

# Competition Results of Weeder

Giulio Pavesi

February 2, 2004

## Running Parameters

On each dataset (of  $k$  sequences), we ran the program with the following parameters:

Find all patterns that appear in  $q$  out of  $k$  sequences, of length  $m$  with  $e$  mutations, with:

- $m = 6, e = 1, q = k$ ;
- $m = 6, e = 1, q = \lceil k/2 \rceil$ ;
- $m = 8, e = 1, q = k$ ;
- $m = 8, e = 1, q = \lceil k/2 \rceil$ ;
- $m = 8, e = 2, q = k$ ;
- $m = 8, e = 2, q = \lceil k/2 \rceil$ ;
- $m = 8, e = 3, q = k$ ;
- $m = 8, e = 3, q = \lceil k/2 \rceil$ ;
- $m = 10, e = 2, q = k$ ;
- $m = 10, e = 2, q = \lceil k/2 \rceil$ ;
- $m = 10, e = 3, q = k$ ;
- $m = 10, e = 3, q = \lceil k/2 \rceil$ ;
- $m = 10, e = 4, q = k$ ;
- $m = 10, e = 4, q = \lceil k/2 \rceil$ ;
- $m = 12, e = 4, q = k$ ;
- $m = 12, e = 4, q = \lceil k/2 \rceil$ ;

We did not consider reverse complements. Then, on each run, we scored all patterns satisfying the constraints, and reported the *five* highest scoring motifs (see next section).

## Scoring Motifs

In this section I describe how we scored motifs. This is because in the original Weeder paper the scoring method was just hinted, and also because some of the stuff is quite new.

For patterns of length 6 and 8, we directly used the corresponding oligonucleotide frequencies in the regulatory regions of the organisms the datasets belong to (taken from the RSA tools database of J.van Helden). In other words, given a pattern  $p$  of length  $m$ , we estimated its frequency as:

$$E(p) = \frac{obs(p)}{total_m}$$

where  $obs(p)$  is the number of times  $p$  was found in the regulatory regions, and  $total_m$  is the overall number of length  $m$  oligos in the sequences. The sequences were about 6000 for yeast, 26000 for mand and mouse, and so on. For longer patterns ( $m = 10$  and  $m = 12$ ) we started from length  $m = 8$ , that is, we used a Markov model of order 8.

Then, if  $p$  appears with  $e$  mutations, we estimate its expected frequency as:

$$E(p, e) = \sum_{p' \in \mathcal{H}(p, e)} \frac{obs(p')}{total_m}$$

where  $\mathcal{H}(p, e)$  is the set of patterns within Hamming distance  $e$  from  $p$ . For longer patterns we proceed in a similar way.

Then, let  $p$  be a pattern that appears with at most  $e$  mutations in  $q$  sequences of the set of sequences  $S = S_1 \dots S_k$ , of length  $l_1, \dots, l_k$ . First of all, we use a score based on its best occurrence in each sequence. Let  $e_i$  be the minimum number of mutations  $p$  appears with in the  $i$ -th sequence. Then, the sequence-specific score  $Seq(p)$  of  $p$  is given by:

$$Seq(p) = \sum_i \log \frac{1}{E(p, e_i) \cdot l_i}$$

summed over the sequences  $p$  appears in. Thus, this measure reflects how much  $p$  is conserved among the sequences of the dataset. Then, we also associate with  $p$  a general score, just considering how many times it appears with at most  $e$  mutations:

$$Glo(p) = \log \frac{obs(p, e)}{E(p, e) \cdot L}$$

where  $L = \sum l_i$ . All in all, the score of each pattern  $p$  appearing with at most  $e$  mutations in at least  $q$  sequences of the dataset is given by:

$$Score(p) = Seq(p) + Glo(p)$$

We used this score for all the datasets, except those containing two or one sequence. In the latter case, we just computed  $Glo(p)$ .

All in all, we ran the algorithm as explained in the previous section, and ranked the patterns reported in each run. We saved the five highest scoring patterns of each run.

## Choosing Motifs

This was the first major headache. Usually, if one knows the length in advance, finding the correct motif in say, the top ten, is reported as a success. Or, in case of a real case study, one usually looks in databases in order to check if one of the best motifs resembles something already known (unfortunately, answering 'the third-ranking pattern of length 10 because it is in TRANSFAC' was not considered acceptable by the organizers:) So, we had to find some criterion to choose which length was the correct one, and, in case of different results with different parameters, which was the run to trust. Comparing directly the scores, in our previous tests, did not yield good results.

But, in the same tests (like those we took from the MIRA paper and reproduced) we noticed another thing. Real motifs are 'redundant' in the output. This is best explained with an example. Suppose the real motif is **ASGT**. Since Weeder does not make any assumption on the position of the mutations, we can reasonably expect (or, better, hope) both **ACGT** and **AGGT** to appear among the (five, in our tests) highest-scoring patterns of a run with length four and one mutation.

Then, suppose the real motif is **TASGT**. Then, in the same run of length four, we also expect to find, for example **TACG** that overlaps **ACGT**.

If either thing happens, that is, there are two patterns differing by just one letter among the higher scoring ones) we say that a pattern is *vertically* redundant (since we put our results in an Excel table, and each run had one column).

Then, suppose the real motif is something like **TASGTA**. We expect to find **ACGT** (and/or **TACG**, **TACG** and so on) in the length four run. And, of course, both **TACGTA** and **TAGGTA** in the length six run. Whenever a pattern in the top ranking is a substring of a longer reported pattern (or vice versa, when has one of its substrings among the shortest ones) we say that it is *horizontally* redundant (the Excel columns again).

Horizontal redundant patterns were extracted by comparing the results of the following runs (see first section):

- $m = 6, e = 1; m = 8, e = 2; m = 10, e = 3; m = 12, e = 4$  all with  $q = k$ ;
- $m = 6, e = 1; m = 8, e = 2; m = 10, e = 3; m = 12, e = 4$  all with  $q = \lceil k/2 \rceil$ ;
- $m = 6, e = 1; m = 8, e = 3; m = 10, e = 4$ ; all with  $q = k$ ;
- $m = 6, e = 1; m = 8, e = 3; m = 10, e = 4$ ;  $q = \lceil k/2 \rceil$ ;
- $m = 6, e = 1; m = 8, e = 1$ ; all with  $q = k$ ;

- $m = 6, e = 1; m = 8, e = 1; q = \lceil k/2 \rceil$ ;

All in all, for each of the six above combinations, we report a pattern  $p$  of length  $m$  as 'motif' if:

1. It is the top-scoring motif of a run.
2. It is vertically redundant in its run.
3. By considering patterns of length either  $m + 2$  or  $m - 2$  it is horizontally redundant.

The sole exception are patterns of length 6, that must be top-scoring and horizontally redundant only (since it might happen that the pattern appears with no mutation).

If no pattern has these properties in all the combinations, we simply report 'no motif'.

More than a single pattern can satisfy this constraints. If the patterns are just one substring of the other, we report the longest vertically redundant pattern. If the patterns are of different length (i.e. one is not a substring of the other, say, there's one candidate for lengths 6 and 8 and another one for 10 and 12) we report the one that has the highest number of 'relatives' (both in horizontal or in vertical). If the tie happens for patterns of the same length (deriving from different combinations of parameters), (I don't actually remember whether this ever happened in the tests) we just report the highest-scoring one.

Although it may seem complicated, all the above stuff can be performed automatically with a simple script. Thus, we just collected the results, and fed everything to the script, that reported the 'motif' (if any).

We chose these criteria basically to limit the number of false positives (and have a general approach valid for each dataset). In this way, we probably can miss many motifs, since we privilege those that have a conserved core. We could have relaxed point 1 above, checking whether any of the highest ranking motifs was redundant and so on, but in this case it was more common to have different patterns fulfilling the requirements, and the simple rule-of-thumb just described for 'choosing among the chosen' should have been further modified.

## Reporting Occurrences

This was major headache number two. At the end of the script described in the previous section, we have the chosen pattern (if any) with the corresponding number of errors  $e$  and quorum  $q$ . Now: which of its occurrences should be reported? A priori, every occurrence with  $e$  mutations. But, this would have generated again a number of false positives.

Then, we wrote another script. It collects all the occurrences of the pattern with at most  $e$  mutations, and from them builds a frequency matrix (profile). Each of the collected occurrences is scored against the profile. Let  $p' = p_1 \dots p_m$  be an occurrence, and  $M$  be the  $4 \times m$  profile matrix. Then, its score is computed as

$$M(p) = \sum_{j=1}^m \log m_{i,j}$$

where  $m_{i,j}$  is the entry of the matrix corresponding to the  $j$ -th nucleotide of  $p$  (each column of the matrix is modified adding a pseudo-count of .001 so to avoid  $\log 0$  values). Then, we also compute  $Max(M)$  and  $Min(M)$  by summing over the maximum and minimum entry of each column of the matrix. All in all, the score of each occurrence  $p'$  of the pattern is scored with

$$Occ(p') = \frac{100 \times (M(p) - Min(M))}{Max(M) - Min(M)}$$

so to have for each occurrence a percentage value between 0 and 100. All occurrences with score greater than 90% are kept. It might happen that, even if a motif was reported to appear in all the sequences, its score in one or more sequences is lower than this threshold. Finally: since motifs are required to be vertical-redundant, we report as occurrences the intersection of the occurrences of the motif and of the other redundant (vertically) patterns, each set of occurrences computed with the corresponding matrix.

This last step could have been further improved, for example by using conditional probabilities (instead of independent positions in the matrix), or by re-building another matrix from the highest scoring occurrences, and so on. We save this for 'future improvements'.

And, basically, this is it.