucleotide sequences related to a certain event or signal, one typically uses *sequence logos* [5] obtained from Position Weight Matrices (PWMs, e.g. [1]). A major drawback of PWMs is that they only capture zeroth-order information and thus cannot properly represent longer motifs appearing in an either-or fashion. To alleviate this problem, first order extensions have been suggested (e.g. WAMs in [8]). More recently much higher order information has been used to classify sequences, e.g. with kernel methods like Support Vector Machines [6]. While such methods are considerably more accurate, they do not allow an easy visualization of discriminative sequence features as previously done using sequence logos.

In [7] we proposed Positional Oligomer Importance Matrices (POIMs) as a novel and powerful way to visualize sequence motif scoring functions. We demonstrated that they enable us to systematically extract and visualize the discriminative features from SVM-based splice site and promoter detectors, which are the most accurate state-of-the-art classifiers for these tasks. In this report, we elaborate on the technicalities of our recursive procedure to compute POIMs.

1.2 Notation and Main Definitions

Let $\mathbf{X} \in \Sigma^N$ be random sequences of length N produced by a probabilistic source generating sequences over the alphabet Σ. For example, for subsequences of genomes, $\Sigma = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. Fixed sequences will be denoted by lower-case letters $\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$. Further we will write $\mathbf{x}[i]^k := x_i x_{i+1} \ldots \mathbf{x}_{i+k-1}$ to denote the substring of $\mathbf{x}$ that starts at position i and has length $|\mathbf{x}[i]^k| = k$. For simplicity of notation, in comparisons like $\mathbf{X}[i] = \mathbf{y}$, $\mathbf{X}[i]$ will denote the substring of $\mathbf{X}$ starting at position j that has the same length as the string compared to (here, $\mathbf{y}$), thus $\big(\mathbf{X}[i] = \mathbf{y}\big) := \big(\mathbf{X}[i]^{|\mathbf{y}|} = \mathbf{y}\big)$. Let $\mathcal{I} := \bigcup_{k=1}^{K} \big(\Sigma^k \times \{1, \ldots, N-k+1\}\big)$ be the index set containing to all possible pairs $(\mathbf{y}, i)$ of k-mers

$\mathbf{y}$ upto length K and their possible starting positions i in $\mathbf{X}$. We will call pairs $(\mathbf{y}, i) \in \mathcal{I}$ *positional oligomers* (POs).

Consider that we are given a *scoring function* for sequences which is a linear combination of indicator functions $\mathbb{I}\left\{\mathbf{X}[i]^{|\mathbf{y}|} = \mathbf{y}\right\}$,

$$s(\mathbf{x}) \;=\; \sum_{k=1}^{K} \sum_{i=1}^{n-k+1} w_{(\mathbf{y},i)} \mathbb{I}\left\{\mathbf{X}[i] = \mathbf{y}\right\} + b \;, \tag{1}$$

where $\mathbf{w}$ is a fixed vector in $\mathbb{R}^{|\mathcal{I}|}$, $b \in \mathbb{R}$. Such scoring functions arise in the context of statistical classifiers like classifiers based on Markov models or kernel-based classifiers with spectrum, weighted-degree and weighted degree with shifts [2, 3, 4, 7].

We define the *positional oligomer importance* of each positional oligomer $(\mathbf{z}, j) \in \mathcal{I}$ as the contribution of its occurrence in random sequences to the score (1),

$$Q(\mathbf{z}, j) \;:=\; \mathbb{E}\big[\, s(\mathbf{X}) \,\big|\, \mathbf{X}[j] = \mathbf{z} \,\big] - \mathbb{E}\left[s(\mathbf{X})\right]. \tag{2}$$

The computation of expectations requires a probability distribution. For our derivations in this paper we will assume a Markov model of order D for the random sequences $\mathbf{X}$. For the applications we are looking at in [7], a zeroth-order Markov model turns out to be sufficient.

2 Computation of Positional Oligomer Importances

Despite the (conceptual) simplicity of equation (2), the computation of POIMs is demanding, as it involves over $|\Sigma|^N$ values (for each of the $|\Sigma|^k$ k-mers $\mathbf{z}$, and each of the $N - k + 1$ positions). We make use of three observations to reduce the computational burden.

Observation 1 (Independent PO Terms Vanish)

There are two reasons why we use the subtractive normalization w.r.t. the unconditionally expected score in (2). The first one is conceptual: the magnitude of an expected score is hard to interpret by itself without knowing the cutoff value for the classification, and therefore it is more revealing to see how a feature would *change* the odds. The second one is computational: this normalization makes the problem tractable, especially when features are mostly independent.

To see why independent features are of computational advantage, recall that computation of a single (possibly conditional) expectation would require summing over all features (i.e. over $\mathcal{I}$),

$$\begin{aligned}
\mathbb{E}\left[s(X)\right] \;&=\; \mathbb{E}\left[\sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} \mathbb{I}\left\{\mathbf{X}[i] = \mathbf{y}\right\} + b\right] && (3)\\
&=\; \sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} \mathbb{E}\big[\mathbb{I}\left\{\mathbf{X}[i] = \mathbf{y}\right\}\big] + b && (4)\\
&=\; \sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i] = \mathbf{y}\right) + b \;, && (5)
\end{aligned}$$

and respectively

$$\begin{aligned}
\mathbb{E}\big[\, s(X) \,\big|\, \mathbf{X}[j] = \mathbf{z} \,\big] \;&=\; \mathbb{E}\left[\sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} \mathbb{I}\left\{\mathbf{X}[i] = \mathbf{y}\right\} + b \;\middle|\; \mathbf{X}[j] = \mathbf{z}\right] && (6)\\
&=\; \sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} \mathbb{E}\big[\, \mathbb{I}\left\{\mathbf{X}[i] = \mathbf{y}\right\} \,\big|\, \mathbf{X}[j] = \mathbf{z} \,\big] + b && (7)\\
&=\; \sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} Pr\big(\, \mathbf{X}[i] = \mathbf{y} \,\big|\, \mathbf{X}[j] = \mathbf{z} \,\big) + b \;. && (8)
\end{aligned}$$

By subtraction however, all features which are independent of $(\mathbf{z}, j)$ vanish from the difference because in this case the conditional probabilities are equal to the unconditional probabilities. That is,

$$Pr\big(\,\mathbf{X}[i] = \mathbf{y} \,\big|\, \mathbf{X}[j] = \mathbf{z}\,\big) \;\; = \;\; Pr\big(\mathbf{X}[i] = \mathbf{y}\big) \tag{9}$$

whenever $\mathbf{X}[i] = \mathbf{y}$ and $\mathbf{X}[j] = \mathbf{z}$ are statistically independent events, which we denote by $(\mathbf{y}, i) \perp (\mathbf{z}, j)$. Otherwise, i.e. if they are (possibly) dependent, we write $(\mathbf{y}, i) \not\perp (\mathbf{z}, j)$. Thus,

$$\begin{aligned} Q(\mathbf{z}, j) &= \mathbb{E}\big[\, s(X) \,\big|\, \mathbf{X}[j] = \mathbf{z}\,\big] - \mathbb{E}\left[s(X)\right] && (10)\\ &= \sum_{(\mathbf{y},i)\in\mathcal{I}} w_{(\mathbf{y},i)} \big[Pr\left(\,\mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z}\,\right) - Pr\left(\mathbf{X}[i] = \mathbf{y}\right)\big] && (11)\\ &= \sum_{(\mathbf{y},i)\in\mathcal{I},\, (\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} \big[Pr\left(\,\mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z}\,\right) - Pr\left(\mathbf{X}[i] = \mathbf{y}\right)\big] && (12) \end{aligned}$$

Note that, under the Markov model of order D, two POs $(\mathbf{y}, i)$ and $(\mathbf{z}, j)$ are always independent if they are separated by at least D positions. This means that either $i+|\mathbf{y}|-1 < j-D$ or $j+|\mathbf{z}|-1 < i-D$. Otherwise they are dependent in all but degenerate cases.

Observation 2 (Incompatible Conditional PO Terms Vanish)

Another property of PO pairs is that of compatibility. We say that $(\mathbf{y}, i)$ and $(\mathbf{z}, j)$ are *compatible*, denoted by $(\mathbf{y}, i) \sim (\mathbf{z}, j)$, if they agree on any shared positions they might have. For example, $(TATA, 30)$, and $(AAA, 32)$ are incompatible, since they share positions $\{32, 33\}$ but disagree on position 32, whereas $(TATA, 30)$ and $(TACCA, 32)$ are compatible. If $(\mathbf{y}, i)$ and $(\mathbf{z}, j)$ are incompatible, then it holds that $Pr\left(\,\mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z}\,\right) = 0$. Thus, the sum (12) can be simplified to range over less summands,

$$\begin{aligned} Q(\mathbf{z}, j) &= \sum_{(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)} w_{(\mathbf{y},i)} Pr\big(\,\mathbf{X}[i] = \mathbf{y} \,\big|\, \mathbf{X}[j] = \mathbf{z}\,\big) && (13)\\ &\quad - \sum_{(\mathbf{y},i)\in\mathcal{I},\, (\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i] = \mathbf{y}\right) \;\; , && (14) \end{aligned}$$

where we denote by

$$\mathcal{I}(\mathbf{z}, j) \;\; := \;\; \{\, (\mathbf{y}, i) \in \mathcal{I} \mid (\mathbf{y}, i) \not\perp (\mathbf{z}, j) \text{ and } (\mathbf{y}, i) \sim (\mathbf{z}, j) \,\} \tag{15}$$

the set of POs that are dependent on and compatible with $(\mathbf{z}, j)$.

Observation 3 (PO Importances are Weighted Sums of Conditional Terms)

As a last observation we show that once we precomputed *conditional* sums (13), we can use easily compute the positional oligomers importances.

Denote the two sum terms from (13) and (14) as

$$\begin{aligned} u(\mathbf{z}, j) &:= \sum_{(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)} Pr\left(\,\mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z}\,\right) w_{(\mathbf{y},i)} && (16)\\ v(\mathbf{z}, j) &:= \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i] = \mathbf{y}\right) \;\; . && (17) \end{aligned}$$

Thus, $Q(\mathbf{z}, j) = u(\mathbf{z}, j) - v(\mathbf{z}, j)$. To compute $v(\mathbf{z}, j)$, we show that we only have to use terms of the type $u(\mathbf{z}', j)$, with $|\mathbf{z}'| = |\mathbf{z}|$. First note, for Markov chains of any order D, for any $\mathbf{z}' \in \Sigma^{|\mathbf{z}|}$, the set

equality $\{\,(\mathbf{y},i)\in\mathcal{I}\mid(\mathbf{y},i)\not\perp(\mathbf{z},j)\,\}=\{\,(\mathbf{y},i)\in\mathcal{I}\mid(\mathbf{y},i)\not\perp(\mathbf{z}',j)\,\}$. With this,

$$
\begin{aligned}
v(\mathbf{z},j) &= \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i]=\mathbf{y}\right) && (18)\\
&= \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[i]=\mathbf{y}\wedge\mathbf{X}[j]=\mathbf{z}'\right) && (19)\\
&= \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[i]=\mathbf{y}\;\middle|\;\mathbf{X}[j]=\mathbf{z}'\right) Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) && (20)\\
&= \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i]=\mathbf{y}\;\middle|\;\mathbf{X}[j]=\mathbf{z}'\right) && (21)\\
&= \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) \sum_{(\mathbf{y},i)\not\perp(\mathbf{z},j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i]=\mathbf{y}\;\middle|\;\mathbf{X}[j]=\mathbf{z}'\right) && (22)\\
&= \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) \sum_{(\mathbf{y},i)\not\perp(\mathbf{z}',j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i]=\mathbf{y}\;\middle|\;\mathbf{X}[j]=\mathbf{z}'\right) && (23)\\
&= \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) \sum_{(\mathbf{y},i)\in\mathcal{I}(\mathbf{z}',j)} w_{(\mathbf{y},i)} Pr\left(\mathbf{X}[i]=\mathbf{y}\;\middle|\;\mathbf{X}[j]=\mathbf{z}'\right) && (24)\\
&= \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) u(\mathbf{z}',j)\ . && (25)
\end{aligned}
$$

Finally, we arrive at the following formula for the POIM computation, which shows that positional oligomers importances can be easily computed from the table of all conditional terms,

$$
Q(\mathbf{z},j) \;=\; u(\mathbf{z},j) - \sum_{\mathbf{z}'\in\Sigma^{|\mathbf{z}|}} Pr\left(\mathbf{X}[j]=\mathbf{z}'\right) u(\mathbf{z}',j)\ . \qquad (26)
$$

2.1 Recursive computation of $Q(\mathbf{z},j)$ for the general Markov chain of order d

From (26) we see that we can compute $Q(\mathbf{z},j)$ by summing over the index set $\mathcal{I}(\mathbf{z},j)$ *only* (instead of $\mathcal{I}$). However, even the reduced set $\mathcal{I}(\mathbf{z},j)$ of relevant POs is too large to allow for efficient naive summation over it for each PO $(\mathbf{z},j)$. We thus develop an efficient recursive algorithm which will allow for use of efficient tree structures. The crucial idea is to treat the POs in $\mathcal{I}(\mathbf{z},j)$ separately according to their relative position to $(\mathbf{z},j)$. To do so for the general Markov chain of order D, we subdivide the set $\mathcal{I}(\mathbf{z},j)$ into substrings, superstrings, left neighbors of $(\mathbf{z},j)$ with gaps of at most $D-1$, and right neighbors of $(\mathbf{z},j)$ with gaps of at most $D-1$. Figure 1 illustrates these four cases.

$$
\begin{aligned}
\text{(substrings)}\quad & \mathcal{I}^{\vee}(\mathbf{z},j) &:=\ & \{\,(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)\mid i\geq j\ \wedge\ i+|\mathbf{y}|\leq j+|\mathbf{z}|\,\} && (27)\\
\text{(superstrings)}\quad & \mathcal{I}^{\wedge}(\mathbf{z},j) &:=\ & \{\,(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)\mid i\leq j\ \wedge\ i+|\mathbf{y}|\geq j+|\mathbf{z}|\,\} && (28)\\
\text{(left neighb.)}\quad & \mathcal{I}^{<}(\mathbf{z},j) &:=\ & \{\,(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)\mid i<j\ \wedge\ i+|\mathbf{y}|<j+|\mathbf{z}|\ \wedge\ i+|\mathbf{y}|>j-D\,\} && (29)\\
\text{(right neighb.)}\quad & \mathcal{I}^{>}(\mathbf{z},j) &:=\ & \{\,(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)\mid i>j\ \wedge\ i+|\mathbf{y}|>j+|\mathbf{z}|\ \wedge\ j+|\mathbf{z}|>i-D\,\} && (30)
\end{aligned}
$$



Figure 1: An example for a substring, a superstring, a left partial overlap, and a right partial overlap $\mathbf{y}$ of the string $\mathbf{z}$ = AATACGTAC.

With these sets we can decompose $u(\mathbf{z}, j)$ defined in (16) as follows (note that $(\mathbf{z}, j)$ is element of both $\mathcal{I}^{\vee}(\mathbf{z}, j)$ and $\mathcal{I}^{\wedge}(\mathbf{z}, j)$, and that $w_{(\mathbf{z},j)}$ is thus counted twice):

$$u(\mathbf{z}, j) \;=\; u^{\vee}(\mathbf{z}, j) + u^{\wedge}(\mathbf{z}, j) + u^{<}(\mathbf{z}, j) + u^{>}(\mathbf{z}, j) - w_{(\mathbf{z},j)} \;\;, \tag{31}$$

where, for each $\star \in \{\vee, \wedge, >, <\}$:

$$u^{\star}(\mathbf{z}, j) \;:=\; \sum_{(\mathbf{y},i)\in\mathcal{I}^{\star}(\mathbf{z},j)} Pr\left(\, \mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z} \,\right) w_{(\mathbf{y},i)} \;\;. \tag{32}$$

We further define the auxilliary conditional sums of left and right extensions of $\mathbf{z} \in \Sigma^p$:

$$L(\mathbf{z}, j) \;:=\; \sum_{k=0}^{K-p} \sum_{\mathbf{t}\in\Sigma^k} \underbrace{Pr\Big(\, \mathbf{X}[j-k] = \mathbf{tz} \,\Big|\, \mathbf{X}[j] = \mathbf{z} \,\Big)}_{=Pr\left(\, \mathbf{X}[j-k]=\mathbf{t} \,\middle|\, \mathbf{X}[j]=\mathbf{z} \,\right)} w_{(\mathbf{tz},j-k)} \tag{33}$$

$$R(\mathbf{z}, j) \;:=\; \sum_{k=0}^{K-p} \sum_{\mathbf{t}\in\Sigma^k} \underbrace{Pr\Big(\, \mathbf{X}[j] = \mathbf{zt} \,\Big|\, \mathbf{X}[j] = \mathbf{z} \,\Big)}_{=Pr\left(\, \mathbf{X}[j+p]=\mathbf{t} \,\middle|\, \mathbf{X}[j]=\mathbf{z} \,\right)} w_{(\mathbf{zt},j)} \;\;. \tag{34}$$

As the following theorem shows, $u^{\vee}(\mathbf{z}, j)$, $u^{\wedge}(\mathbf{z}, j)$, $L(\mathbf{z}, j)$ and $R(\mathbf{z}, j)$ can be computed recursively, in a way which will enable us to use efficient tree data structures for their computation. Additionally, $u^{>}(\mathbf{z}, j)$ and $u^{>}(\mathbf{z}, j)$ can be computed from the tables of values of $L(\mathbf{z}, j)$ and $R(\mathbf{z}, j)$. This theorem, together with equations (16) and (26), enables us to compute PO importance matrices efficiently.

Theorem (Markov chains of order D). Let the sequences $\mathbf{X}$ be Markov chains of order D. For $0 \le |\mathbf{z}| \le K-2$ and $\sigma, \tau \in \Sigma$, we then have the following upward recursion for substrings:

$$u^{\vee}(\sigma, j) \;=\; w_{(\sigma,j)} \tag{35}$$

$$u^{\vee}(\sigma\mathbf{z}\tau, j) \;=\; w_{(\sigma\mathbf{z}\tau,j)} + u^{\vee}(\sigma\mathbf{z}, j) + u^{\vee}(\mathbf{z}\tau, j+1) - u^{\vee}(\mathbf{z}, j+1) \;\;. \tag{36}$$

Further, for $2 \le p \le K$, $p := |\mathbf{z}|$, we have the following downward recursions for superstrings and neighbor sequences:

$$u^{\wedge}(\mathbf{z}, j) \;=\; w_{(\mathbf{z},j)} - \frac{1}{Pr\left(\mathbf{X}[j] = \mathbf{z}\right)} \sum_{(\sigma,\tau)\in\Sigma^2} Pr\left(\mathbf{X}[j+1] = \sigma\mathbf{z}\tau\right) u^{\wedge}(\sigma\mathbf{z}\tau, j-1) \tag{37}$$

$$+ \frac{1}{Pr\left(\mathbf{X}[j] = \mathbf{z}\right)} \left[\sum_{\sigma\in\Sigma} Pr\left(\mathbf{X}[j] = \sigma\mathbf{z}\right) u^{\wedge}(\sigma\mathbf{z}, j-1) + \sum_{\tau\in\Sigma} Pr\left(\mathbf{X}[j+1] = \mathbf{z}\tau\right) u^{\wedge}(\mathbf{z}\tau, j) \right] \tag{38}$$

$$u^{<}(\mathbf{z}, j) \;=\; \sum_{\mathbf{t}\in\Sigma^D} \sum_{\sigma\in\Sigma} \frac{Pr\left(\mathbf{X}[j-D-1] = \sigma\mathbf{t}\right)}{Pr\left(\mathbf{X}[j-D] = \mathbf{t}\right)} \sum_{l=1}^{\min\{p+D,K\}-1} L\left(\sigma((\mathbf{tz})[1]^l), j-D-1\right) \tag{39}$$

$$u^{>}(\mathbf{z}, j) \;=\; \sum_{\mathbf{t}\in\Sigma^D} \sum_{\tau\in\Sigma} \frac{Pr\left(\mathbf{X}[j+p] = \mathbf{t}\tau\right)}{Pr\left(\mathbf{X}[j+p] = \mathbf{t}\right)} \sum_{l=1}^{\min\{p+D,K\}-1} R\left((\mathbf{zt})[p+D-l+1]^l\, \tau, j+p+D-l\right) \;\;, \tag{40}$$

with the recursion anchored at too long sequences by $u^{\wedge}(\mathbf{z}, j) = 0$, $u^{<}(\mathbf{z}, j) = 0$, and $u^{>}(\mathbf{z}, j) = 0$ for $p = |\mathbf{z}| \le K$. Here L and R are computed recursively for $p \le K$ by

$$L(\mathbf{z}, j) \;=\; w_{(\mathbf{z},j)} + \sum_{\sigma\in\Sigma} \frac{Pr\left(\mathbf{X}[j-1] = \sigma\mathbf{z}\right)}{Pr\left(\mathbf{X}[j] = \mathbf{z}\right)} L(\sigma\mathbf{z}, j-1) \tag{41}$$

$$R(\mathbf{z}, j) \;=\; w_{(\mathbf{z},j)} + \sum_{\tau\in\Sigma} \frac{Pr\left(\mathbf{X}[j] = \mathbf{z}\tau\right)}{Pr\left(\mathbf{X}[j] = \mathbf{z}\right)} R(\mathbf{z}\tau, j) \;\;, \tag{42}$$

with $L(\mathbf{z}, j) = R(\mathbf{z}, j) = 0$ otherwise (i.e. for $p > K$).

2.2 Recursive computation of $Q(\mathbf{z}, j)$ for the zeroth-order Markov distribution

For the special case of the zeroth-order Markov distribution, i.e. $D = 0$, the above equations simplify considerably. First, note that dependence of two POs $(\mathbf{y}, i)$ and $(\mathbf{z}, j)$ is easy to check: in all but degenerate cases, dependence is equivalent with being overlapping in $\mathbf{X}$. Thus, instead of considering neighbors with distances up to $D - 1$, we can now focus on overlaps.

The second simplification is that the zeroth-order Markov model decouples the positions completely, i.e. we have an independent single-symbol distribution at each position. Thus the probabilities in the theorem can be computed simply by

$$Pr\left(\mathbf{X}[i] = \mathbf{y}\right) = \prod_{l=1}^{|\mathbf{y}|} Pr\left(X_{i+l-1} = y_l\right) \quad . \tag{43}$$

To aid in understanding the recursions, we additionally define the following unions of sets containing all strings with the same suffix respectively prefix $\mathbf{z}$, i.e. complete overlaps of the positional p-mer $\mathbf{z}$:

$$\mathcal{L}(\mathbf{z}, j) \quad := \quad \bigcup_{l=0}^{K-p} \left\{ \left(\mathbf{t}\mathbf{z}, j - l\right) \,\middle|\, \mathbf{t} \in \Sigma^l \right\} \quad , \tag{44}$$

$$\mathcal{R}(\mathbf{z}, j) \quad := \quad \bigcup_{l=0}^{K-p} \left\{ \left(\mathbf{z}\mathbf{t}, j\right) \,\middle|\, \mathbf{t} \in \Sigma^l \right\} \quad , \tag{45}$$

Now we can decompose left and right partial overlaps as unions of sets with fixed prefixes and suffixes:

$$\mathcal{I}^{<}(\mathbf{z}, j) \quad = \quad \bigcup_{\sigma \in \Sigma} \bigcup_{l=1}^{p-1} \mathcal{L}\left(\sigma(\mathbf{z}[1]^l), j - 1\right) \tag{46}$$

$$\mathcal{I}^{>}(\mathbf{z}, j) \quad = \quad \bigcup_{\tau \in \Sigma} \bigcup_{l=1}^{p-1} \mathcal{R}\left(\mathbf{z}[p - l + 1]^l \tau, j\right) \quad . \tag{47}$$

The structure of these equations is mimiked by the recursions in the following theorem.

Corollary 1 (Markov Chains of order 0). Let the sequences $\mathbf{X}$ be Markov chains of order 0. For $2 \leq p \leq K$, $p := |\mathbf{z}|$, we then have the following downward recursions for superstrings and partial overlaps:

$$u^{\wedge}(\mathbf{z}, j) \quad = \quad w_{(\mathbf{z},j)} - \sum_{(\sigma,\tau) \in \Sigma^2} Pr\left(\mathbf{X}[j-1] = \sigma\right) Pr\left(\mathbf{X}[j+p] = \tau\right) u^{\wedge}(\sigma\mathbf{z}\tau, j - 1) \tag{48}$$

$$+ \sum_{\sigma \in \Sigma} Pr\left(\mathbf{X}[j-1] = \sigma\right) u^{\wedge}(\sigma\mathbf{z}, j - 1) + \sum_{\tau \in \Sigma} Pr\left(\mathbf{X}[j+p] = \tau\right) u^{\wedge}(\mathbf{z}\tau, j) \tag{49}$$

$$u^{<}(\mathbf{z}, j) \quad = \quad \sum_{\sigma \in \Sigma} Pr\left(\mathbf{X}[j-1] = \sigma\right) \sum_{l=1}^{\min\{p,K\}-1} L\left(\sigma(\mathbf{z}[1]^l), j - 1\right) \tag{50}$$

$$u^{>}(\mathbf{z}, j) \quad = \quad \sum_{\tau \in \Sigma} Pr\left(\mathbf{X}[j+p] = \tau\right) \sum_{l=1}^{\min\{p,K\}-1} R\left(\mathbf{z}[p - l + 1]^l \tau, j + p - l\right) \quad , \tag{51}$$

where L and R are computed recursively by

$$L(\mathbf{z}, j) \quad = \quad w_{(\mathbf{z},j)} + \sum_{\sigma \in \Sigma} Pr\left(\mathbf{X}[j-1] = \sigma\right) L(\sigma\mathbf{z}, j - 1) \tag{52}$$

$$R(\mathbf{z}, j) \quad = \quad w_{(\mathbf{z},j)} + \sum_{\tau \in \Sigma} Pr\left(\mathbf{X}[j+p] = \tau\right) R(\mathbf{z}\tau, j) \quad . \tag{53}$$

2.3 Recursive computation of $Q(\mathbf{z}, j)$ for the uniform distribution

We will also state the folowing corollary for the simplest possible case, for the uniform distribution over $\mathbf{X}$ (with length length $|\mathbf{X}| = N$), for which

$$Pr\left(\mathbf{X} = \mathbf{x}\right) \quad = \quad |\Sigma|^{-N} \ . \tag{54}$$

It is easy to see that this is equivalent to the assumption that at each position in the sequence, each element of the alphabet Σ is equally likely with probability $1/|\Sigma|$. It also implies that single characters at each position are independent of characters at all other positions, and $Pr\left(\mathbf{X}[j] = \mathbf{z}\right) = |\Sigma|^{-p}$, where $p = |\mathbf{z}|$. This makes the computation of PO importances much easier.

Corollary 2 (Uniform distribution). For the uniform distribution, with the notations from above, the PO importances $Q(\mathbf{z}, j) = u(\mathbf{z}, j) - v(\mathbf{z}, j)$ can be computed from

$$Q(\mathbf{z}, j) \quad = \quad u(\mathbf{z}, j) - \frac{1}{|\Sigma|^p} \sum_{\mathbf{z}' \in \Sigma^p} u(\mathbf{z}', j) \tag{55}$$

by the partial terms

$$u^{\wedge}(\mathbf{z}, j) \quad = \quad w_{(\mathbf{z},j)} - \frac{1}{|\Sigma|^2} \sum_{(\sigma,\tau)\in\Sigma^2} u^{\wedge}(\sigma\mathbf{z}\tau, j-1) + \frac{1}{|\Sigma|} \sum_{\sigma\in\Sigma} u^{\wedge}(\sigma\mathbf{z}, j-1) + \frac{1}{|\Sigma|} \sum_{\tau\in\Sigma} u^{\wedge}(\mathbf{z}\tau, j) \tag{56}$$

$$u^{<}(\mathbf{z}, j) \quad = \quad \frac{1}{|\Sigma|} \sum_{\sigma\in\Sigma} \sum_{l=1}^{p-1} L(\sigma\mathbf{z}[1]^l, j-1) \tag{57}$$

$$u^{>}(\mathbf{z}, j) \quad = \quad \frac{1}{|\Sigma|} \sum_{\tau\in\Sigma} \sum_{l=1}^{p-1} R(\mathbf{z}[p-l+1]^l \tau, j+p-l) \ , \tag{58}$$

and L and R are computed recursively by

$$L(\mathbf{z}, j) \quad = \quad w_{(\gamma,j)} + \frac{1}{|\Sigma|} \sum_{\sigma\in\Sigma} L(\sigma\mathbf{z}, j-1) \tag{59}$$

$$R(\mathbf{z}, j) \quad = \quad w_{(\gamma,j)} + \frac{1}{|\Sigma|} \sum_{\tau\in\Sigma} R(\mathbf{z}\tau, j) \ . \tag{60}$$

3 Proof of Theorem

We show how to compute the values of the functions $u^{\vee}$, $u^{\wedge}$, $u^{<}$, and $u^{>}$ recursively. Recall first that for general Markov chains of order D

$$Pr\left(\mathbf{X}[i] = \mathbf{y}\right) \quad = \quad \prod_{l=1}^{D} Pr\left(\, X_{i+l-1} = y_l \,\middle|\, \mathbf{X}[i]^{l-1} = y[1]^{l-1} \,\right) \tag{61}$$

$$\times \prod_{l=D+1}^{|\mathbf{y}|} Pr\left(\, X_{i+l-1} = y_l \,\middle|\, \mathbf{X}[i+l-D-1]^D = y[l-D]^D \,\right) \ . \tag{62}$$

3.1 Substrings

Note that for any substring $(\mathbf{y}, i) \in \mathcal{I}^{\vee}(\mathbf{z}, j)$, it holds that

$$Pr\left(\, \mathbf{X}[i] = \mathbf{y} \mid \mathbf{X}[j] = \mathbf{z} \,\right) \quad = \quad 1 \ . \tag{63}$$

Therefore,

$$u^{\vee}(\mathbf{z},j) \;=\; \sum_{(\mathbf{y},i)\in\mathcal{I}^{\vee}(\mathbf{z},j)} w_{(\mathbf{y},i)} \;\;. \tag{64}$$

For $|\mathbf{z}| \geq 2$, let $\sigma,\tau \in \Sigma$ such that $\mathbf{z} = \sigma\mathbf{t}\tau$. Then the following set relations are easy to see:

$$\mathcal{I}^{\vee}(\mathbf{z},j) \;=\; \{(\mathbf{z},j)\} \cup \mathcal{I}^{\vee}(\sigma\mathbf{t},j) \cup \mathcal{I}^{\vee}(\mathbf{t}\tau,j+1) \tag{65}$$

$$\begin{aligned}
\mathcal{I}^{\vee}(\sigma\mathbf{t},j) \cap \mathcal{I}^{\vee}(\mathbf{t}\tau,j+1) &= \mathcal{I}^{\vee}(\mathbf{t},j+1) && (66)\\
\mathcal{I}^{\vee}(\sigma\mathbf{t},j) \cap \{(\mathbf{z},j)\} &= \emptyset && (67)\\
\mathcal{I}^{\vee}(\mathbf{t}\tau,j+1) \cap \{(\mathbf{z},j)\} &= \emptyset \;\;. && (68)
\end{aligned}$$

Thus, for $|\mathbf{z}| \geq 2$, we can set up the following recursion:

$$\begin{aligned}
u^{\vee}(\mathbf{z},j) &= \sum_{(\mathbf{y},i)\in\mathcal{I}^{\vee}(\mathbf{z},j)} w_{(\mathbf{y},i)} && (69)\\
&= w_{(\mathbf{z},j)} + \sum_{(\mathbf{y},i)\in\mathcal{I}^{\vee}(\sigma\mathbf{t},j)} w_{(\mathbf{y},i)} + \sum_{(\mathbf{y},i)\in\mathcal{I}^{\vee}(\mathbf{t}\tau,j+1)} w_{(\mathbf{y},i)} - \sum_{(\mathbf{y},i)\in\mathcal{I}^{\vee}(\mathbf{t},j+1)} w_{(\mathbf{y},i)} && (70)\\
&= w_{(\mathbf{z},j)} + u^{\vee}(\sigma\mathbf{t},j) + u^{\vee}(\mathbf{t}\tau,j+1) - u^{\vee}(\mathbf{t},j+1) \;\;. && (71)
\end{aligned}$$

3.2 Superstrings

For superstrings $(\mathbf{y},i) \in \mathcal{I}^{\wedge}(\mathbf{z},j)$, it is easy to see that

$$Pr\,(\,\mathbf{X}[i]=\mathbf{y}\;|\,\mathbf{X}[j]=\mathbf{z}\,) \;=\; \frac{Pr\,(\mathbf{X}[i]=\mathbf{y}\;\wedge\;\mathbf{X}[j]=\mathbf{z})}{Pr\,(\mathbf{X}[j]=\mathbf{z})} = \frac{Pr\,(\mathbf{X}[i]=\mathbf{y})}{Pr\,(\mathbf{X}[j]=\mathbf{z})} \;\;. \tag{72}$$

Note that

$$\mathcal{I}^{\wedge}(\mathbf{z},j) \;=\; \{(\mathbf{z},j)\} \cup \left[\bigcup_{\sigma\in\Sigma} \mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1)\right] \cup \left[\bigcup_{\sigma\in\Sigma} \mathcal{I}^{\wedge}(\mathbf{z}\sigma,j)\right] \tag{73}$$

$$\begin{aligned}
\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1) \cap \mathcal{I}^{\wedge}(\mathbf{z}\tau,j) &= \mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j-1), && \forall\sigma,\tau\in\Sigma\\
\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1) \cap \{(\mathbf{z},j)\} &= \emptyset, && \forall\sigma\in\Sigma\\
\mathcal{I}^{\wedge}(\mathbf{z}\tau,j) \cap \{(\mathbf{z},j)\} &= \emptyset, && \forall\tau\in\Sigma\\
\mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j-1) \cap \mathcal{I}^{\wedge}(\sigma'\mathbf{z}\tau',j-1) &= \emptyset, && \forall(\sigma,\tau)\neq(\sigma',\tau')\\
\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1) \cap \mathcal{I}^{\wedge}(\sigma'\mathbf{z},j-1) &= \emptyset, && \forall\sigma\neq\sigma'\\
\mathcal{I}^{\wedge}(\mathbf{z}\tau,j) \cap \mathcal{I}^{\wedge}(\mathbf{z}\tau',j) &= \emptyset, && \forall\tau\neq\tau' \;\;.
\end{aligned} \tag{74}$$

Thus

$$\begin{aligned}
\left[\bigcup_{\sigma\in\Sigma} \mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1)\right] \cap \left[\bigcup_{\sigma\in\Sigma} \mathcal{I}^{\wedge}(\mathbf{z}\sigma,j)\right] &= \bigcup_{(\sigma,\tau)\in\Sigma^2} \big(\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1) \cap \mathcal{I}^{\wedge}(\mathbf{z}\tau,j)\big) && (75)\\
&= \bigcup_{(\sigma,\tau)\in\Sigma^2} \big(\mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j-1)\big) && (76)
\end{aligned}$$

and therefore

$$\begin{aligned}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\mathbf{z},j)} \cdots &= \sum_{(\mathbf{y},i)=(\mathbf{z},j)} \cdots + \sum_{(\mathbf{y},i)\in\left[\bigcup_{\sigma\in\Sigma}\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1)\right]} \cdots \\ &\quad + \sum_{(\mathbf{y},i)\in\left[\bigcup_{\sigma\in\Sigma}\mathcal{I}^{\wedge}(\mathbf{z}\sigma,j)\right]} \cdots - \sum_{(\mathbf{y},i)\in\bigcup_{(\sigma,\tau)\in\Sigma^2}(\mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j-1))} \cdots \\ &= \sum_{(\mathbf{y},i)=(\mathbf{z},j)} \cdots + \sum_{\sigma\in\Sigma}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1)} \cdots + \sum_{\tau\in\Sigma}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\mathbf{z}\tau,j)} \cdots - \sum_{(\sigma,\tau)\in\Sigma^2}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j-1)} \cdots\end{aligned} \tag{77}$$

It follows that

$$\begin{aligned}u^{\wedge}(\mathbf{z},j) &= \frac{1}{Pr\left(\mathbf{X}[j]=\mathbf{z}\right)} \sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\mathbf{z},j)} Pr\left(\mathbf{X}[i]=\mathbf{y}\right) w_{(\mathbf{y},i)} \\ &= \frac{1}{Pr\left(\mathbf{X}[j]=\mathbf{z}\right)}\Bigg[Pr\left(\mathbf{X}[j]=\mathbf{z}\right) w_{(\mathbf{z},j)} + \sum_{\sigma\in\Sigma}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\sigma\mathbf{z},j-1)} Pr\left(\mathbf{X}[i]=\mathbf{y}\right) w_{(\mathbf{y},i)} \\ &\qquad + \sum_{\tau\in\Sigma}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\mathbf{z}\tau,j-1)} Pr\left(\mathbf{X}[i]=\mathbf{y}\right) w_{(\mathbf{y},i)} \\ &\qquad - \sum_{(\sigma,\tau)\in\Sigma^2}\sum_{(\mathbf{y},i)\in\mathcal{I}^{\wedge}(\sigma\mathbf{z}\tau,j+1)} Pr\left(\mathbf{X}[i]=\mathbf{y}\right) w_{(\mathbf{y},i)}\Bigg]\end{aligned} \tag{78}$$

By considering the definition of $u^{\wedge}$ and correcting for the conditional probabilities, we can set up the following recursion:

$$u^{\wedge}(\mathbf{z},j) = w_{(\mathbf{z},j)} - \frac{1}{Pr\left(\mathbf{X}[j]=\mathbf{z}\right)} \sum_{(\sigma,\tau)\in\Sigma^2} Pr\left(\mathbf{X}[j-1]=\sigma\mathbf{z}\tau\right) u^{\wedge}(\sigma\mathbf{z}\tau,j-1) \tag{79}$$

$$+ \frac{1}{Pr\left(\mathbf{X}[j]=\mathbf{z}\right)} \sum_{\sigma\in\Sigma}\left[Pr\left(\mathbf{X}[j-1]=\sigma\mathbf{z}\right) u^{\wedge}(\sigma\mathbf{z},j-1) + Pr\left(\mathbf{X}[j]=\mathbf{z}\sigma\right) u^{\wedge}(\mathbf{z}\sigma,j)\right] \tag{80}$$

3.3 Left and right neighbors

Recall the definition of $\mathcal{L}(\mathbf{z},j)$ and $\mathcal{R}(\mathbf{z},j)$ as unions of sets containing all strings with the same suffix respectively prefix $\mathbf{z}$ of length $p := |\mathbf{z}|$:

$$\mathcal{L}(\mathbf{z},j) := \bigcup_{l=0}^{K-p}\left\{\left(\mathbf{t}\mathbf{z},j-l\right)\,\middle|\,\mathbf{t}\in\Sigma^l\right\}\ , \tag{81}$$

$$\mathcal{R}(\mathbf{z},j) := \bigcup_{l=0}^{K-p}\left\{\left(\mathbf{z}\mathbf{t},j\right)\,\middle|\,\mathbf{t}\in\Sigma^l\right\}\ , \tag{82}$$

Generalizing the zeroth-order case in (46) and (47) to $D \geq 0$, they allow us to further decompose the sets of left and right neighbors as unions of *disjoint* sets with fixed prefixes and suffixes:

$$\mathcal{I}^{<}(\mathbf{z},j) = \bigcup_{\mathbf{t}\in\Sigma^D}\bigcup_{\sigma\in\Sigma}\bigcup_{l=1}^{p+D-1}\mathcal{L}\left(\sigma((\mathbf{t}\mathbf{z})[1]^l),j-D-1\right) \tag{83}$$

$$\mathcal{I}^{>}(\mathbf{z},j) = \bigcup_{\mathbf{t}\in\Sigma^D}\bigcup_{\tau\in\Sigma}\bigcup_{l=1}^{p+D-1}\mathcal{R}\left((\mathbf{z}\mathbf{t})[p+D-l+1]^l\,\tau,j\right)\ . \tag{84}$$

Thus we can write

$$u^{<}(\mathbf{z},j) = \sum_{\mathbf{t}\in\Sigma^D}\sum_{\sigma\in\Sigma}\sum_{l=1}^{p+D-1}\sum_{(\mathbf{y},i)\in\mathcal{L}\left(\sigma((\mathbf{tz})[1]^l),j-D-1\right)} Pr\left(\mathbf{X}[i]=\mathbf{y}\mid\mathbf{X}[j]=\mathbf{z}\right) w_{(\mathbf{y},i)} \quad (85)$$

$$u^{>}(\mathbf{z},j) = \sum_{\mathbf{t}\in\Sigma^D}\sum_{\tau\in\Sigma}\sum_{l=1}^{p+D-1}\sum_{(\mathbf{y},i)\in\mathcal{R}\left((\mathbf{zt})[p+D-l+1]^l\tau,j\right)} Pr\left(\mathbf{X}[i]=\mathbf{y}\mid\mathbf{X}[j]=\mathbf{z}\right) w_{(\mathbf{y},i)} \ . \quad (86)$$

By considering the definition of L and R and correcting for the conditional probabilities we obtain equations (39) and (40).

References

[1] M. Gribskov, A.D. McLachlan, and D. Eisenberg. Profile analysis: Detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84:4355–4358, 1987.

[2] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In R.B. Altman et al., editor, *PSB*. River Edge, NJ, World Scientific, 2002.

[3] G. Rätsch and S. Sonnenburg. Accurate splice site detection for *C. elegans*. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Comput. Biol.* MIT Press, 2004.

[4] G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: Recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

[5] T.D. Schneider and R.M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.*, 18:6097–6100, 1990.

[6] B. Schölkopf and A. J. Smola. *Learning with Kernels*. Cambridge, MA, MIT Press, 2002.

[7] S. Sonnenburg, A. Zien, P. Philips, and G. Rätsch. Positional Oligomer Importance Matrices. *Workshop "Machine Learning in Computational Biology"*, 2007. NIPS 2007, Whistler, Canada (Accepted).

[8] M. Q. Zhang and T. G. Marr. A weight array method for splicing signal analysis. *Comput Appl Biosci.*, 9(5):499–509, October 1993.