CliXO 1.0 (G, α, β, m)

Inputs: Weighted graph G representing proteins (nodes) and pairwise IAS scores (edges); nonnegative floats α, β in (0, 1]; positive float $m < 1$

Outputs: C, a set of protein communities, where each community is a set of proteins

Initializations:

E = Stack of all edges $\in G$, sorted by descending weight

Score threshold t = Weight(Top(E))

$C = \emptyset$; $G_{infer} = \emptyset$; $G_{actual} = \emptyset$

```
while (E ≠ ∅ and t > 0) {
      E_new ← Pop edges from E with weight > (t - α)
      G_infer ← G_infer + E_new;
      G_actual ← G_actual + E_new;
      C_current ← findCliques(G_infer);
      C_new ← C_current;
      for all cliques c_k ∈ C_current {
            while C_new ≠ ∅ {
                  C_l ← pop(C_new);
                  if isSimilar(c_k, c_l, G_actual, β) {
                        c_merge = c_k ∪ c_l;
                        G_infer ← G_infer + G[c_merge];
                        C_current ← C_current \ {c_k, c_l} ∪ {c_merge};
                        C_new ← C_new ∪ {c_merge};
                  }
            }
      }
      for (c ∈ C_current)
            if (c ∉ C) && isModular(c, m, G_actual)
                  C ← C ∪{c};
      t ← t - α;
}
```

For **findCliques** we used a maximal clique enumeration algorithm proposed by Chiba and Nishizeki (Chiba and Nishizeki, 1985). Other subroutines used in the algorithm above are described as follows:

isSimilar(*c*$_1$, *c*$_2$, *G*, *β*)

```
if
```

$$S = \frac{\sum_{v \in C_1, w \in C_2} \max(A_{vw} - k_v k_w / 2n, 0)}{\sum_{i,j \in C_1 \cup C_2} |A_{ij} - k_i k_j / 2n|} > \beta$$

```
return true;
else return false;
```

where A is the graph adjacency matrix with A_{ij} = 1 if *(i, j)* is an edge in *G*, otherwise 0; k_i is the degree of vertex *i* in *G*; and *n* is the number of edges in *G*. This function evaluates the similarity between two protein sets c_1 and c_2 given network *G* and similarity threshold *β*. The more similar c_1 and c_2, the edges in *G* induced by $c_1 \cup c_2$ become more similar to the edges between c_1 and c_2. The similarity calculation weighs edges according to their contribution to modularity.

isModular(*c*, *m*, *G*)

$$Q = \frac{1}{2n} \sum_{i,j \in C} (A_{ij} - \frac{k_i k_j}{2n})$$

```
if Q > m
      return true;
else return false;
```

where *n* is the number of edges in the current graph *G*. For a set of proteins *c*, this function calculates its quality function Q (contribution to modularity).