

Weight matrix matching

Jakub Orzechowski Westholm, Adam Ameer

26th May 2003

The following is a description of our approach for matching weight matrices to nucleotide sequences. Given a weight matrix and a sequence, the problem is to determine whether the sequence is likely to be sampled from the matrix or not. The main benefit of our way of doing it is that our approach, unlike many others, doesn't rely on arbitrarily selected threshold values. The theory behind our weight matrix matching is described in section 1.

We have implemented two applications of the weight matrix matching. The first is a program that can be used for detecting transcription factor binding sites (represented as weight matrices) in the upstream regions of genes (nucleotide sequences). The second is a program that aligns multiple short nucleotide sequences. The two applications are described in section 2 and 3.

1 Theory

A weight matrix W is a matrix:

$$W = \begin{bmatrix} N_{1A} & N_{1C} & N_{1G} & N_{1T} \\ N_{2A} & N_{2C} & N_{2G} & N_{2T} \\ \vdots & \vdots & \vdots & \vdots \\ N_{lA} & N_{lC} & N_{lG} & N_{lT} \end{bmatrix}$$

where l is the length of W , and $W[i, \alpha] = N_{i\alpha}$ is the number of occurrences of base α at position i . Let W_i be the total number of bases at a given position,

$$W_i = \sum_{\alpha \in A, C, G, T} N_{i\alpha}$$

Let ω_i^α be the frequency of base α at position i ,

$$\omega_i^\alpha = \frac{N_{i\alpha}}{W_i}$$

These notations are used in the definition of the entropy of a position in a weight matrix.

Definition 1.1 *The entropy of position i in a weight matrix W is defined as*

$$E_i(W) = \frac{1}{1 - \beta} \log \left(\sum_{\alpha \in \{A,C,G,T\}} (\omega_i^\alpha)^\beta \right)$$

The entropy is a measure of the information content at a certain position. Positions with low entropies have a lot of variation of the frequencies ω_i^α (and thus a high information content), whereas positions with high entropies have more similar frequencies. The parameter β is a non-negative number.

Other important definitions are about adding nucleotide sequences to weight matrices.

Definition 1.2 *A nucleotide sequence S of length l is a sequence $S = \{s_i\}_{i=1}^l$ where $s_i \in \{A, C, G, T\}$, $1 \leq i \leq l$*

Definition 1.3 *A nucleotide sequence S can be added to a weight matrix W of the same length l to obtain a new weight matrix W_S , where*

$$W_S [i, \alpha] = \begin{cases} W [i, \alpha] + 1 & \text{if } s_i = \alpha \\ W [i, \alpha] & \text{otherwise} \end{cases}$$

The idea behind our matching is that a sequence that matches a weight matrix must decrease the entropy of the positions when being added to it. More formally, we define a score that distinguishes between matching and non-matching sequences.

Definition 1.4 *Let W be a weight matrix.*

A sequence S matches $W \iff \text{Score}(W, S) \geq 0$, where

$$\text{Score}(W, S) = \sum_{i=1}^l W_i (E_i(W) - E_i(W_S))$$

In the following two sections, we show how the definition above is used in practice in the programs 'wm_match' and 'wm_align'.

2 'wm_match' - Matching weight matrices to sequences

The most obvious application of the weight matrix matching is to use the score in definition 1.4 to build an algorithm for matching weight matrices to nucleotide sequences. Our implementation, 'wm_match', is described in this section.

The algorithm

The parameter β in the definition of entropy (definition 1.1) is set to a small value ($\beta = 0.0001$). Small values of β gives the kind of behaviour that we want when computing the score: the penalty of a mismatch at a position is greater than the benefit of a match. Our algorithm searches for small sections of the nucleotide sequence that match a given weight matrix. This is done by computing the score of matching the weight matrix to every possible subsequence (of same length as the weight matrix). Every time when the score is non-negative, a new match is reported.

Executing the program

The code is written in C and consists of the two modules `wm_match` and `weight_matrix`. The program is executed by the command:

```
>./wm_match [Weight_matrix_file] [Sequence_file] [Revcomp]
```

- **Weight_matrix_file** contains a weight matrix on the form:

```
N[1,A] N[1,C] N[1,G] N[1,T]
N[2,A] N[2,C] N[2,G] N[2,T]
...    ...    ...    ...
N[l,A] N[l,C] N[l,G] N[l,T]
```

- **Sequence_file** is a file with nucleotide sequences on fasta format.
- **Revcomp** is an optional flag. If it is set to `'-revcomp'`, we report matches on both the given sequence and its reverse complement strand. Otherwise, only the given sequence is scanned.
- **Output** from the program is written to stdout, on the format:

```
>Sequence_id1 Hit1
>Sequence_id2 Hit2
...
>Sequence_idN HitN
```

The `Sequence_ids` are the ids stored in the fasta headers. The Hits are the number of matches for the sequences in the fasta file to the weight matrix.

We have optimized the code to make the program run as fast as possible and tested the performance of our program against fuzznuc, a publicly available program for aligning short nucleotide sequences to longer sequences. It is only possible to compare the performance of the two programs when matching sequences that are totally unambiguous, since the fuzznuc program doesn't handle weight matrices. The two programs gave the same results when matching short sequences to a file with long nucleotide sequences, but our program was considerably faster.

3 'wm_align' - Multiple alignment of sequences

We have also constructed a multiple alignment algorithm based on the weight matrix matching. The algorithm is suitable for short nucleotide sequences that shall be aligned without gaps. Like the program 'wm_match', 'wm_align' uses the score in definition 1.4.

The algorithm

Input is a set of short DNA sequences that we want to align in the best possible way. The result of the alignment is a weight matrix. For example, the sequences ATCGGTT, CGGTT, CCGGT and ACCTGT can be aligned to:

```
ATCGGTT
  CGGTT
   CCGGT
    ACCTGT
```

In weight matrix form, this looks like:

```
A T C G
2 0 0 0
0 2 0 1
0 4 0 0
0 0 3 1
0 0 4 0
0 0 0 4
0 0 0 2
```

The idea behind our alignment algorithm is to add one sequence at a time to a weight matrix. The weight matrix is longer than the sequences, so there are several possible ways of adding them. One problem is to find the best way of doing it. Another is that the result is depending on the order in which we add the sequences to the weight matrix, and it is not clear what order to choose. We deal

with both these problems with heuristics based on the score from definition 1.4. In the pseudo-code algorithm below, the following functions are used:

- **empty()** Returns an empty weight matrix
- **add(s,W)** Adds the sequence **s** to a weight matrix **W** in the best possible way, i.e the way that gives the best score. The resulting weight matrix is returned.
- **bestScore(s,W)** Returns the score of adding the sequence **s** to the weight matrix **W** in the best possible way.
- **entropy(W)** Returns the entropy of a weight matrix.

```
S = {all sequences}
for all s in S
  We <- empty()
  Wstart <- add(s,We)
  S' <- S-{s}
  for all s' in S'
    Compute bestScore(s',Wstart)
  O <- all sequences in S' ordered according bestScore
  Ws = Wstart
  for all o in O
    Ws <- add(o,Ws)
  Compute entropy(Ws)
return the weight matrix with the lowest entropy
```

It is possible to make an implementation of the algorithm above so that the reverse complement of sequences may also be aligned. When computing the best score of a match between a sequence and a weight matrix, we also consider all possibilities of matching the reverse complement of the sequence. Then, either the sequence or its reverse complement is added to the weight matrix (or possibly none of them).

Executing the program

The code is written in C, and basically consists of the two modules `wm_align` and `weight_matrix`. The program is executed by the command:

```
> ./wm_align [Sequence_file] [Revcomp]
```

- **Sequence_file** is a file short nucleotide sequences (at most 50 bp)

```
DNA_sequence1
DNA_sequence2
...
DNA_sequenceN
```

The DNA sequences consist of the letters A,C,G,T.

- **Revcomp** is an optional flag. If it is set to '-revcomp', also the reverse complement of the DNA sequences are considered.
- **Output** from the program is a weight matrix that is written to stdout:

```
N[1,A] N[1,C] N[1,G] N[1,T]
N[2,A] N[2,C] N[2,G] N[2,T]
...    ...    ...    ...
N[l,A] N[l,C] N[l,G] N[l,T]
```

$N[i, \alpha]$ is the number of sequences in the resulting multiple alignment with base α at position i .