Supplementary data for Identifying Discriminative Classification Based Motifs in Biological Sequences

Celine Vens[1,2], Marie-Noëlle Rosso,[2] and Etienne G.J. Danchin[2]

S1 Pseudocode for the MERCI algorithm

Table S1 provides pseudo-code for the MERCI algorithm.

S2 Running times of the MERCI algorithm

We evaluated the running times of MERCI as follows. First, we took a subset of 100 positive and 100 negative sequences, we set $k = 10$ and initialized F_P to 10 ($F_N = 0$), and we searched for motifs in the 50 first positions using the classification by Koolman and Rohm [1996]. We allowed one gap of maximal length 2. Running our tool with this set-up took 4.36 seconds (averaged over 5 runs) on an Intel Q9400 2.66GHz processor. Then, we subsequently changed one of the following settings:

- the number of positive sequences (100, 200, 500, 1000),
- the number of negative sequences (100, 200, 500, 1000),
- k (10, 20, 30, 40),
- the number of positions considered (20, 50, 100, full length),
- the classification scheme (none, Koolman and Rohm [1996], Betts and Russell [2003], Sayle and Milner-White [1995]),
- the number of gaps (no gaps, 1, 2, 3), and
- the maximal gap size (1, 2, 5, 10).

We plot the effect of each of these changes in Figure S1.

When changing the number of positive sequences, we kept F_P to 10% of the number of sequences. When the number of sequences exceeded the number of sequences in the original set, sequences were repeated. For the RasMol classification [Sayle and Milner-White, 1995], we used a maximal motif length of 8.

As we can see, the number of sequences and the number of motifs (k) have little influence on the running times. However, the number of positions considered has a larger influence on the running times. In general, about half of the running time is spent on checking the presence of the candidate

[1]Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan 200A, 3001 Leuven, Belgium
[2]Institut National de la Recherche Agronomique, U.M.R. - I.B.S.V. INRA-UNSA-CNRS, 400 route des Chappes, BP 167, 06903 Sophia-Antipolis Cedex, France

Table S1: High level pseudo code for the MERCI algorithm searching discriminative motifs with classes. Only the most important procedures are shown. The parameter k denotes the number of motifs to return, fp and fn are the F_P and F_N parameters described in the article, *dag* is the directed acyclic graph defined over the amino acids and their classification scheme.

procedure MERCI (*positive_sequences*,*negative_sequences*,*k*,*fp*,*fn*,*dag*)
// the input arguments are global variables
1: StoreFrequentPatternAndSequences(*empty_pattern*, *positive_sequences*)
2: *tree* = ConvertDagToSpanningTree(*dag*) // global variable
3: *new_candidates* = GenerateCandidates(*empty_pattern*, *tree*)
4: *valid_motifs* = ∅ // global variable
5: **for each** *candidate* ∈ *new_candidates*
6: Mine(*candidate*,*false*)
7: *top_k_motifs* = SelectTopK(*valid_motifs*, *fp*)
8: **return** *top_k_motifs*

procedure Mine (*candidate*,*parent_infrequent_in_negatives*)
1: **if** ParentsFrequent(*candidate*)
2: *sequences_to_check* = IntersectParentSequences(*candidate*)
3: *sequences_containing_candidate* = CheckOccurrence(*candidate*, *sequences_to_check*)
4: **if** |*sequences_containing_candidate*| ≥ *fp*
5: StoreFrequentPatternAndSequences(*candidate*, *sequences_containing_candidate*)
6: **if** *parent_infrequent_in_negatives* or InfrequentInNeg(*candidate*, *negative_sequences*, *fn*)
7: OutputMotif(*candidate*)
8: *valid_motifs* = *valid_motifs* ∪ (*candidate*, |*sequences_containing_candidate*|)
9: *fp* = PossiblyIncreaseThreshold(*validmotifs*, *k*)
10: *infrequent_in_negatives* = *true*
11: **else** *infrequent_in_negatives* = *false*
12: *new_candidates* = GenerateCandidates(*candidate*, *tree*)
13: **for each** *candidate* ∈ *new_candidates*
14: Mine(*candidate*,*infrequent_in_negatives*)

procedure GenerateCandidates (*parent_candidate*,*tree*)
1: *extensions* = AddFirstLevelNodesToEnd(*parent_candidate*, *tree*)
2: *specializations* = ReplaceLastElementWithChildNodes(*parent_candidate*, *tree*)
3: **return** *extensions* ∪ *specializations*

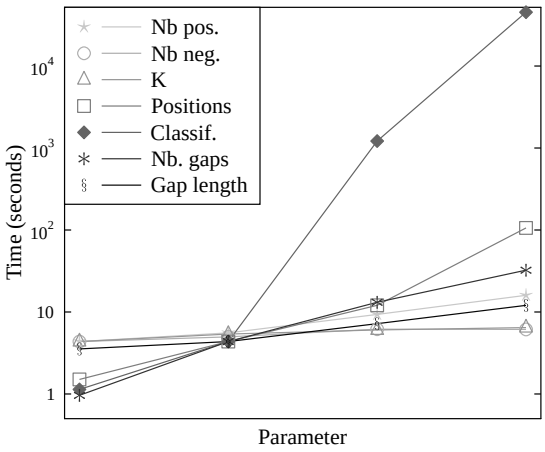

Figure S1: Running times for our motif discovery program.

motifs in the pruned positive sequences. Trying to decrease this time by storing not only sequence ids, but also start positions for the parent patterns did not result in an improvement.

The classification scheme has the largest impact on the running times. Especially the RasMol classification [Sayle and Milner-White, 1995], which has large classes, results in very large sets of candidates that are frequent in the positive set and need to be processed. When constructing a spanning tree for the RasMol classification scheme, the root has 8 subtrees, meaning that in fact, we could run 8 tasks in parallel on multi-processor machines. This is supported by MERCI, and explained in the corresponding manual.

S3 Evaluating classifications based on physico-chemical properties

As shown in the article, the use of degenerate positions results in motifs with a higher frequency. In our work, a degenerate position corresponds to a group of amino acids sharing a physico-chemical property. One may wonder if any other predefined grouping of amino acids would yield similar results. In order to answer this question, we performed the following experiment. We constructed a random classification scheme, with the same structure as the Koolman and Rohm [1996] scheme, and searched for all motifs that are absent from the negative set, and have a frequency higher than 8 (which is the highest frequency obtained when no classification scheme is used) in the positive set, allowing one gap of maximal length 5. With the true Koolman and Rohm [1996] classification, we obtained 51 motifs, while with the randomized classifications, we obtained - averaged over ten randomizations - between 0 and 19 motifs, with an average of 6.9 motifs. This result shows that it is indeed useful to consider a classification of amino acids based on physico-chemical properties.

References

M.J. Betts and R.B. Russell. Amino acid properties and consequences of subsitutions. In *Bioinformatics for Geneticists*. Wiley, 2003.

J. Koolman and K.H. Rohm. *Colour Atlas of Biochemistry*. Thieme, Stuttgart, 1996.

R. Sayle and E.J. Milner-White. RasMol: Biomolecular graphics for all. *Trends in Biochemical Sciences*, 20(9):374, 1995.