funBIalign: a hierachical algorithm for functional motif discovery based on mean squared residue score - Supplementary Material

Jacopo Di Iorio*
Dept. of Statistics, Penn State University

Marzia A. Cremona
Dept. of Operations and Decision Systems, Université Laval
CHU de Québec, Université Laval Research Center

Francesca Chiaromonte
Dept. of Statistics, Penn State University
Institute of Economics and L'EMbeDS, Sant'Anna School of Advanced Studies

August 20, 2024

S1 Proof of Theorem 1

Proof. Given a functional motif Q composed of n_Q portions, the functional mean squared residue score is defined as

$$H(Q) = \frac{1}{n_Q}\frac{1}{l}\sum_{k=1}^{n_Q}\int_0^l \left(p_k(t) - \overline{p}_k - \overline{p}(t) + \overline{p}\right)^2 dt,$$

where we have

$$\overline{p}_k = \frac{1}{l}\int_0^l p_k(t)dt,$$

$$\overline{p}(t) = \frac{1}{n_Q}\sum_{k=1}^{n_Q} p_k(t),$$

$$\overline{p} = \frac{1}{n_Q}\frac{1}{l}\sum_{k=1}^{n_Q}\int_0^l p_k(t)dt.$$

*Corresponding author. E-mail: jqd5830@psu.edu

We define $g_k(t) := p_k(t) - \overline{p}_k - \overline{p}(t) + \overline{p}$, so that $H(Q) = \frac{1}{n_Q}\frac{1}{l}\sum_{k=1}^{n_Q}\int_0^l g_k^2(t)dt$. We have that:

$$\begin{aligned}
g_k(t) &= p_k(t) - \overline{p}_k - \frac{1}{n_Q}\sum_{j=1}^{n_Q} p_j(t) + \frac{1}{n_Q}\sum_{j=1}^{n_Q}\overline{p}_j \\
&= \frac{n_Q - 1}{n_Q}p_k(t) - \frac{n_Q - 1}{n_Q}\overline{p}_k - \frac{1}{n_Q}\sum_{j\neq k} p_j(t) + \frac{1}{n_Q}\sum_{j\neq k}\overline{p}_j \\
&= A + B + C + D.
\end{aligned}$$

Then $g_k^2(t)$ is:

$$g_k^2(t) = A^2 + B^2 + C^2 + D^2 + 2AB + 2AC + 2AD + 2BC + 2BD + 2CD$$

where

$$A^2 = \frac{(n_Q - 1)^2}{n_Q^2}p_k^2(t),$$

$$B^2 = \frac{(n_Q - 1)^2}{n_Q^2}\overline{p}_k^2,$$

$$C^2 = \frac{1}{n_Q^2}\sum_{j\neq k} p_j^2(t) + \frac{1}{n_Q^2}\sum_{j\neq k}\left(p_j(t)\sum_{i\neq\{j,k\}} p_i(t)\right),$$

$$D^2 = \frac{1}{n_Q^2}\sum_{j\neq k}\overline{p}_j^2 + \frac{1}{n_Q^2}\sum_{j\neq k}\left(\overline{p}_j\sum_{i\neq\{j,k\}}\overline{p_i}\right),$$

$$2AB = -2\frac{(n_Q - 1)^2}{n_Q^2}p_k(t)\overline{p}_k,$$

$$2AC = -2\frac{n_Q - 1}{n_Q^2}p_k(t)\sum_{j\neq k} p_j(t),$$

$$2AD = 2\frac{n_Q - 1}{n_Q^2}p_k(t)\sum_{j\neq k}\overline{p}_j,$$

$$2BC = 2\frac{n_Q - 1}{n_Q^2}\overline{p}_k\sum_{j\neq k} p_j(t),$$

$$2BD = -2\frac{n_Q - 1}{n_Q^2}\overline{p}_k\sum_{j\neq k}\overline{p}_j,$$

$$2CD = -2\frac{1}{n_Q^2}\left(\sum_{j\neq k} p_j(t)\overline{p}_j\right).$$

Defining $z_k(t) := p_k^2(t)$, we can rewrite A^2 and C^2 in the following way:

$$A^2 = \frac{(n_Q-1)^2}{n_Q^2} z_k(t),$$

$$C^2 = \frac{1}{n_Q^2}\sum_{j\neq k} z_j(t) + \frac{1}{n_Q^2}\sum_{j\neq k}\left(p_j(t)\sum_{i\neq\{j,k\}} p_i(t)\right).$$

Noting that $2\frac{1}{l}\int_0^l BCdt + 2\frac{1}{l}\int_0^l BDdt = 0$, we can then compute:

$$\begin{aligned}
\frac{1}{l}\int_0^l g_k^2(t)dt =& \frac{1}{l}\int_0^l A^2dt + \frac{1}{l}\int_0^l B^2dt \\
&+ \frac{1}{l}\int_0^l C^2dt + \frac{1}{l}\int_0^l D^2dt + \\
&+ 2\frac{1}{l}\int_0^l ABdt + 2\frac{1}{l}\int_0^l ACdt + 2\frac{1}{l}\int_0^l ADdt + 2\frac{1}{l}\int_0^l CDdt \\
=& \frac{(n_Q-1)^2}{n_Q^2}\overline{z}_k + \frac{(n_Q-1)^2}{n_Q^2}\overline{p}_k^2 \\
&+ \frac{1}{n_Q^2}\sum_{j\neq k}\overline{z}_j + \frac{1}{n_Q^2}\frac{1}{l}\int_0^l\sum_{j\neq k}\left(p_j(t)\sum_{i\neq\{j,k\}}p_i(t)\right)dt + \frac{1}{n_Q^2}\sum_{j\neq k}\overline{p}_j^2 + \frac{1}{n_Q^2}\sum_{j\neq k}\left(\overline{p}_j\sum_{i\neq\{j,k\}}\overline{p}_i\right) + \\
&- 2\frac{(n_Q-1)^2}{n_Q^2}\overline{p}_k^2 - 2\frac{n_Q-1}{n_Q^2}\frac{1}{l}\int_0^l\left(p_k(t)\sum_{j\neq k}p_j(t)\right)dt + 2\frac{n_Q-1}{n_Q^2}\overline{p}_k\sum_{j\neq k}\overline{p}_j - 2\frac{1}{n_Q^2}\left(\sum_{j\neq k}\overline{p}_j\right)^2.
\end{aligned}$$

We now compute the coefficients corresponding to each of the elements $\overline{z}_\kappa$, $\overline{p}_\kappa^2$, $\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt$, and $\overline{p}_\kappa\overline{p}_\iota$, with $\iota\neq\kappa$, in the H-score of a motif Q composed of n_Q portions, multiplied by n_Q:

$$\begin{aligned}
n_Q H(Q) &= \sum_{k=1}^{n_Q}\frac{1}{l}\int_0^l g_k^2(t)dt \\
&= \sum_{\kappa=1}^{n_Q}\left[h^Q_{\overline{z}_\kappa}\overline{z}_\kappa + h^Q_{\overline{p}_\kappa^2}\overline{p}_\kappa^2 + \sum_{\iota\neq\kappa}\left(h^Q_{\overline{p}_{\iota\kappa}}\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt + h^Q_{\overline{p}_\kappa\overline{p}_\iota}\overline{p}_\kappa\overline{p}_\iota\right)\right].
\end{aligned}$$

We start computing $h^Q_{\overline{z}_\kappa}$, the coefficient referring to $\overline{z}_\kappa$. This element appears twice: in $\frac{(n_Q-1)^2}{n_Q^2}\overline{z}_k$ when $k=\kappa$, and in $\frac{1}{n_Q^2}\sum_{j\neq k}\overline{z}_j$ when $k\neq\kappa$ (that is, n_Q-1 times). Hence, we have

$$h^Q_{\overline{z}_\kappa} = \frac{(n_Q-1)^2}{n_Q^2} + \frac{n_Q-1}{n_Q^2} = \frac{n_Q-1}{n_Q}.$$

To compute the coefficient $h^Q_{\overline{p}_\kappa^2}$ referring to the element $\overline{p}_\kappa^2$, we need to consider that the element appears four times: in $\frac{(n_Q-1)^2}{n_Q^2}\overline{p}_k^2$ and in $-2\frac{(n_Q-1)^2}{n_Q^2}\overline{p}_k^2$ when $k=\kappa$; in $\frac{1}{n_Q^2}\sum_{j\neq k}\overline{p}_j^2$ and in

$-2\frac{1}{n_Q^2}\left(\sum_{j\neq k}\overline{p}_j\right)^2$ when $k \neq \kappa$ (that is, $n_Q - 1$ times). Hence, we have

$$h^Q_{\overline{p}^2_\kappa} = \frac{(n_Q-1)^2}{n_Q^2} - 2\frac{(n_Q-1)^2}{n_Q^2} + \frac{n_Q-1}{n_Q^2} - 2\frac{n_Q-1}{n_Q^2} = -\frac{n_Q-1}{n_Q}.$$

For the coefficient of the element $\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt$ for $\iota \neq \kappa$, i.e. $h^Q_{\overline{p}_{\iota\kappa}}$, we need to consider the term $-2\frac{n_Q-1}{n_Q^2}\frac{1}{l}\int_0^l\left(p_k(t)\sum_{j\neq k}p_j(t)\right)dt$ when $k = \kappa$ or $k = \iota$ (that is, 2 times), and the term $\frac{1}{n_Q^2}\frac{1}{l}\int_0^l\sum_{j\neq k}\left(p_j(t)\sum_{i\neq\{j,k\}}p_i(t)\right)dt$ when $k \neq \{\kappa,\iota\}$ (that is, $2(n_Q-2)$ times). We obtain:

$$h^Q_{\overline{p}_{\iota\kappa}} = -2\left(-2\frac{n_Q-1}{n_Q^2}\right) + 2\frac{n_Q-2}{n_Q^2} = -\frac{2}{n_Q}.$$

Finally, we compute $h^Q_{\overline{p}_\kappa\overline{p}_\iota}$ for $\iota \neq \kappa$, the coefficient corresponding to $\overline{p}_\kappa\overline{p}_\iota$. This element appears in the term $2\frac{n_Q-1}{n_Q^2}\overline{p}_k\sum_{j\neq k}\overline{p}_j$ when $k \neq \kappa$ or $k \neq \iota$ (that is, 2 times), in the term $\frac{1}{n_Q^2}\sum_{j\neq k}\left(\overline{p}_j\sum_{i\neq\{j,k\}}\overline{p}_i\right)$ when $k \neq \{\kappa,\iota\}$ (that is, $2(n_Q-2)$ times), and in the term $-2\frac{1}{n_Q^2}\left(\sum_{j\neq k}\overline{p}_j\right)^2$ when $k \neq \{\kappa,\iota\}$ (that is, $2(n_Q-2)$ times). Hence, we have:

$$h^Q_{\overline{p}_\kappa\overline{p}_\iota} = 2\left(2\frac{n_Q-1}{n_Q^2}\right) + 2\frac{n_Q-2}{n_Q^2} + 2(n_Q-2)\left(-2\frac{1}{n_Q^2}\right) = \frac{2}{n_Q}.$$

We now consider $\overline{H}_n$ for $n = 2, 3, \ldots, n_Q$, the average fMRS of all $\binom{n_Q}{n}$ sub-motifs of Q obtained selecting exactly n of the n_Q portions $p_k(t)$ belonging to Q:

$$\begin{aligned}\overline{H}_n &= \binom{n_Q}{n}^{-1}\sum_{\substack{R\subseteq Q\\|R|=n}} H(R)\\ &= \sum_{\kappa=1}^{n_Q}\left[\overline{h}^n_{\overline{z}_\kappa}\overline{z}_\kappa + \overline{h}^n_{\overline{p}^2_\kappa}\overline{p}^2_\kappa + \sum_{\iota\neq\kappa}\left(\overline{h}^n_{\overline{p}_{\iota\kappa}}\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt + \overline{h}^n_{\overline{p}_\kappa\overline{p}_\iota}\overline{p}_\kappa\overline{p}_\iota\right)\right].\end{aligned}$$

We start computing $\overline{h}^n_{\overline{z}_\kappa}$, the coefficient corresponding to $\overline{z}_\kappa$. This element appears in the $H(R)$ of the sub-motifs comprising the portion $p_\kappa(t)$, that are exactly $\binom{n_Q-1}{n-1}$. Hence, we have:

$$\overline{h}^n_{\overline{z}_\kappa} = \frac{\binom{n_Q-1}{n-1}}{\binom{n_Q}{n}}\frac{h^R_{\overline{z}_\kappa}}{n} = \frac{n-1}{nn_Q}.$$

To compute $\overline{h}^n_{\overline{p}^2_\kappa}$, the coefficient of the element $\overline{p}^2_\kappa$, we need to consider the sub-motifs comprising the portion $p_\kappa(t)$. There are exactly $\binom{n_Q-1}{n-1}$ such motifs, so we obtain:

$$\overline{h}^n_{\overline{p}^2_\kappa} = \frac{\binom{n_Q-1}{n-1}}{\binom{n_Q}{n}}\frac{h^R_{\overline{p}^2_\kappa}}{n} = -\frac{n-1}{nn_Q}$$

To compute the coefficient $\overline{h}^n_{\overline{p}_{\iota\kappa}}$ of the element $\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt$ with $\iota \neq \kappa$ we need to consider the $\binom{n_Q-2}{n-2}$ sub-motifs comprising both portions $p_\kappa(t)$ and $p_\iota(t)$, obtaining:

$$\overline{h}^n_{\overline{p}_{\iota\kappa}} = \frac{\binom{n_Q-2}{n-2}}{\binom{n_Q}{n}} \frac{h^R_{\overline{p}_{\iota\kappa}}}{n} = -2\frac{n-1}{n_Q(n_Q-1)n}.$$

Finally, for the coefficient $\overline{h}^n_{\overline{p}_\kappa\overline{p}_\iota}$ corresponding to the element $\overline{p}_\kappa\overline{p}_\iota$ with $\iota \neq \kappa$, we need to consider the $\binom{n_Q-2}{n-2}$ sub-motifs comprising both portions $p_\kappa(t)$ and $p_\iota(t)$. Hence, we have:

$$\overline{h}^n_{\overline{p}_\kappa\overline{p}_\iota} = \frac{\binom{n_Q-2}{n-2}}{\binom{n_Q}{n}} \frac{h^R_{\overline{p}_\kappa\overline{p}_\iota}}{n} = 2\frac{n-1}{n_Q(n_Q-1)n}.$$

Putting all the terms together, we obtain:

$$\overline{H}_n = \frac{n-1}{nn_Q}\sum_{\kappa=1}^{n_Q}\left[\overline{z}_\kappa + \overline{p}_\kappa^2 - \frac{2}{n_Q-1}\sum_{\iota\neq\kappa}\left(\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt - \overline{p}_\kappa\overline{p}_\iota\right)\right].$$

We are interested in the relationship between $\overline{H}_{n+1}$ and $\overline{H}_n$. We have:

$$\begin{aligned}\overline{H}_{n+1} &= \frac{n}{(n+1)n_Q}\sum_{\kappa=1}^{n_Q}\left[\overline{z}_\kappa + \overline{p}_\kappa^2 - \frac{2}{n_Q-1}\sum_{\iota\neq\kappa}\left(\frac{1}{l}\int_0^l p_\kappa(t)p_\iota(t)dt - \overline{p}_\kappa\overline{p}_\iota\right)\right] \\ &= \frac{n^2}{(n+1)(n-1)}\overline{H}_n,\end{aligned}$$

from which we obtain

$$\overline{H}_{n+1} = \overline{H}_n\frac{n^2}{n^2-1}.$$

□

S2 Comparison of fMSR and adjusted fMSR

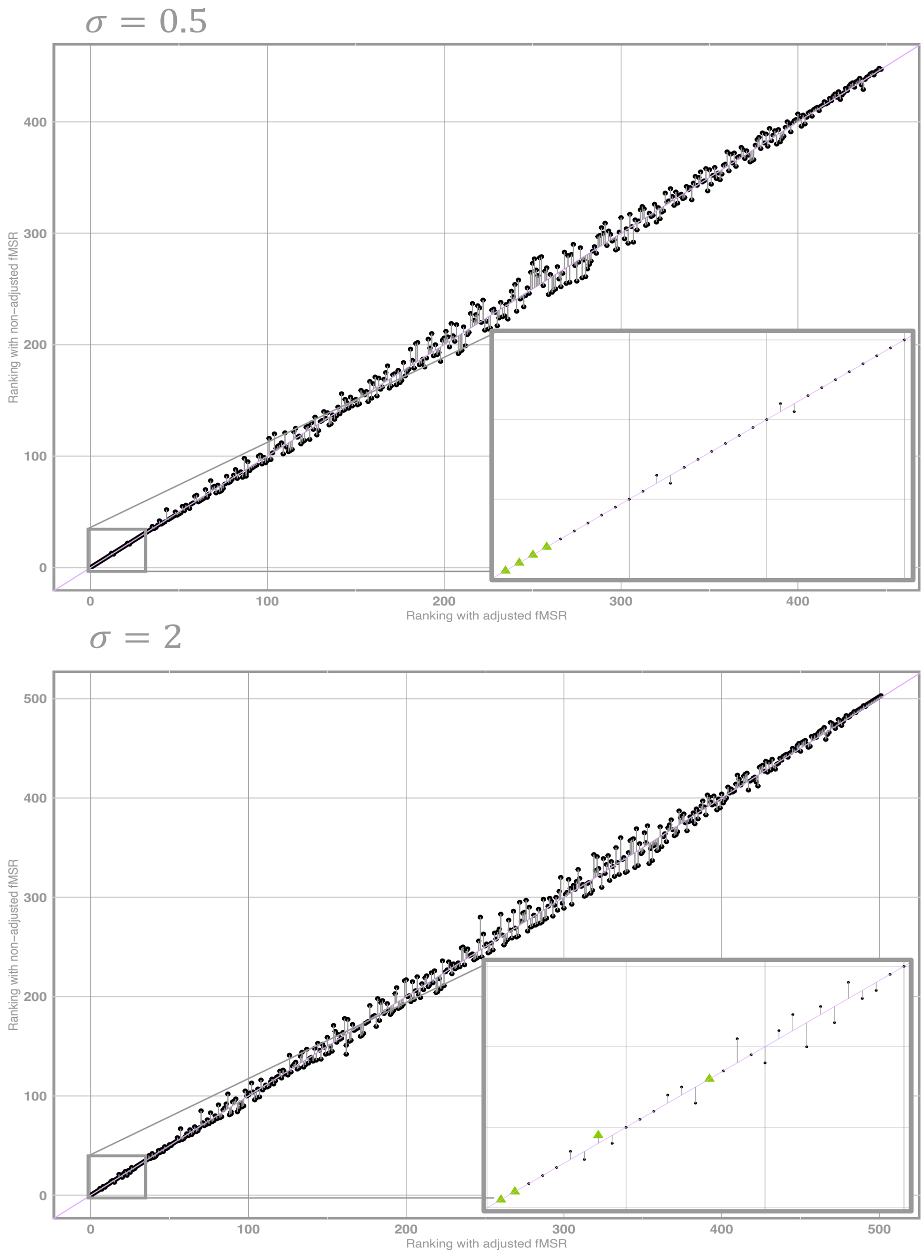


Figure S1: Comparison of the rankings obtained running funBIalign with adjusted fMSR and (non-adjusted) fMSR using the motif set n. 7 (with 8 occurrences) presented in the main text, with $\sigma = 0.5$ and $\sigma = 2$. The first 30 motifs are presented in a zoomed-in side panel where the embedded motifs are highlighted with a green triangle. In both noise regimes, and in particular when $\sigma = 2$, the rankings differ in their mid-ranges. However, the top-ranges of the rankings (the first 30 motifs in the zoomed-in panel) are fairly stable, presenting changes of at most 3 positions (for $\sigma = 2$). This is due to the fact that the embedded motifs have, by construction, an high number of occurrences and a level of noise much lower than the the motifs randomly generated – a scenario that is not always assured a priori.

S3 Selection of ℓ and n_{min}

S3.1 Selection of the motif length ℓ

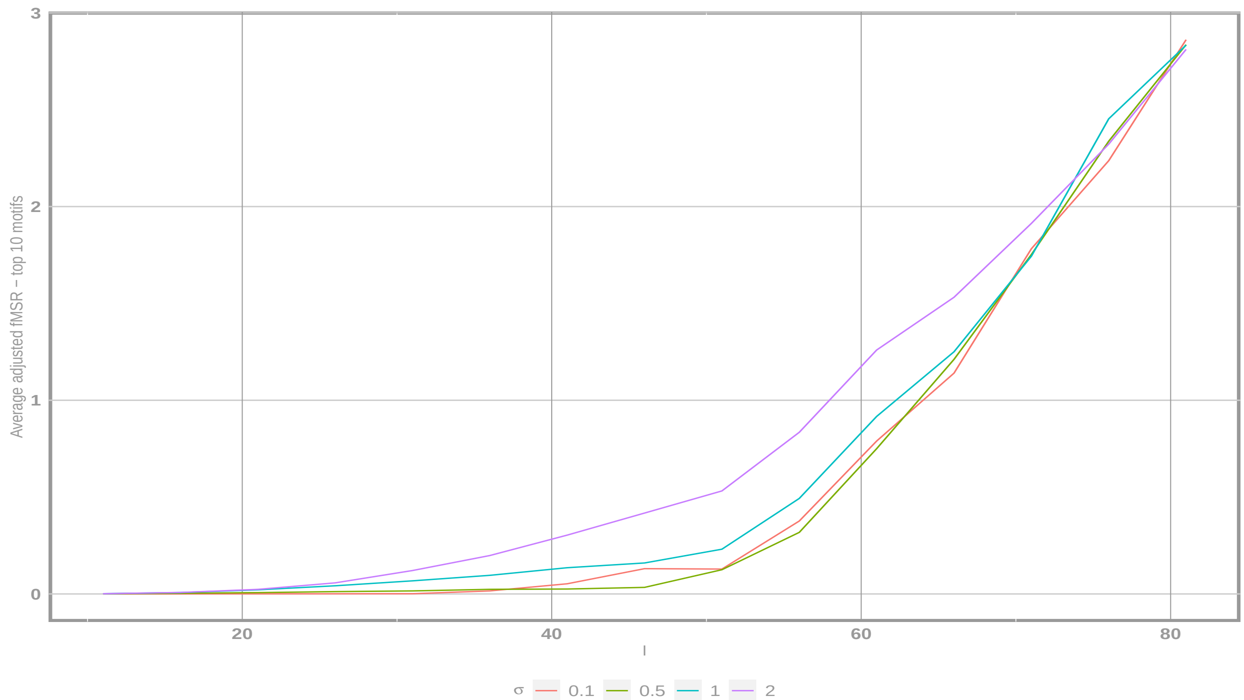


Figure S2: Average adjusted fMSR for the top 10 motifs identified in all the simulations ($\sigma = \{0.1, 0.5, 1, 2\}$) employing 8 instances of motif set n. 7 with $\ell = \{11, 16, \ldots, 76, 81\}$. It is evident that there is an elbow around $\ell = 51$. Simulated motifs have length 41, but due to the nature of the b-splines used in the simulation process, motifs naturally stretch beyond their assigned intervals. Thus, in practice, it makes sense for the length identified by the elbow approach to be slighty larger than 41.

S3.2 Impact of the choice of n_{min}

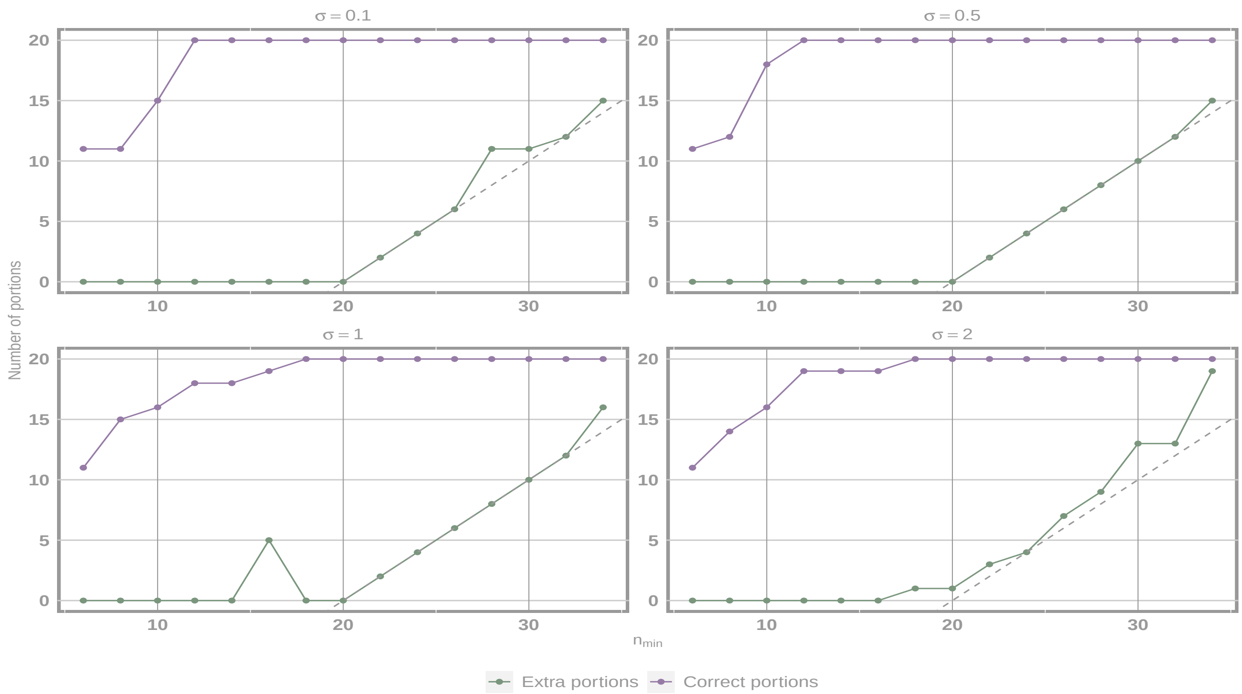


Figure S3: Number of (extra or correct) portions identified by *funBIalign* with $n_{min} = 6, 8, \ldots, 32$, and 34. The dashed grey line represents the minimum number of extra portions that the method is forced to identified given n_{min}. The 20 embedded portions of the motif are (almost) completely identified with $n_{min} \geq 12$. In general, the complete identification of the motif is harder with higher levels of noise.

S4 Motifs with shared noise level

S4.1 Motifs with 8 occurrences

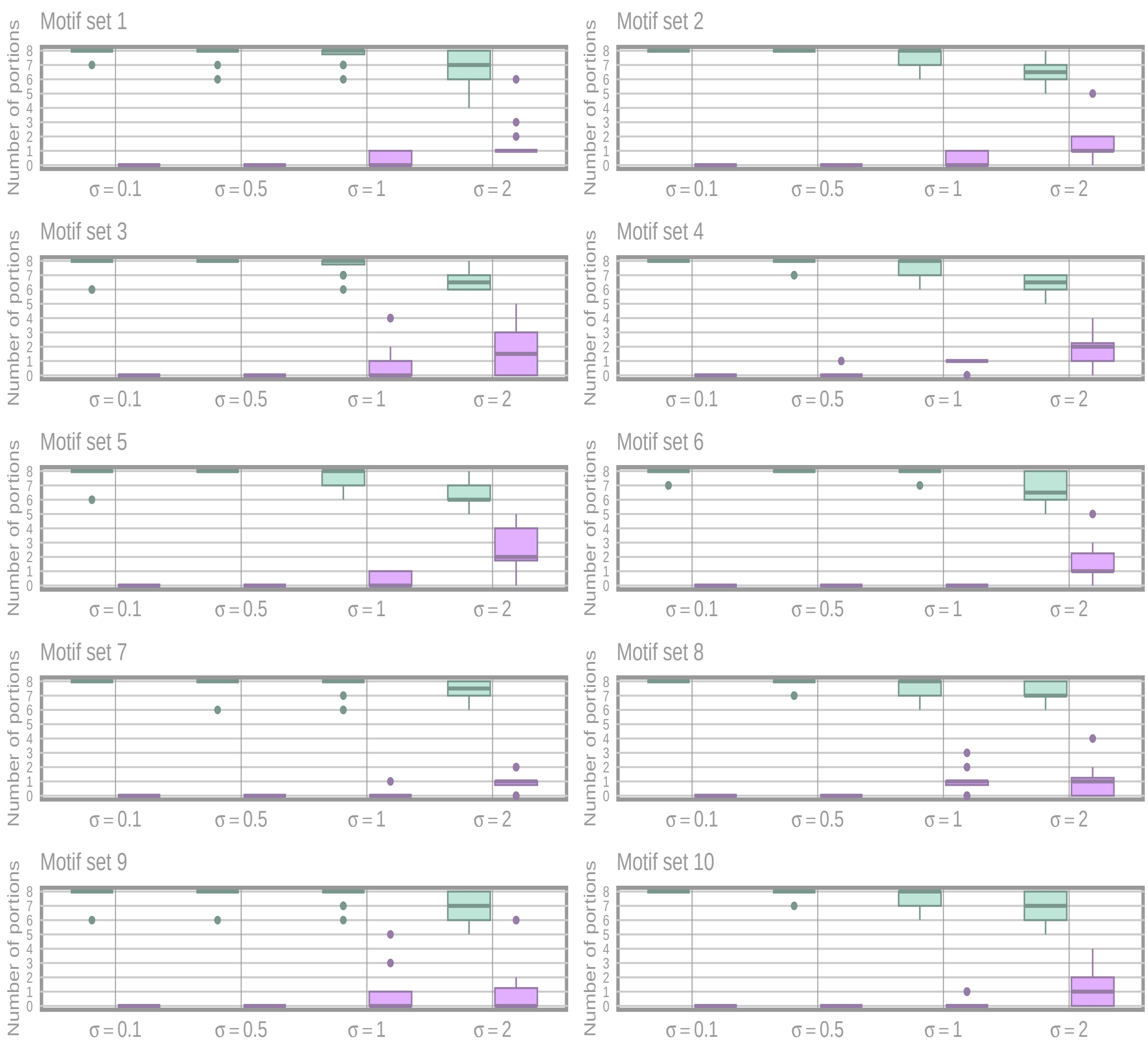


Figure S4: Performance of *funBIalign* for all simulations where motifs have 8 occurrences and shared noise level. Different panels show the performance pooling across runs of the algorithm with minimum cardinalities $n_{min} = 5, 6, 7, 8$, but separately for each of the 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

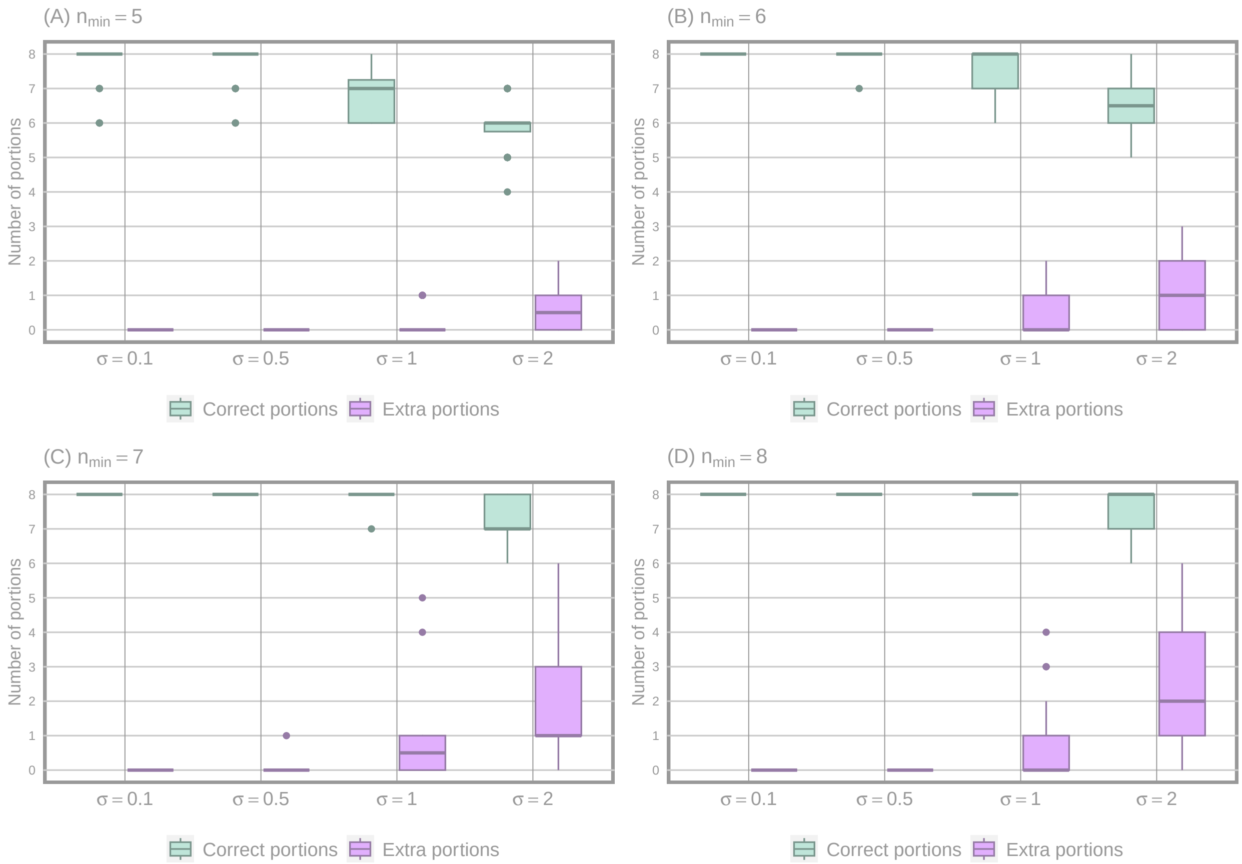


Figure S5: Performance of *funBIalign* for all simulations where motifs have 8 occurrences and shared noise level. The four panels show the performance of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

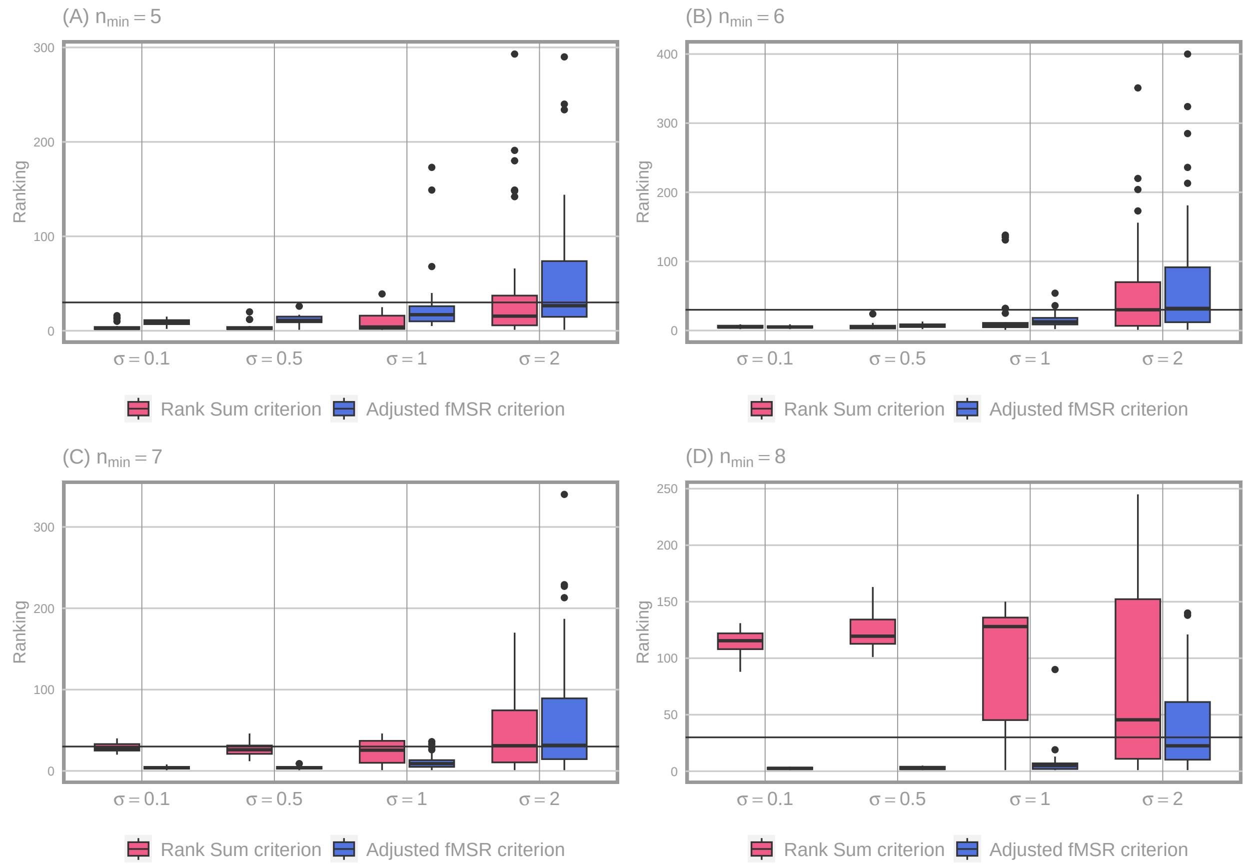


Figure S6: Ranking of the results of *funBIalign* which are most similar to the intentionally embedded motifs, for all simulations where motifs have 8 occurrences and shared noise level. The four panels show the ranking of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the red and blue boxplots (left and right) represent ranking according to the rank sum and the adjusted fMSR criterion, respectively. Horizontal line indicates rank 30.

S4.2 Motifs with 10 occurrences

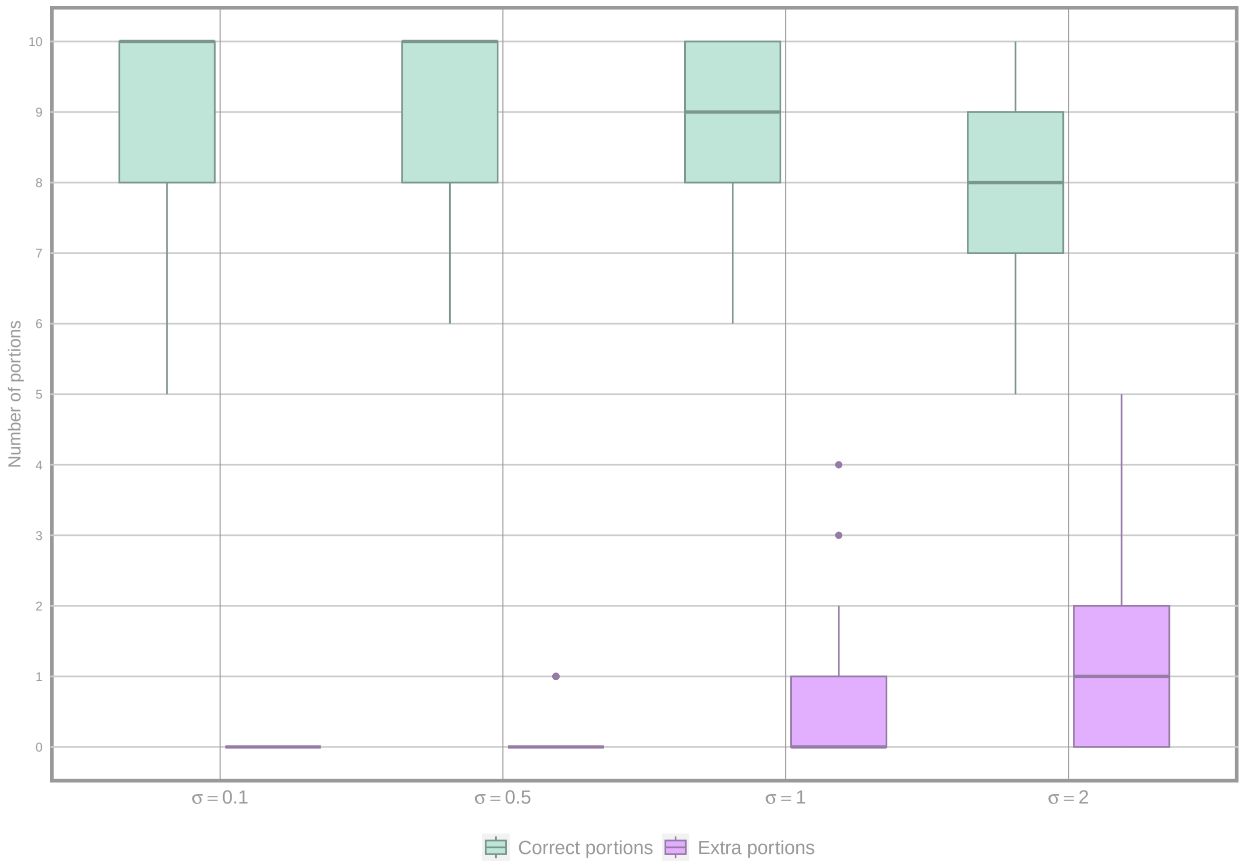


Figure S7: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and shared noise levels. The algorithm is run with minimum cardinalities $n_{min} = 5, 6, 7, 8$ and results are pooled. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively. See Figure 2.A.

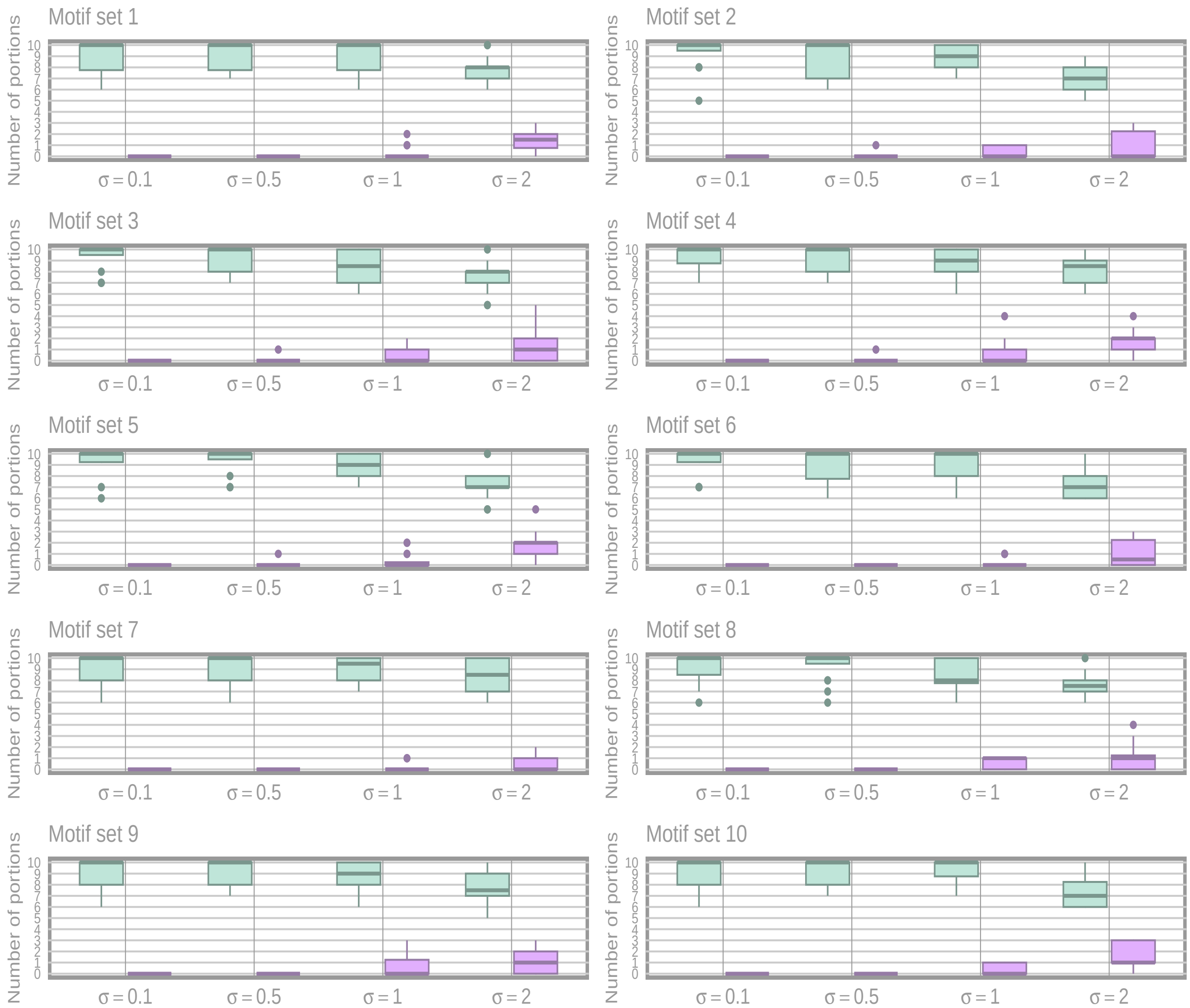


Figure S8: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and shared noise level. Different panels show the performance pooling across runs of the algorithm with minimum cardinalities $n_{min} = 5, 6, 7, 8$, but separately for each of the 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

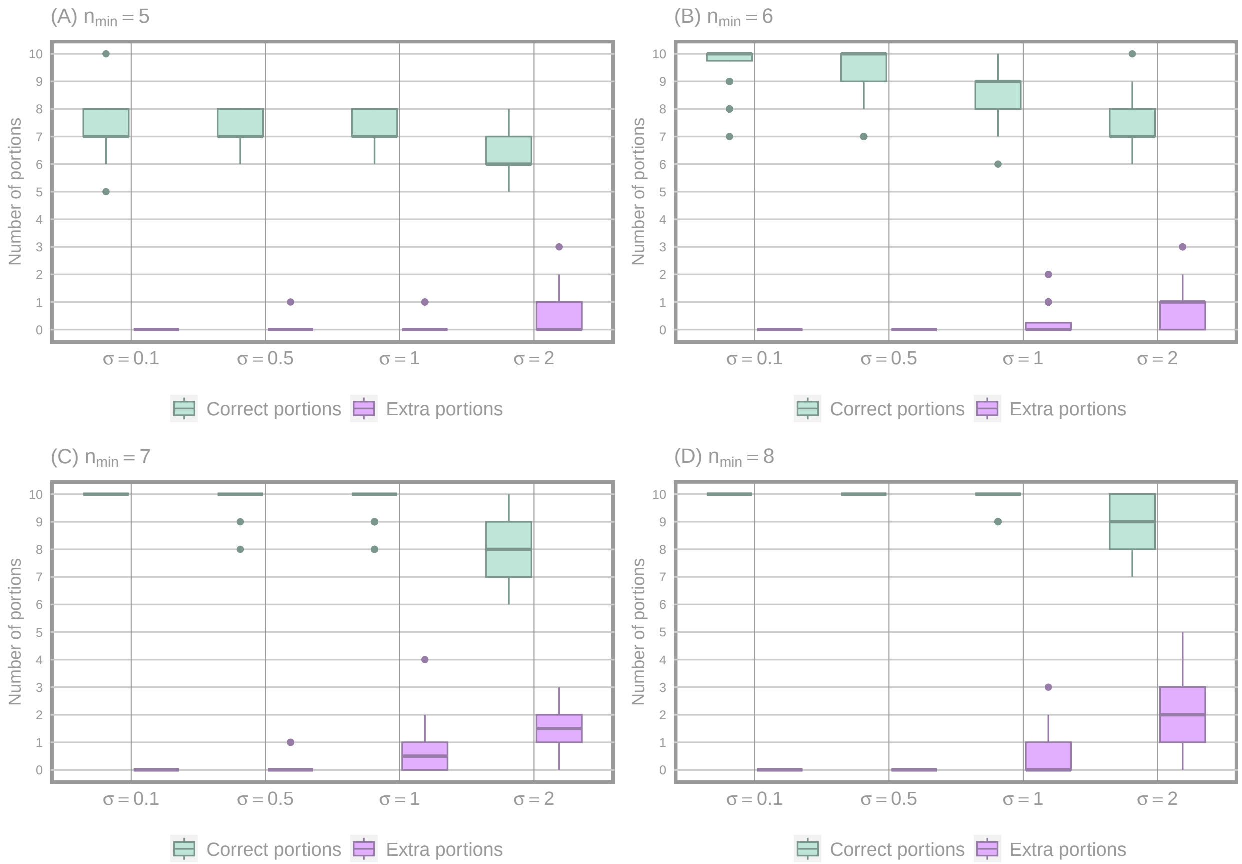


Figure S9: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and shared noise level. The four panels show the performance of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

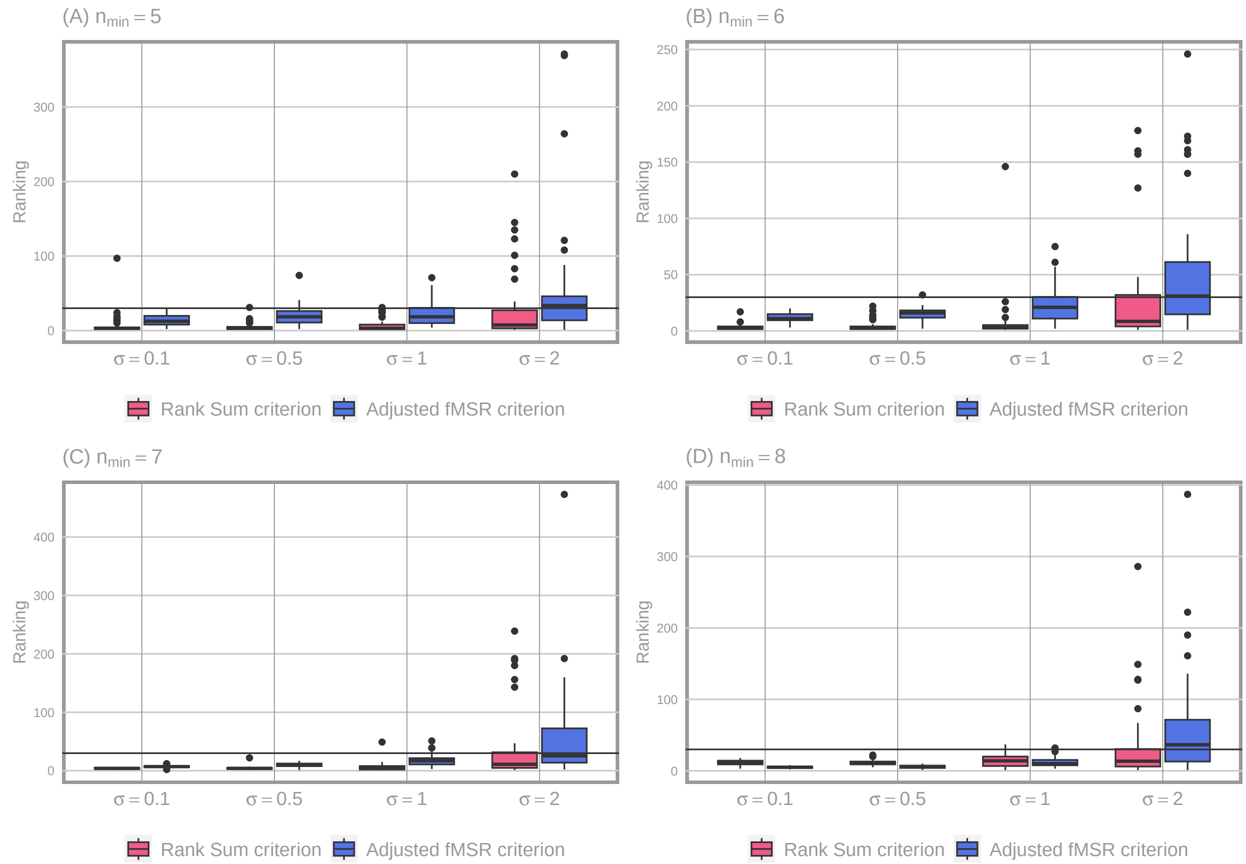


Figure S10: Ranking of the results of *funBIalign* which are most similar to the intentionally embedded motifs, for all simulations where motifs have 10 occurrences and shared noise level. The four panels show the ranking of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the red and blue boxplots (left and right) represent ranking according to the rank sum and the adjusted fMSR criterion, respectively. horizontal line indicates rank 30.

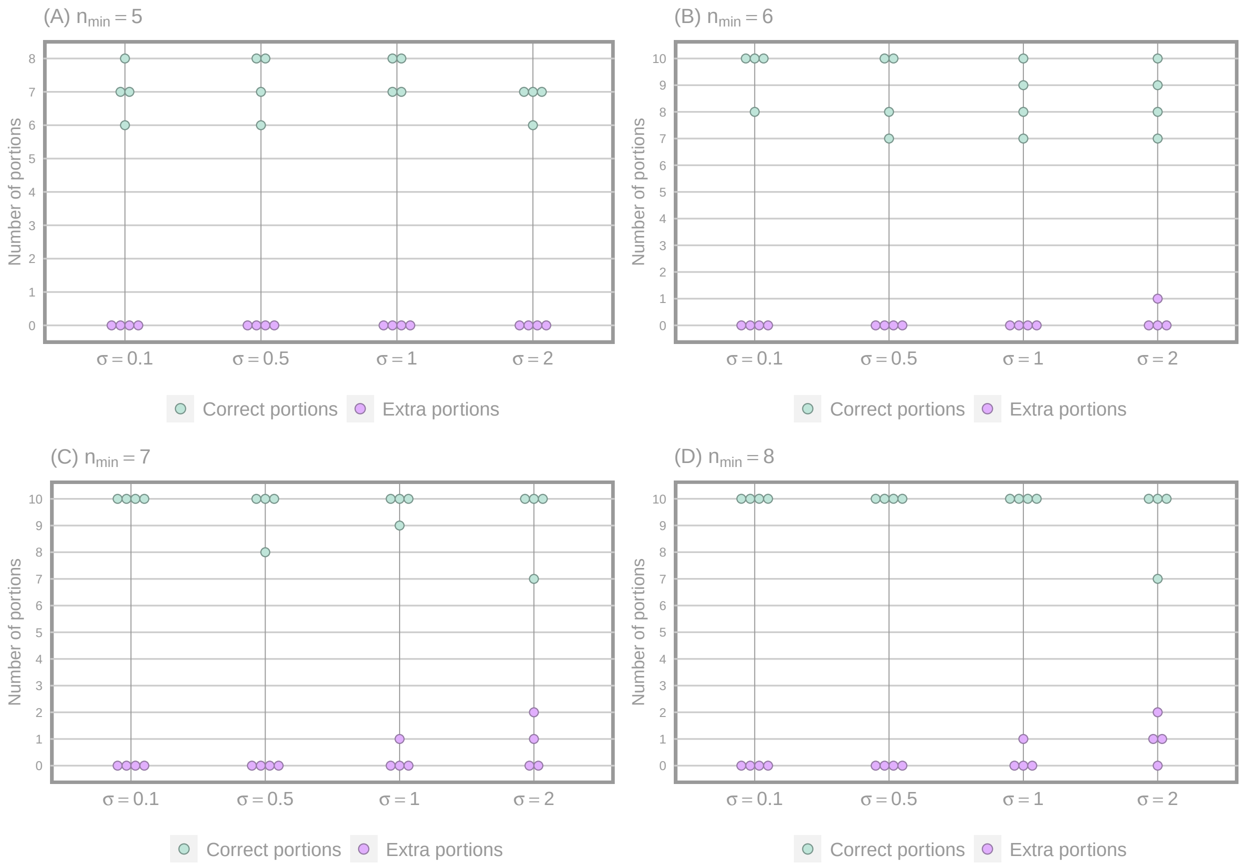
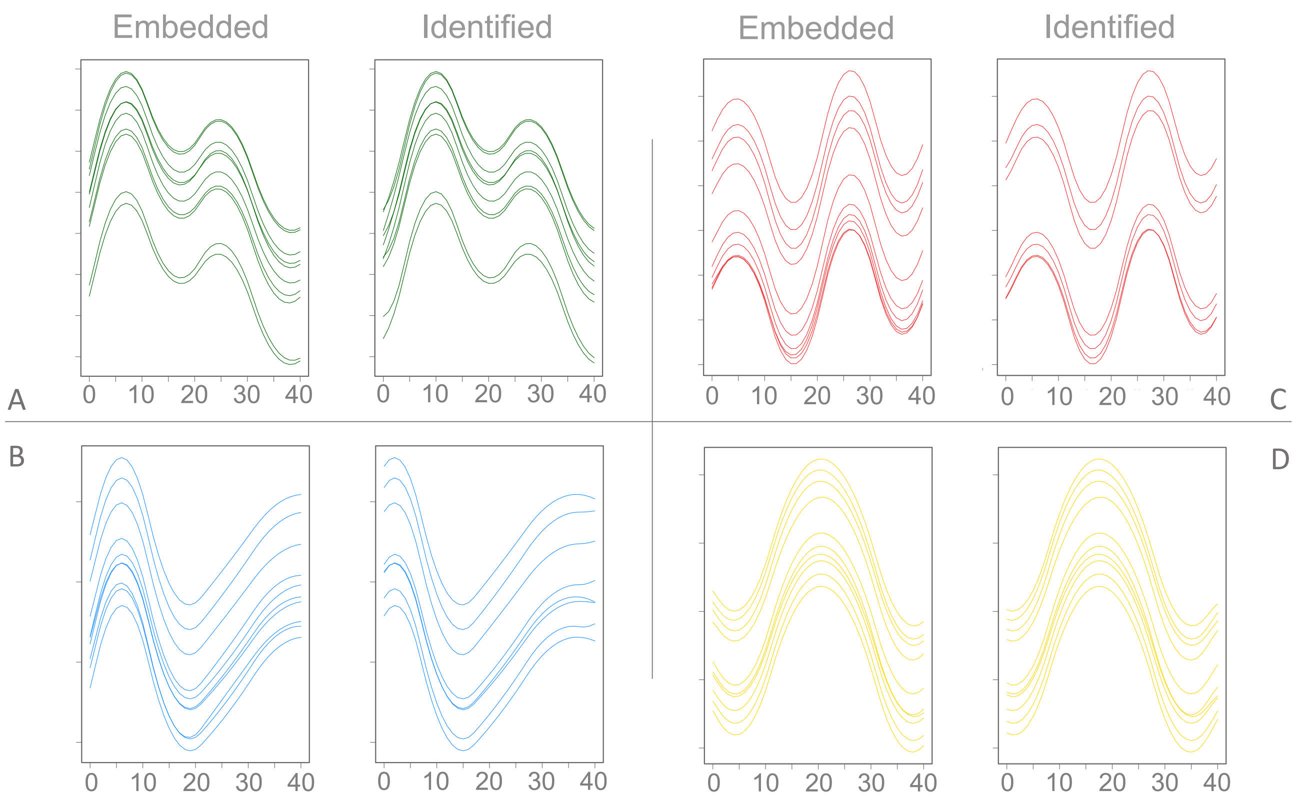
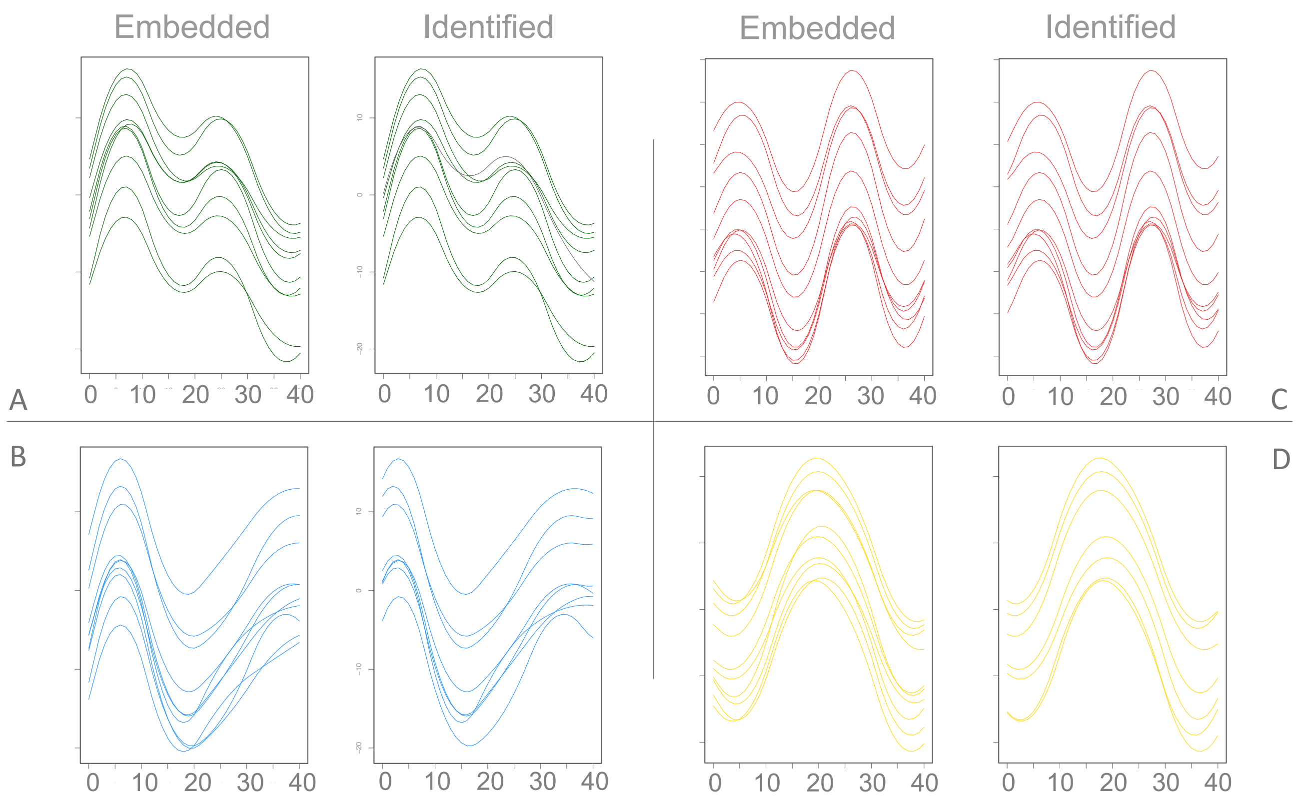


Figure S11: Performance of *funBIalign* for simulations employing motif set n. 7 (where motifs have 10 occurrences and shared noise level), shown separately for runs with varying n_{min}. Green and pink jittered dots represent correctly identified portions and extra portions, respectively.



	Adj fMSR ranking	Rank Sum ranking	Correct occurrences	Extra occurrences	Missing occurrences
A	23	2	10	0	0
B	11	5	8	0	2
C	2	14	7	0	3
D	21	1	10	0	0

Figure S12: Motif identification for the simulation employing motif set n. 7 and $\sigma = 0.5$, where motifs have 10 occurrences. *funBIalign* is run with $n_{min} = 6$. For each of the 4 motifs, color-coded in green, red, blue and yellow, left and right panels show the 10 occurrences embedded in the curve and the most similar portions identified by the algorithm, respectively. The table provides rankings and numbers of correctly identified, extra and missing occurrences for each motif. No extra occurrences are identified, but 5 embedded occurrences are missed.



	Adj fMSR ranking	Rank Sum ranking	Correct occurrences	Extra occurrences	Missing occurrences
A	54	4	9	1	1
B	37	14	8	0	2
C	43	1	10	0	0
D	4	32	7	0	3

Figure S13: Motif identification for the simulation employing motif set n. 7 and $\sigma = 2$, where motifs have 10 occurrences. *funBIalign* is run with $n_{min} = 6$. See legend for Figure S12. Extra occurrences are represented in gray among the identified portions panels.

S5 Motifs with different noise levels

S5.1 Motifs with 8 occurrences

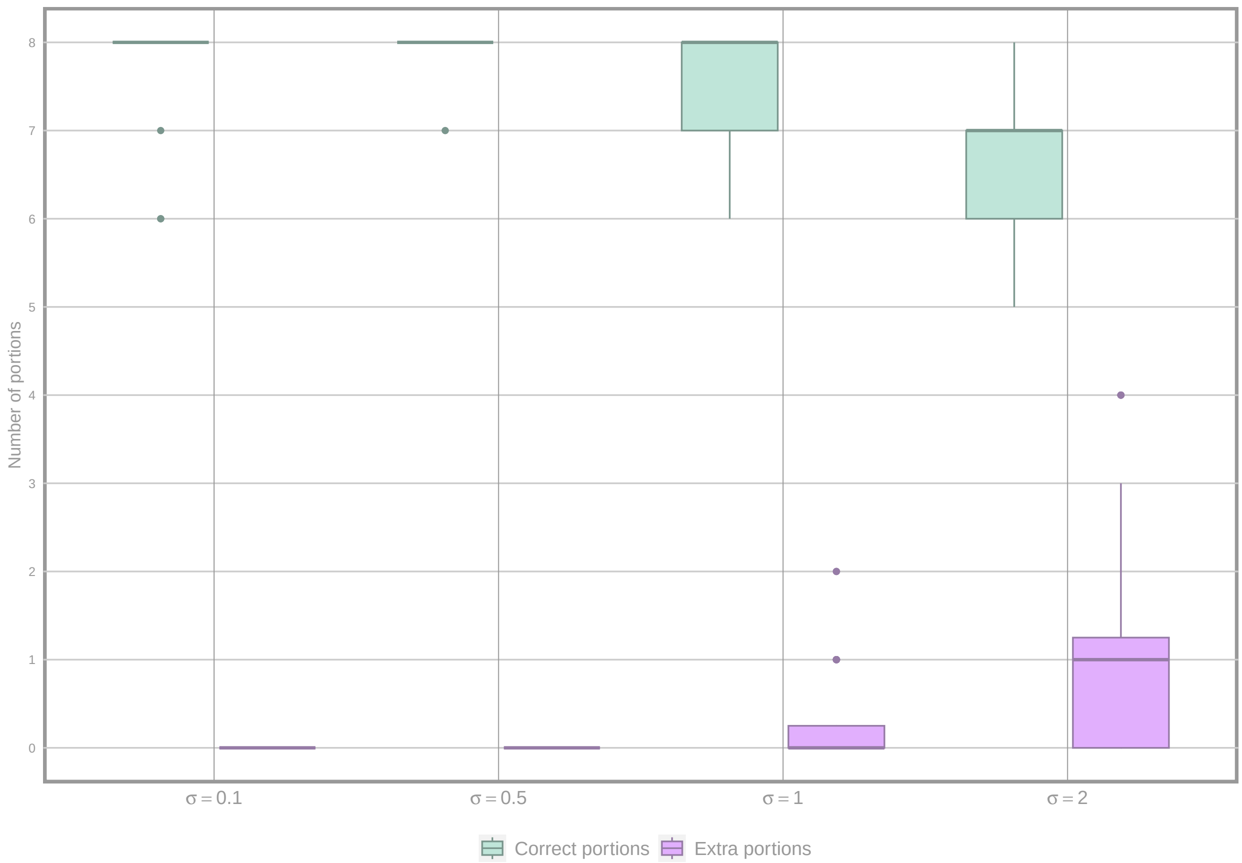


Figure S14: Performance of *funBIalign* for all simulations where motifs have 8 occurrences and different noise levels. The algorithm is run with minimum cardinalities $n_{min} = 5, 6, 7, 8$ and results are pooled. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively. See Figure 2.A.

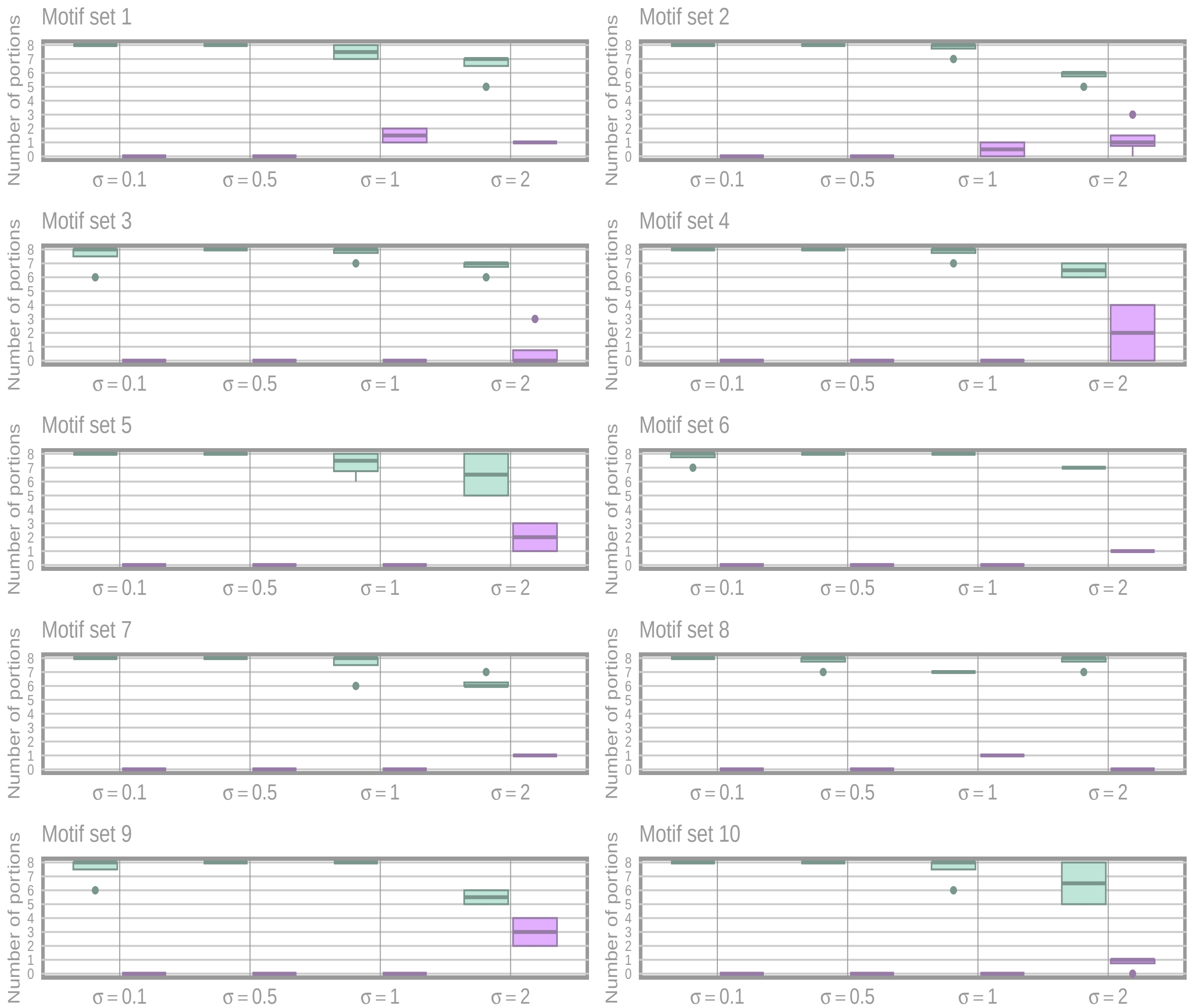


Figure S15: Performance of *funBIalign* for all simulations where motifs have 8 occurrences and different noise level. Different panels show the performance pooling across runs of the algorithm with minimum cardinalities $n_{min} = 5, 6, 7, 8$, but separately for each of the 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

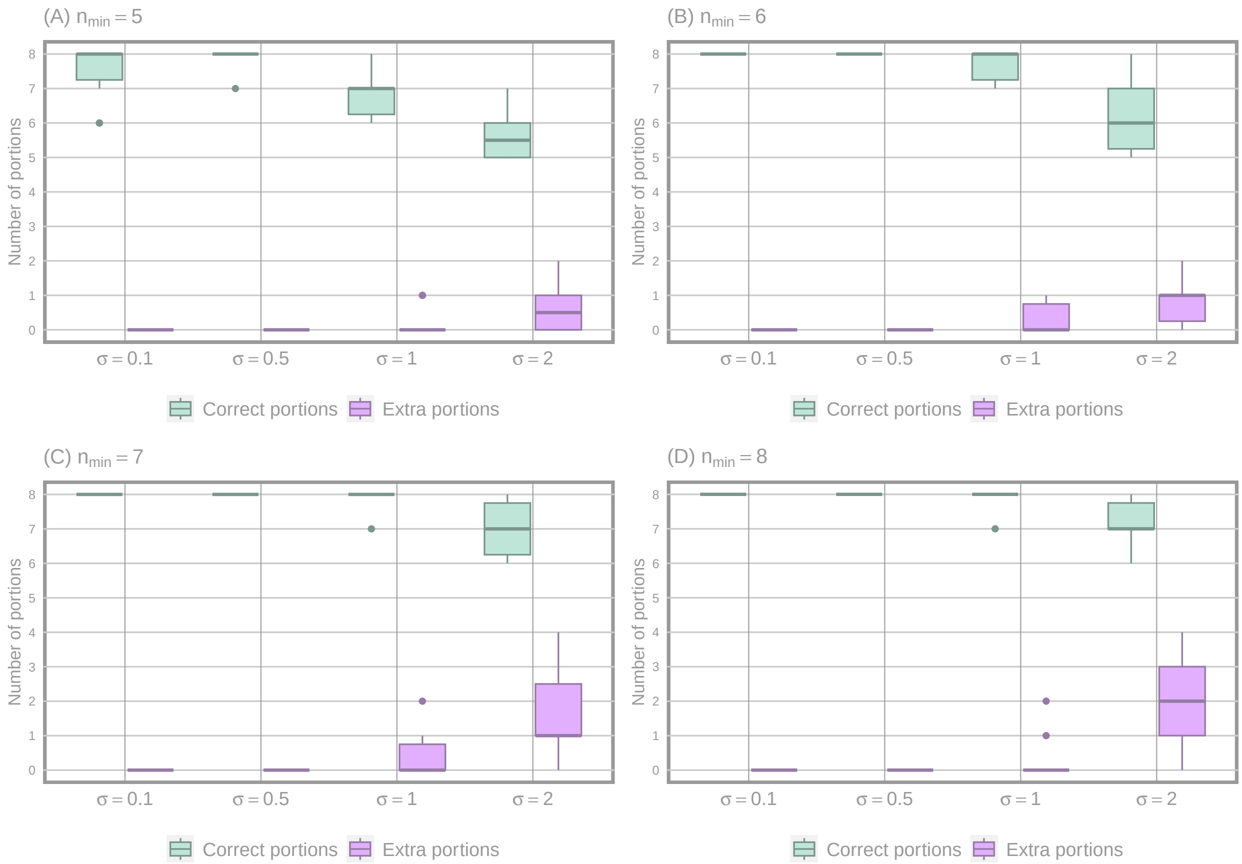


Figure S16: Performance of *funBIalign* for all simulations where motifs have 8 occurrences and different noise level. The four panels show the performance of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

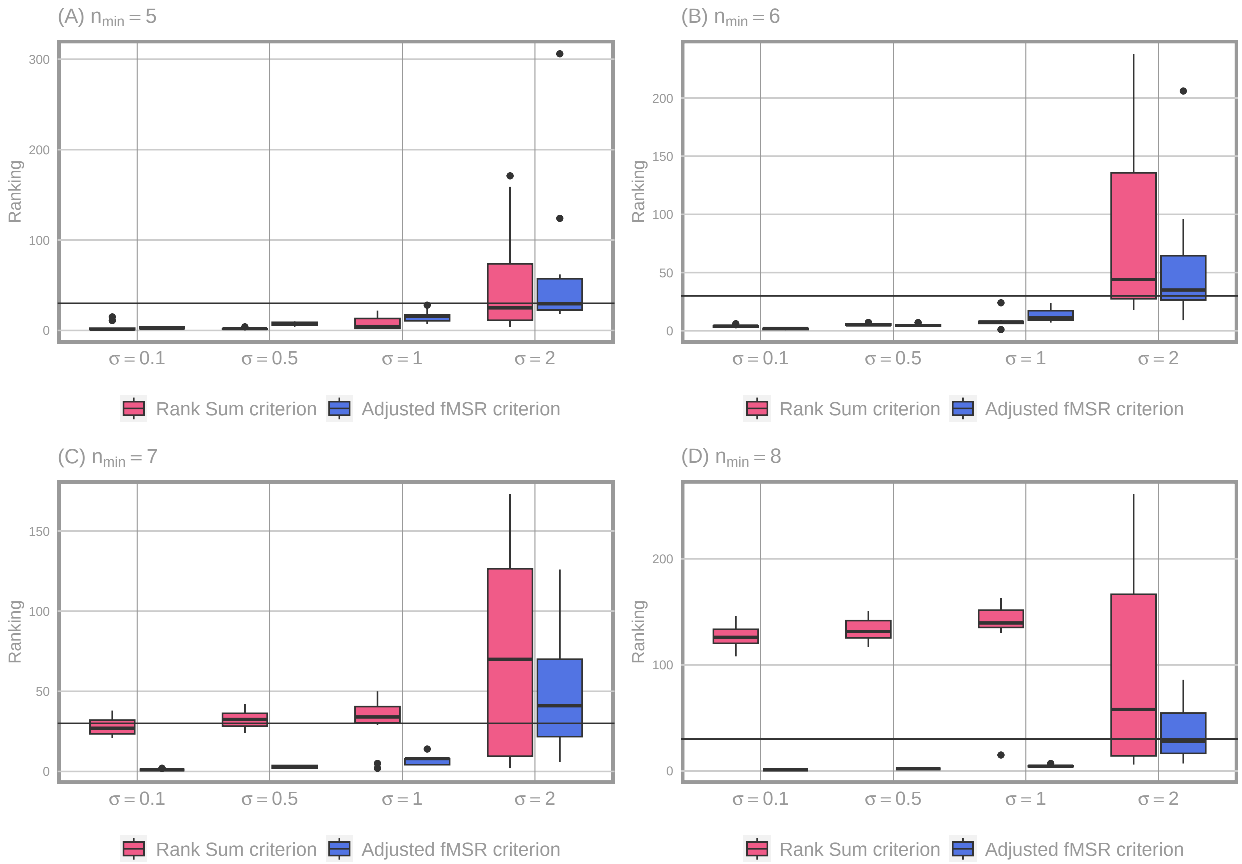


Figure S17: Ranking of the results of *funBIalign* which are most similar to the intentionally embedded motifs, for all simulations where motifs have 8 occurrences and different noise level. The four panels show the ranking of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the red and blue boxplots (left and right) represent ranking according to the rank sum and the adjusted fMSR criterion, respectively. Horizontal line indicates rank 30.

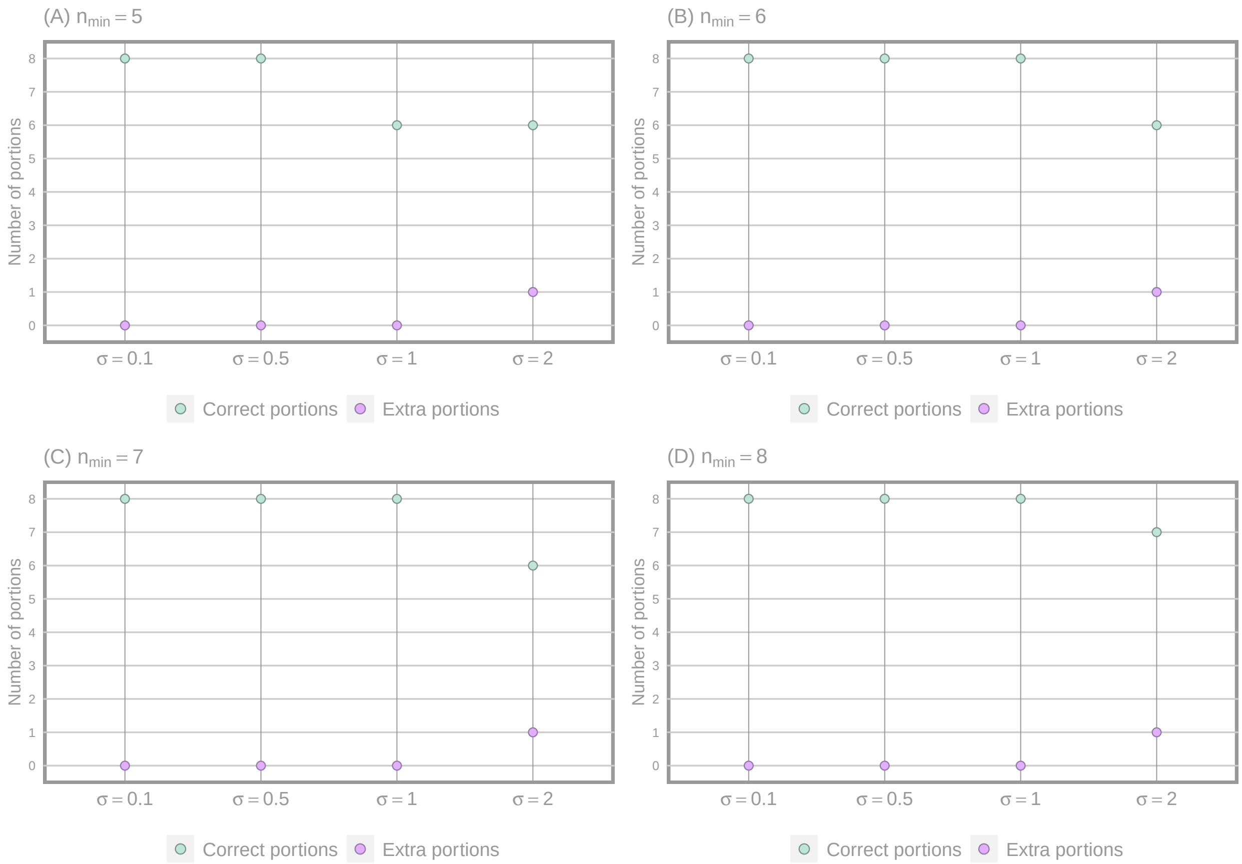
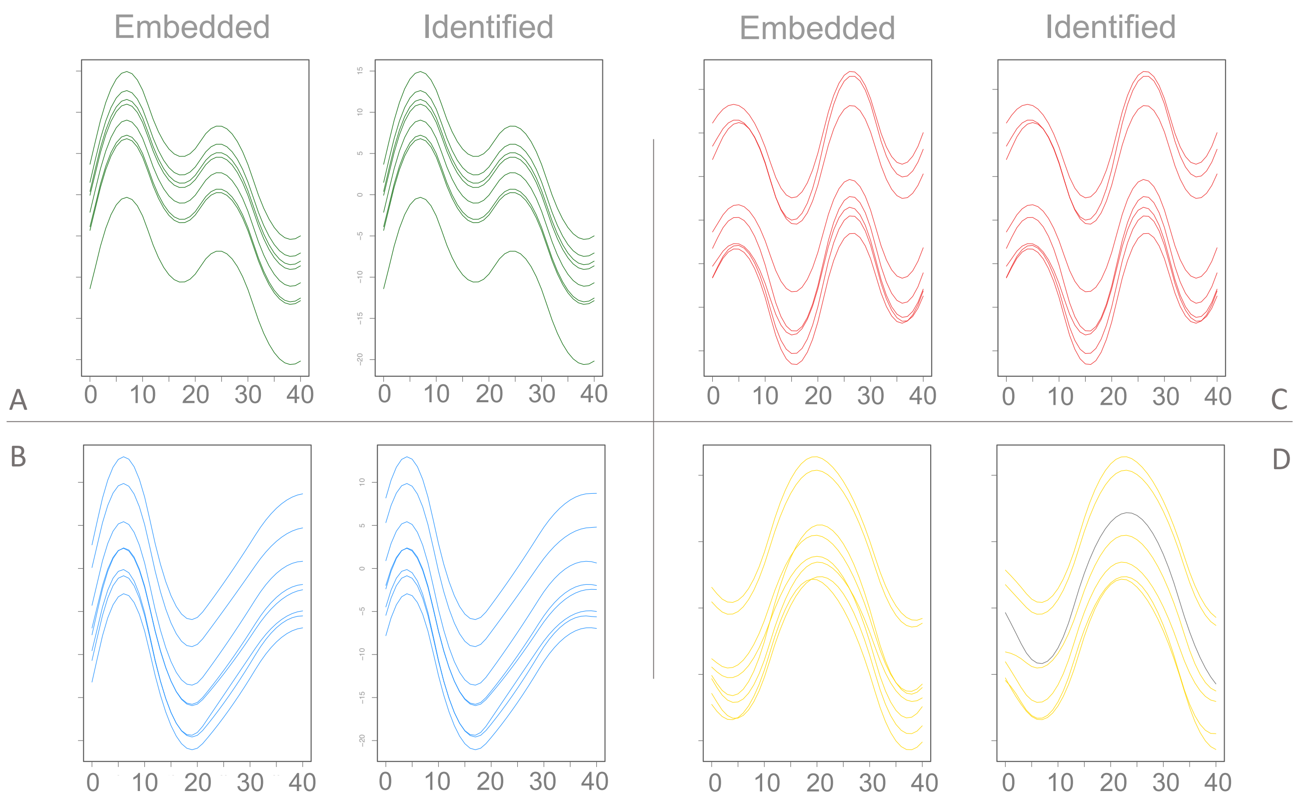


Figure S18: Performance of *funBIalign* for simulations employing motif set n. 7 (where motifs have 8 occurrences and different noise level), shown separately for runs with varying n_{min}. Green and pink jittered dots represent correctly identified portions and extra portions, respectively.



	Adj fMSR ranking	Rank Sum ranking	Correct occurrences	Extra occurrences	Missing occurrences
A	1	6	8	0	0
B	4	7	8	0	0
C	7	8	8	0	0
D	9	24	6	1	2

Figure S19: Comparisons between the four embedded motifs (left) and the most similar ones identified by *funBIalign* (right) for motif set n. 7 and different noise level (where motifs have 8 occurrences). The algorithm is run with minimum cardinality $n_{min} = 6$. The table reports the ranking according to the adjusted fMSR and the rank sum criteria, as well as details on the number of correct, extra, and missing occurrences.

S5.2 Motifs with 10 occurrences

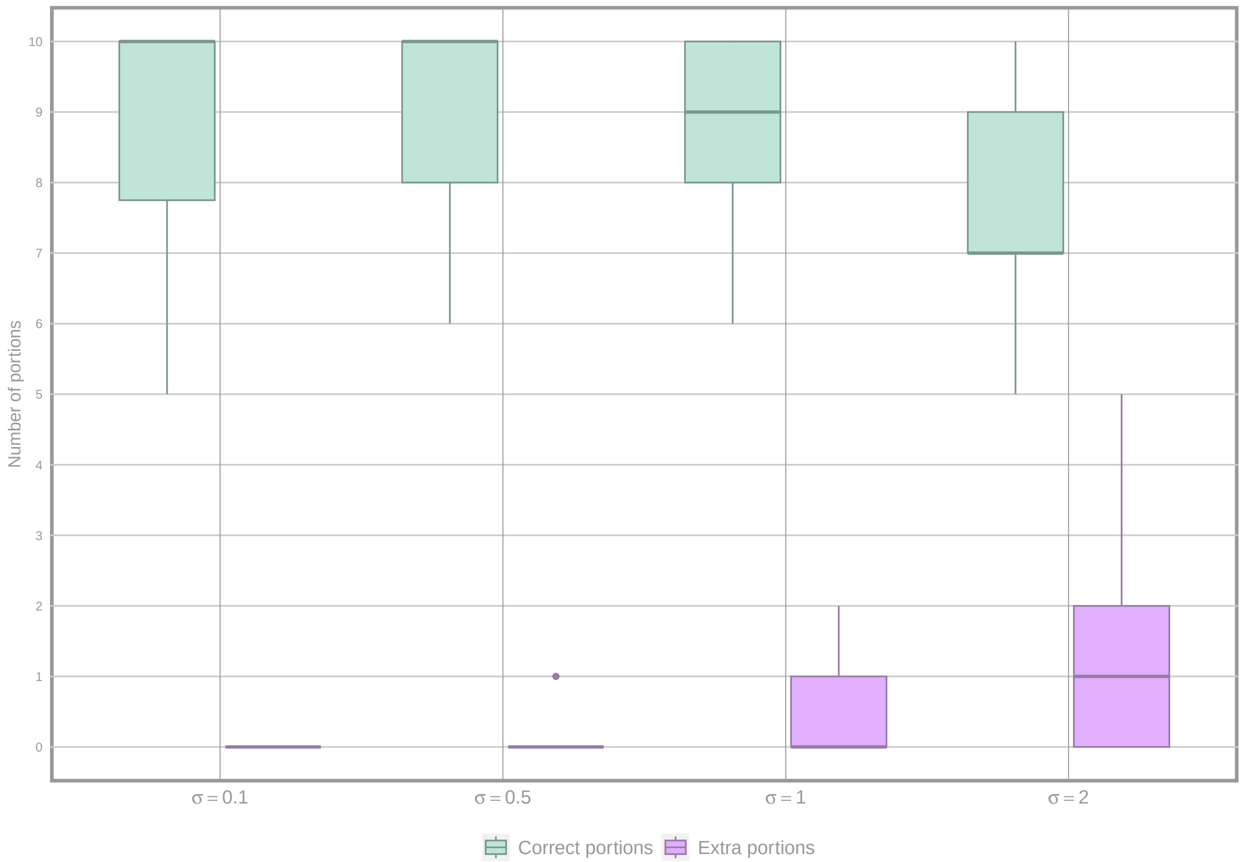


Figure S20: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and different noise levels. The algorithm is run with minimum cardinalities $n_{min} = 5, 6, 7, 8$ and results are pooled. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively. See Figure 2.A.

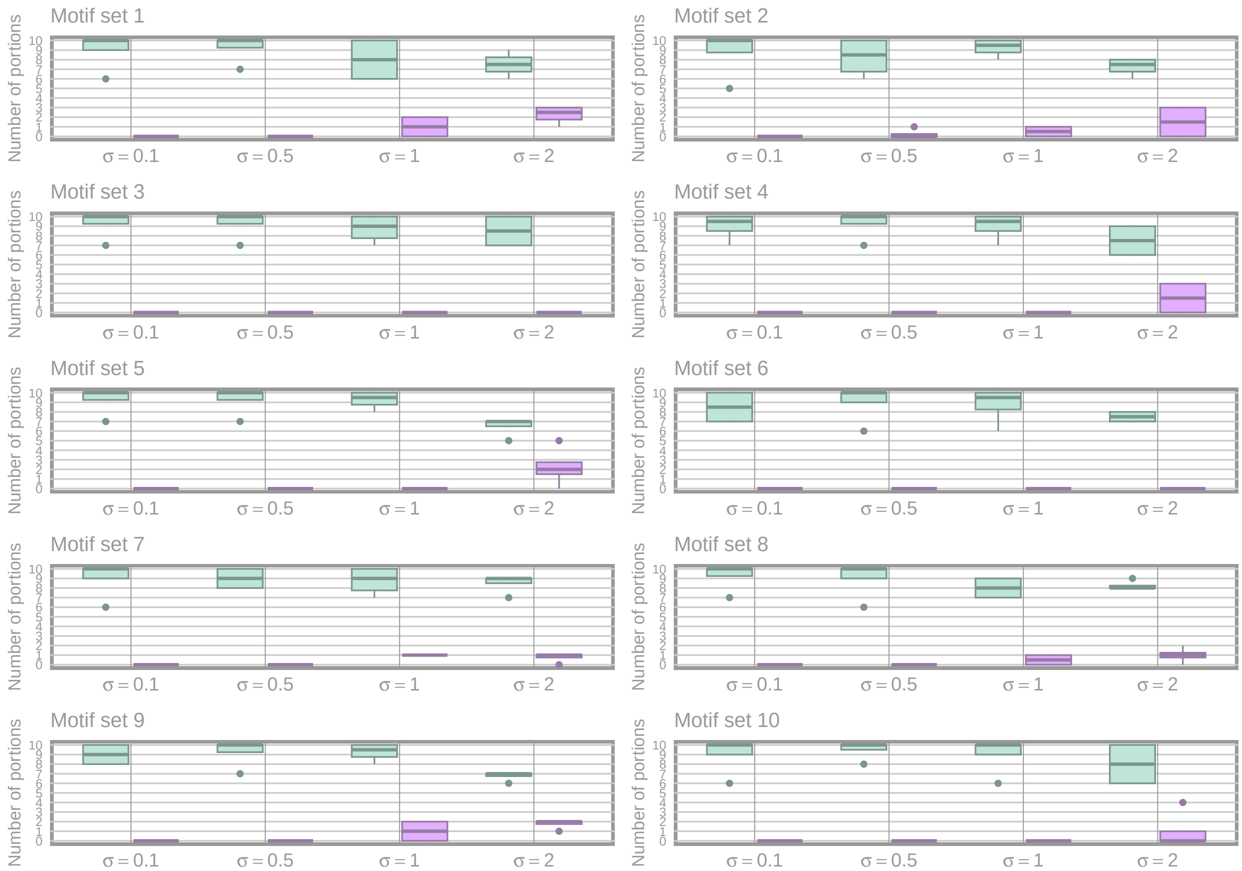


Figure S21: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and different noise level. Different panels show the performance pooling across runs of the algorithm with minimum cardinalities $n_{min} = 5, 6, 7, 8$, but separately for each of the 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

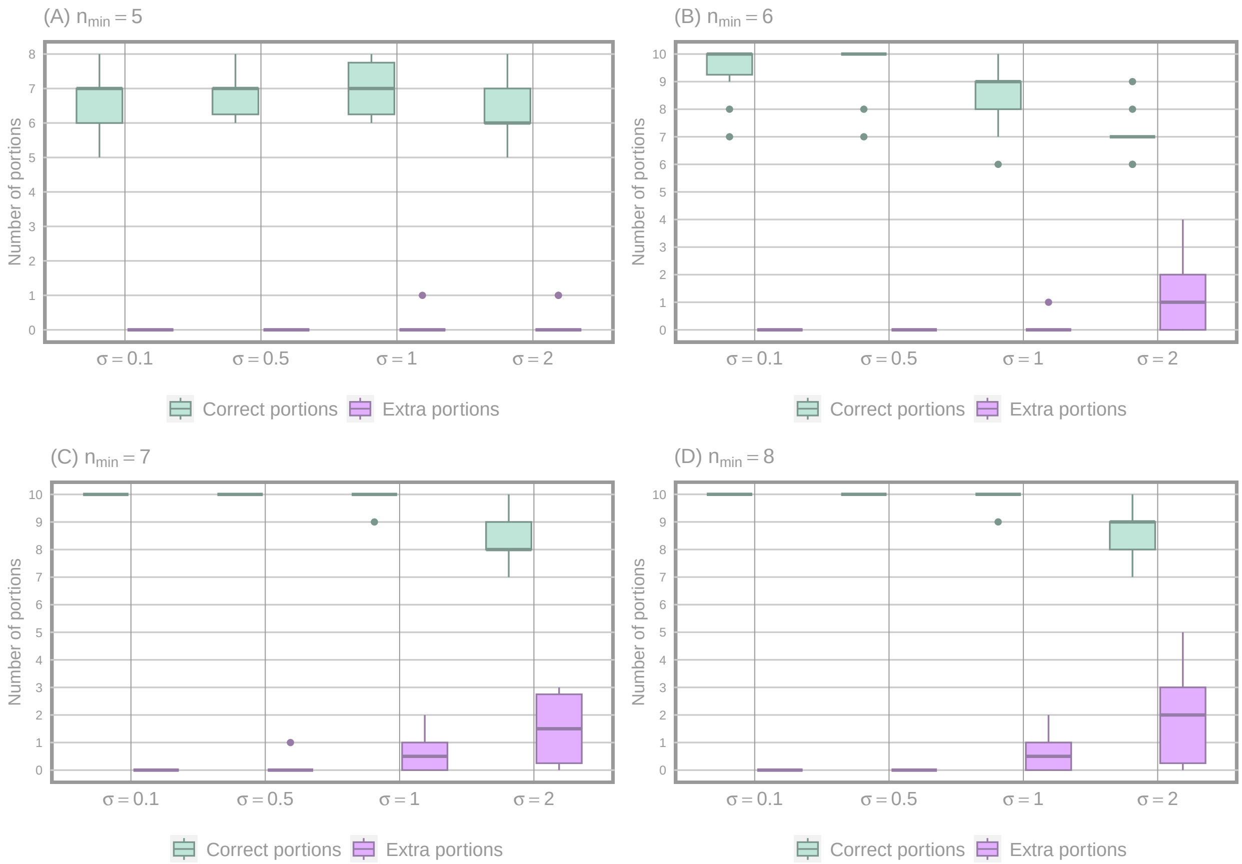


Figure S22: Performance of *funBIalign* for all simulations where motifs have 10 occurrences and different noise level. The four panels show the performance of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the green and pink boxplots (left and right) represent correctly identified portions and extra portions, respectively.

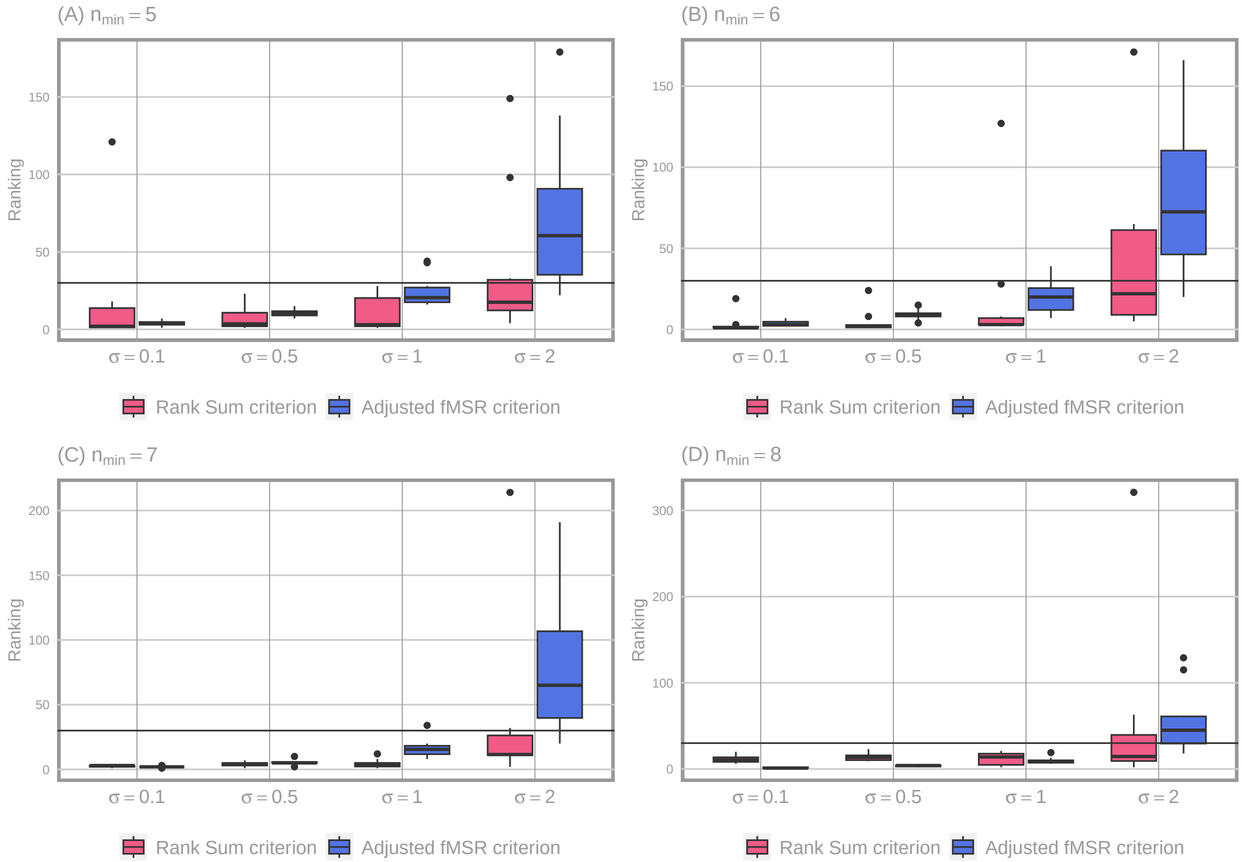
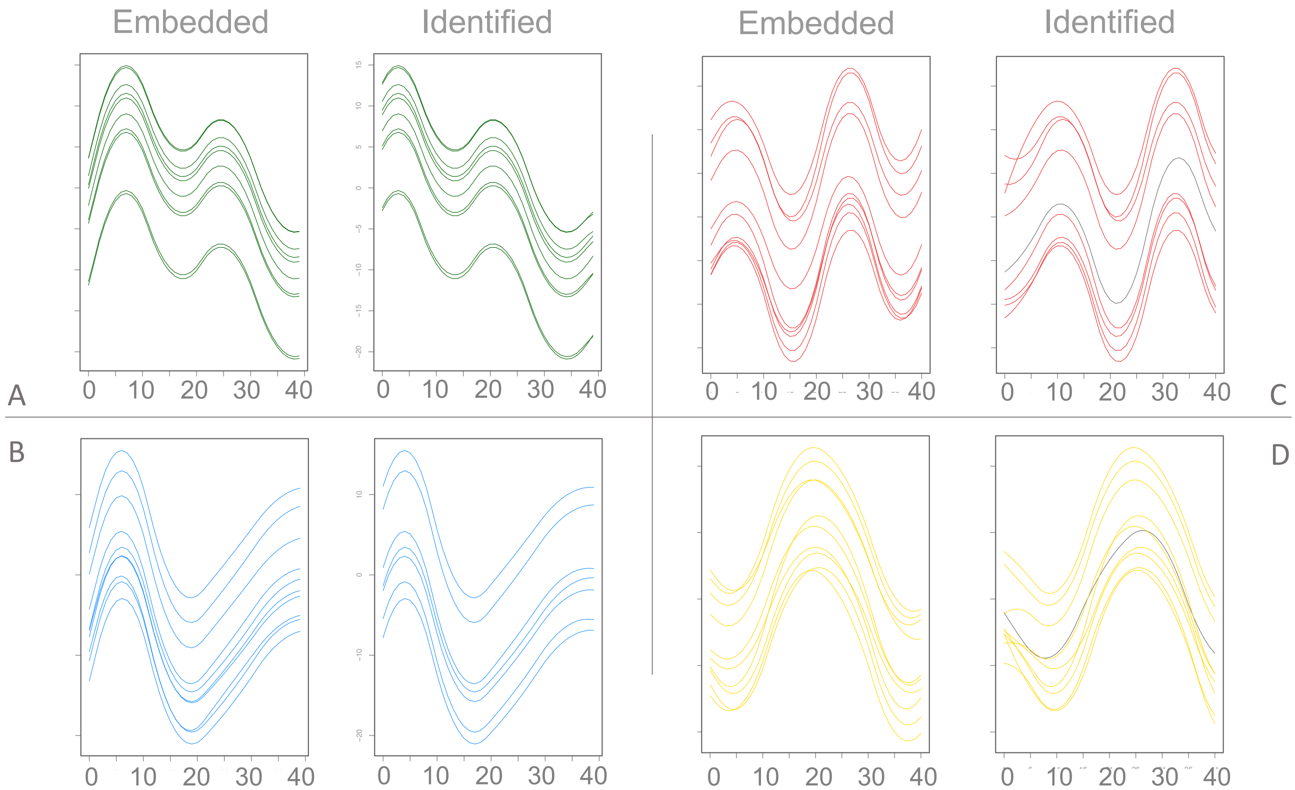


Figure S23: Ranking of the results of *funBIalign* which are most similar to the intentionally embedded motifs, for all simulations where motifs have 10 occurrences and different noise level. The four panels show the ranking of the algorithm run with minimum cardinalities $n_{min} = 5, 6, 7, 8$, respectively, but pooling across 10 alternative motif sets. For each σ, the red and blue boxplots (left and right) represent ranking according to the rank sum and the adjusted fMSR criterion, respectively. Horizontal line indicates rank 30.



	Adj fMSR ranking	Rank Sum ranking	Correct occurrences	Extra occurrences	Missing occurrences
A	3	1	10	0	0
B	9	8	8	0	2
C	26	8	8	1	2
D	119	9	9	1	1

Figure S24: Comparisons between the four embedded motifs (left) and the most similar ones identified by *funBIalign* (right) for motif set n. 7 and different noise level (where motifs have 10 occurrences). The algorithm is run with minimum cardinality $n_{min} = 6$. The table reports the ranking according to the adjusted fMSR and the rank sum criteria, as well as details on the number of correct, extra, and missing occurrences.

S6 Comparison with *SCRIMP-MP*

Table S1 presents the comparison between *funBIalign* and *SCRIMP-MP* in the case of motifs with 8 occurrences each and different noise levels, using motif set n. 7 (see Section 4). As in the comparisons of Section 5, *funBIalign* is run with $\ell = 41$ (the true length of the motifs) and $n_{min} = 5, 6, 7, 8$, while *SCRIMP-MP* is run with $u = 50$, $\ell = 41$ and all possible combinations of $R = 3, 10, 25, 50$ and $k_{neighbor} = 4, 6, 8$. As expected, the identification of the motifs is harder for the motif with $\sigma = 2$ with both the methods. In general, *funBIalign* performs better than *SCRIMP-MP*.

Motif	***funBIalign*** - n_{min}				**MP**	R			
	5	6	7	8	$k_{neighbor}$	3	10	25	50
Motif 1 ($\sigma = 0.1$)	**(8,0)**	**(8,0)**	**(8,0)**	**(8,0)**	4	(6,0)	(6,0)	(6,0)	(6,0)
					6	(7,0)	**(8,0)**	**(8,0)**	**(8,0)**
					8	(7,0)	**(8,0)**	**(8,0)**	**(8,0)**
Motif 2 ($\sigma = 0.5$)	**(8,0)**	**(8,0)**	**(8,0)**	**(8,0)**	4	(6,0)	(6,0)	(6,0)	(6,0)
					6	**(8,0)**	**(8,0)**	**(8,0)**	**(8,0)**
					8	**(8,0)**	**(8,0)**	(8,2)	(8,2)
Motif 3 ($\sigma = 1$)	(6,0)	**(8,0)**	**(8,0)**	**(8,0)**	4	(6,0)	(6,0)	(6,0)	(6,0)
					6	(7,1)	(7,1)	(7,1)	(7,1)
					8	(8,1)	(8,2)	(8,2)	(8,2)
Motif 4 ($\sigma = 2$)	(6,1)	(6,1)	(6,1)	(7,1)	4	(4,2)	(4,2)	(4,2)	(4,2)
					6	(5,3)	(5,3)	(5,3)	(5,3)
					8	(6,4)	(7,3)	(7,3)	(7,3)

Table S1: Comparison between *funBIalign* and *SCRIMP-MP*. Cases in which all occurrences are correctly identified without the addition of any extra portions are in bold.

S7 Case studies

S7.1 Case study 1: Food price inflation

The 19 curves considered in this case study represent the food price inflation measurements for the following geographical regions:

- Eastern Africa
- Middle Africa
- Northern Africa
- Southern Africa
- Western Africa
- Northern America

- Central America
- Caribbean
- South America
- Central Asia
- Eastern Asia
- Southern Asia
- South-eastern Asia
- Western Asia
- Eastern Europe
- Northern Europe
- Southern Europe
- Western Europe
- Oceania

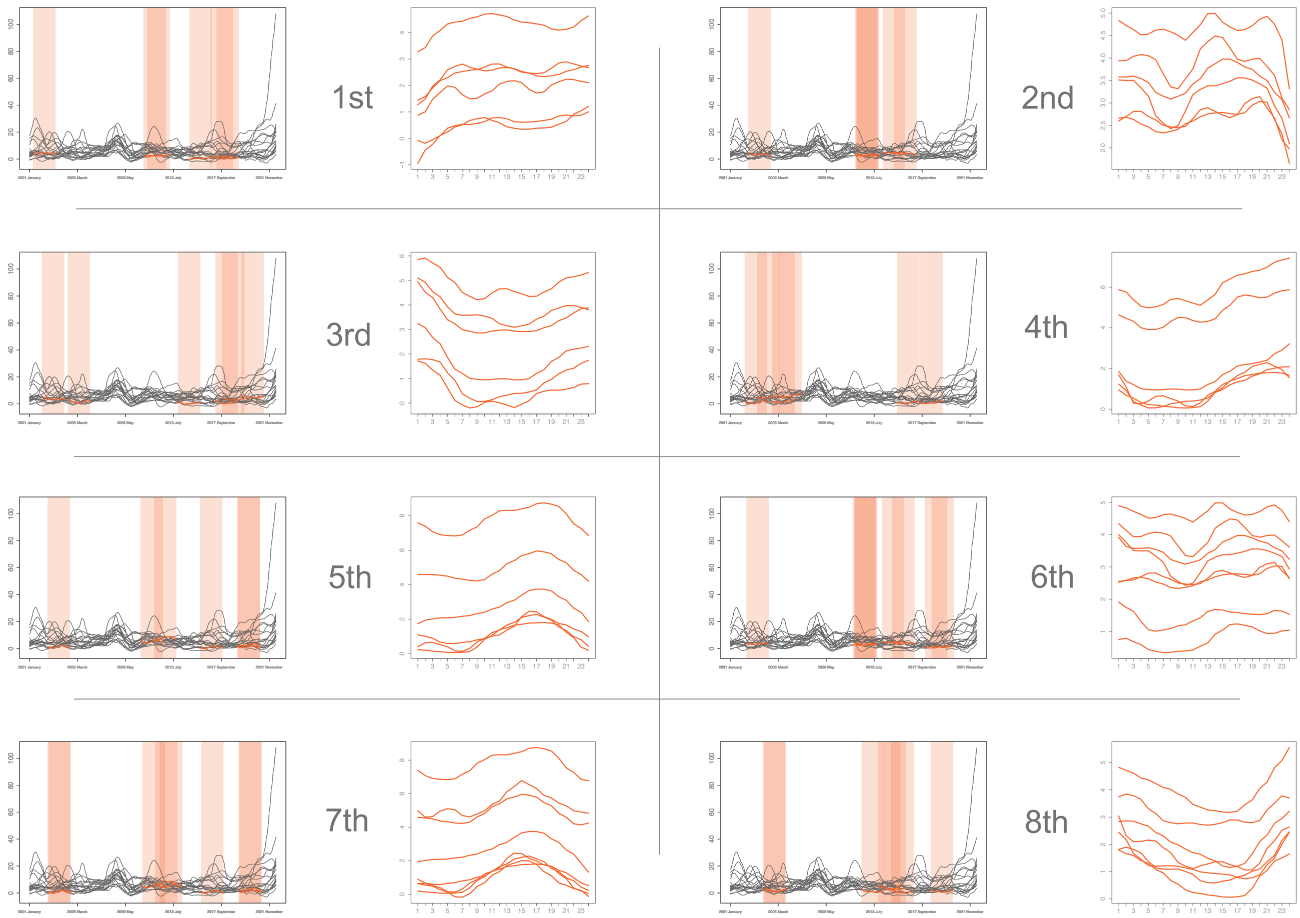


Figure S25: Best 8 motifs according to the adjusted fMSR criterion in the curves for the 19 geographical regions. It is possible to notice how some motif presents occurrences with shared or overlapping domains.

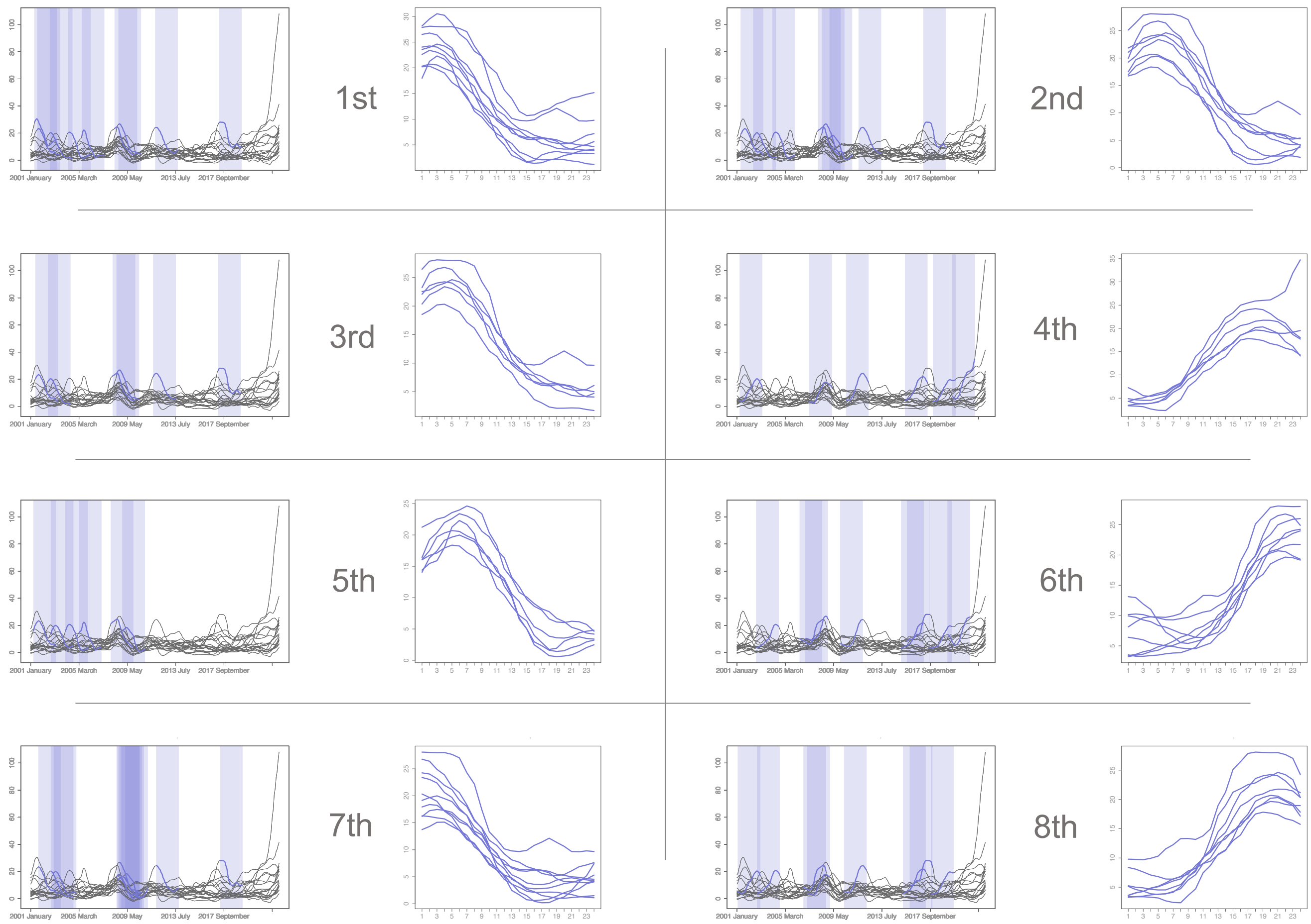


Figure S26: Best 8 motifs according to the variance criterion for the curves for the 19 geographical regions. It is possible to notice how some motif presents occurrences with shared or overlapping domains.

S7.2 Case study 2: Temperature changes

The 19 curves used in this case study represent the temperature change with respect to a baseline climatology for the following regions:

- Eastern Africa
- Middle Africa
- Northern Africa
- Southern Africa
- Western Africa
- Northern America
- Central America
- Caribbean
- South America
- Central Asia
- Eastern Asia
- Southern Asia
- South-eastern Asia
- Western Asia
- Eastern Europe
- Northern Europe
- Southern Europe
- Western Europe
- Oceania