
Algorithm 1 H1 Greedy Elimination

```

function method(motifs)
    supernodes  $\leftarrow \emptyset$ 
    for  $i \leftarrow 1$  to  $n$  do
        shuffled_motifs  $\leftarrow$  random_shuffle(motifs)
        disjoint_motifs  $\leftarrow \emptyset$ 
        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
        for subset  $\in$  subsets 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:   PotentialSupernodes  $\leftarrow \emptyset$ 
3:   Compute the degree for each motif  $\in$  motifs and build a list of ordered motifs called MotifDegree
4:   for node  $\in g$  do
5:     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
9:     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:  return PotentialSupernodes
12: end function
    
```

2

Algorithm 4 H4 Repeated Ranked Elimination

```

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

```

Algorithm 5 H5 Sampled Ranked Elimination

```

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

```
