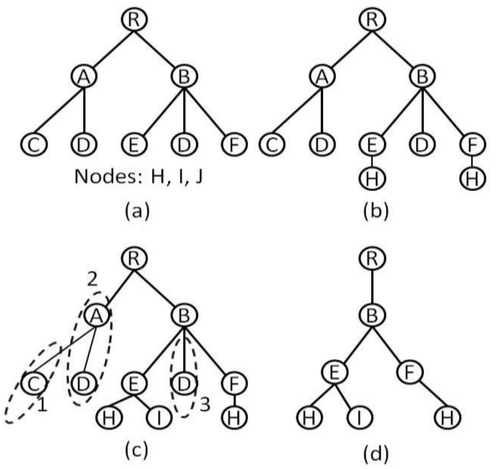
Section A – Illustration of tree construction and clique refinement

In this section, by using examples, the processes of tree construction and clique refinement in TreeMotif are explained using Figures (a)-(d).



Figs. (a)-(d). Illustration of tree construction and clique refinement

Fig. (a) ***An intermediate tree of depth 3 and nodes to append***. Supposing the tree is built of nodes in P_2 to P_4, where $\{R\}\subseteq P_2$, $\{A, B\}\subseteq P_3$, $\{C, D, E, F\}\subseteq P_4$; and new candidate nodes $\{H, I, J\}\subseteq P_5$ are to be appended to the tree. Suppose each node in sets $\{R, A, C\}$, $\{R, A, D\}$, $\{R, B, D\}$, $\{R, B, E\}$, $\{R, B, F\}$ has Hamming distances less than or equal to $2d$ with the other nodes in the same set. Assume H has Hamming distances less than or equal to $2d$ with nodes R, A, B, E, F and more than $2d$ with nodes C, D. Also assume that node I has Hamming distances less than or equal to $2d$ with nodes R, B, C, E and more than $2d$ with nodes A, D, F. Assume J has Hamming distances less than or equal to $2d$ with nodes B, C, D, E, F and more than $2d$ with nodes A, R.

Fig. (b) ***Tree construction with nodes H, I and J***. The tree is filtered using the function *extendable*() (Algorithm 2 of Section 2.2 in the main text) and appended according to queue Q. Specifically, Q should keep several leaf nodes with depth 3 (if any) in the tree as the retained elements after filtering. Firstly, node H is checked. According to the assumptions, Q will change as follows: Q is initialized as $Q=\{R\}$ as $D(R, H)\leq 2d$; then remove R from Q (now $Q=\Phi$) and check its children A, B: insert A into Q as $D(A, H)\leq 2d$ (now $Q=\{A\}$); insert B into Q as $D(B, H)\leq 2d$ (now $Q=\{A, B\}$); then remove A from Q (now $Q=\{B\}$) and check its children C, D: as $D(C, H)>2d$ and $D(D, H)>2d$, none of the children of A is inserted into Q; then remove B from Q (now $Q=\Phi$) and check its children E, D, F: insert E into Q as $D(E, H)\leq 2d$ (now $Q=\{E\}$); skip D as $D(D, H)>2d$ (still $Q=\{E\}$); insert F into Q as $D(F, H)\leq 2d$ (now $Q=\{E, F\}$). The filtering process will terminate as the depth of E is equal to 3, which means leaf nodes of the tree before any appending have been reached. Final elements in Q, i.e. leaf nodes E and F, correspond to branches that are extendable by the new node H, thus append H to these two leaf nodes respectively.

A similar process is carried out for node I. Queue Q changes as follows: Φ, $\{R\}$, Φ, $\{B\}$, Φ, $\{E\}$ (note that all children of A are skipped as $D(A, I)>2d$, which eliminates redundant traversing). So node I can only be appended to leaf node E, as shown in Fig. i(c). For node J, as $D(R, J)>2d$, there is no need to traverse the tree. Now as there are no new nodes from P_5, the filtering and appending process have been completed for S_5.

Fig. (c) ***Elimination of spurious branches in tree construction***. That is, this is an intermediate tree which needs to be checked further if there is any deletable branch for S_5 to be pruned. This related to the first motif refinement strategy (in Section 2.3 in the main text). In detail, leaf nodes C, D (the child of A), D (the child of B) have depth less than 4, therefore three deletable branches (sub-branches) are found in the new tree and need to be pruned away. The paths of backtracking for pruning have been shown in the dashed ellipses.

Fig. (d) ***Merging redundant tree branches after tree construction.*** This is a new tree after processing for all new nodes from P_5. If m=6 (two reference sequences for node selection included), it is a final tree with three branches. In this case, branches of the tree stand for cliques of motif instances. For example, $\{R, B, E, H\}$ and $\{R, B, E, I\}$ are two branches, i.e., they can form two cliques of size 6 (with the reference node pair). Consensus motif is obtained using such cliques, supposing the base on each position is deterministic. Furthermore, note that the two cliques share not only the reference nodes (for node selection) but also nodes $\{R, B, E\}$ and might refer to the same motif. Therefore, such cliques are merged as one clique if all the nodes are within 2d (making the clique size larger than m); otherwise, they are deemed as different cliques. For the above example, a clique of $\{R, B, E, H, I\}$ is formed if $D(H, I)\leqslant 2d$. In this way, more random substrings are included in the final cliques. At the same time, this can guarantee more target instances included in the same clique. This relates to the second clique refinement strategy (in Section 2.3 in the main text).

On the other hand, if m>6, it is a tree that can be retained to be further processed. That is, the new nodes from P_6, P_7… will be tested whether they can be appended onto the new tree. If m>6 and suppose none of the leaf nodes in the tree has a child appended from P_6, there is no need to explore the tree anymore for the other nodes from P_i, i>6. As with OOPS data, if there are no motif instances in P_6 appended onto the tree, OOPS condition is not satisfied. This relates to a termination condition of tree construction (see Algorithm 3 in Section 2.2).

Section B – Experiment on effects of GC content on performance of TreeMotif

This experiment demonstrates the effect of GC-content on the performance of TreeMotif (deterministic algorithm) and compares the performance with MEME (probabilistic algorithm).

1 EXPERIMENTAL DATASETS

By changing GC-content, the similarity of sequences was changed. That is, if AT and GC do not appear with equal opportunity, say GC-content making up 35 percent of all bases, sequences will become more similar to each other (than i.i.d. case). As a result, substrings will also become more similar to each other. This makes the motif discovery problem become more difficult.

In this experiment, eight datasets with 20 sequences of length 600 were generated for each GC-content of 45, 40 and 35 percent. Each sequence was embedded with one instance of a (15, 4)-motif at a random position.

2 RESULTS AND DISCUSSION

The performances (nPC and execution time) of MEME and TreeMotif are compared, as shown in Table 1 and 2. Each mean value in the following tables is an average of eight datasets. To be noted, when GC-content is increased from 55 percent to 65 percent, similar results can be obtained.

Table 1. nPC of MEME and TreeMotif with decreased %GC-content

GC (percentage)		45	40	35
nPC	MEME	0.17±0.24	0.25±0.28	0.38±0.35
	TreeMotif	0.93±0.07	0.89±0.15	0.01±0.01

Table 2. Execution Time of MEME and TreeMotif with decreased %GC-content

GC (percentage)		45	40	35
Time	MEME	3.02±0.06	3.01±0.07	3.00±0.06
	TreeMotif	0.91±0.11	2.39±0.76	159.9±123.11

Unit: second

From Table 1, it can be seen that the nPC of TreeMotif decreased significantly when GC-content decreases. However, this lower nPC does not mean that TreeMotif did not find the correct clique of motif instances. TreeMotif ranks the cliques using information content (Schneider *et~al.*, 1990) which usually works fine to report the correct motif instances. However, in datasets with higher sequence similarity, more spurious cliques will be found. For some spurious cliques, they might show higher information content than the true one. This inhibits the target clique of motif instances to be reported (thus lower nPC). That is, the target clique is (found but) not reported in the top-ranking ones makes the performance of TreeMotif lower. If cliques with deterministic consensus motifs are output, the nPC can be greatly increased. Therefore, a future work to solve this problem would include adopting a better scoring function for ranking the cliques. For MEME, although it still achieves higher nPC when GC-content is 35 percent, the large standard error indicates that MEME does not perform consistently.

From Table 2, it can be seen that the execution time of TreeMotif grows significantly when GC-content decreases. Therefore, its time efficiency still needs to be improved as well.

REFERENCES

Schneider, T. D. and Stephens, R. M. (1990) Sequence Logos: a New Way to Display Consensus Sequences. Nucleic Acids Research, 18(20), 6097-6100.