"main" — 2018/1/18 — 13:15 — page 1 — #1





1

```
Algorithm 1 H1 Greedy Elimination
```

```
function method(motifs)
supernodes \leftarrow \varnothing
for \ i \leftarrow 1 \ to \ n \ do
shuffled\_motifs \leftarrow random\_shuffle(<math>motifs)
disjoint\_motifs \leftarrow \varnothing
for \ each \ motif \in shuffled\_motifs \ do
if \ motif \ has \ not \ overlaps \ in \ disjoint\_motifs \ then
Add \ motif \ in \ disjoint\_motifs
end \ if
end \ for
if \ disjoint\_motifs \ contains \ more \ motifs \ than \ supernodes \ then
supernodes \leftarrow disjoint\_motifs
end \ if
end \ for
return \ supernodes
end \ function
```

Algorithm 2 H2 Ramsey

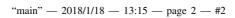
```
function method(motifs, k)
   candidates \leftarrow \emptyset
   if motifs contains overlaps then
      flag \leftarrow True
   end if
   while flag do
      flag \leftarrow False
      Divide motifs in n samples almost of size k, call them subsets
      \textbf{for } subset \in subsets \ \textbf{do}
         Build an overlap graph g
         Perform the Ramsey algorithm on g and save the maximum independent set in mis_g
         Move all motifs in mis_g to candidates
      end for
      if candidates contains overlaps then
         motifs \leftarrow candidates
         flag \leftarrow True
      end if
   end while
   return candidates
end function
```

Algorithm 3 H3 Ranked Elimination

```
1: function method(g, motifs)
2:
     Potential Supernodes \leftarrow \varnothing
      \mbox{Compute the degree for each } motif \in motifs \mbox{ and build a list of ordered motifs called } Motif Degree \mbox{ }
3:
4:
     \mathbf{for}\ node \in g\ \mathbf{do}
         Move the motif that contains node with the lowest degree from MotifDegree to PotentialSupernodes
6:
         Remove all motifs that contain n from MotifDegree
7:
     end for
8:
     for each pair m', m'' of overlapping motifs do
        Remove the motif with the highest degree between m', m'' from PotentialSupernodes. Remove a random one if they have the same degree.
10:
      end for
11:
      {\bf return}\ Potential Supernodes
12: end function
```











Algorithm 4 H4 Repeated Ranked Elimination

2

```
1: function method(g, motifs)
      Supernodes \leftarrow \varnothing
      while motifs \neq \emptyset do
3:
         Potential Supernodes \leftarrow \varnothing
4:
5:
          \mbox{Compute the degree for each } motif \in motifs \mbox{ and build a list of ordered motifs called } Motif Degree \\
6:
         \mathbf{for}\; node \in g\; \mathbf{do}
            Move the motif that contains node with the lowest degree from MotifDegree to PotentialSupernodes
7:
8:
            Remove all motifs that contain n from MotifDegree
10:
         for each pair m', m'' of overlapping motifs do
             Remove the motif with the highest degree between m', m'' from Potential Supernodes. Remove a random one if they have the same
11:
   degree.
12:
         end for
         Consider the list N_{orphan} of all nodes that are not covered by a motif from Potential Supernodes
13:
14:
          Filter motifs keeping only motifs that apply to N_{orphan}
15:
          Move all motifs from Potential Supernodes to Supernodes
       end while
16:
17:
       {\bf return}\ Supernodes
18: end function
```

Algorithm 5 H5 Sampled Ranked Elimination

```
1: function method(motifs, k)
2:
      candidates \leftarrow \varnothing
3:
      if motifs contains overlaps then
4:
         flag \leftarrow True
5:
      end if
      while flag do
6:
         flag \leftarrow False
7:
         Divide motifs in n samples almost of size k, call them subsets
8:
9:
         \textbf{for } subset \in subsets \ \textbf{do}
10:
              Build an overlap graph g
              Remove motifs that overlap from g starting with the motifs with the higher degree than other motifs and save the surviving in mis_g
11:
12:
              Move all motifs in mis_g to candidates
13:
          {\it if}\ candidates\ {\it contains}\ {\it overlaps}\ {\it then}
14:
15:
              motifs \leftarrow candidates
16:
              flag \leftarrow True
          end if
17:
       end while
18:
19:
       {\bf return}\ candidates
20: end function
```



