Figure 1 - Pseudo code for Pro-Kmeans algorithm

**Input:** A training set $D$, $D = \{O_h\}_{h=1..n}$; $n$ is the size of $D$

**Initialize:** $f(V)_{\text{max}} = 0$; iteration $= 0$;

**Repeat**

1. Partition randomly $D$ into $K$ nonempty subsets;

2. **For** each $i \in [1..K]$ do
   - Compute the similarity score of each pair of proteins in the subset $S_i$ using Smith Waterman algorithm;
   - Compute the SumScore($S_i$, $O_j$) of each protein $j$ in $S_i$;
   - The protein $j$ which have the maximum SumScore($S_i$, $O_j$) in $S_i$ is considered as the centroid $R_i$ of the subset $S_i$;

3. **For** each $O_h \in D$ do
   - Compute the similarity score of $O_h$ with each centroid $R_i$ ($i \in [1..K]$), using Smith Waterman algorithm;
   - Assign $O_h$ to the cluster with the nearest $R_i$; (The $R_i$ which have the maximum score of similarity with the object $O_h$)

4. Compute $f(V)$;

5. **If** $f(V) < f(V)_{\text{max}}$ **then**
   - iteration $= \text{iteration} + 1$;
   **Else**
   - $f(V)_{\text{max}} = f(V)$;
   - BestSets = CurrentSets; (CurrentSets are Subsets obtained in this partition)
   - Go back to Step 2;

**Until** iteration $= q$;

**End**

**Output:** BestSets; BestSets is the best partition of $D$ into $K$ clusters; each cluster is defined by a centroid $R_i$