LaTeX Author Guidelines for 8.5×11-Inch Proceedings Manuscripts

Paolo Ienne
Swiss Federal Institute of Technology
Microcomputing Laboratory
IN-F Ecublens, 1015 Lausanne, Switzerland
Paolo.Ienne@di.epfl.ch

Second Author
Institution2
First line of institution2 address
Second line of institution2 address
SecondAuthor@institution2.com

0.1. Algorithm

Let E be some experiment from a research paper with N participants . Then we have that $E = \{P_1, P_2, P_3, ..., P_N\}$. Each participant in the experiment P_i has a set of l microbial samples taken from a particular body site, and each sample m has a time when it was taken t. Then we have that $S^t(P_i) = \{m_{S_1}, m_{S_2}, ..., m_{S_l}\}$ where each S_j is a sample from participant P_i, and $i > 0$, $j > 0$, and $l > 0$. Then each microbial sample m_{S_j} is a set of x operational taxonomic units or bugs such that we have $m_{S_j} = \{b_1, b_2, ..., b_x\}$ where $x > 0$.

Assume an attacker has k samples taken from a single individual at a time t referred to as mr.duck smith where k greater than or equal to 1. Then $A = \{a_1, a_2, a_3, ..., a_k\}$ is the set of the samples from mr. duck smith that the attacker has collected.

Then let P be all the participants in experiment E, S be all the samples in experiment E, and O be all the possible operational taxonomic units (OTU) . Let N be the number of participants in experiment E, M be the number of samples in experiment E, and L be the number of OTUs in O.

Then the following procedures define an algorithm for determining if mr. duck smith is a participant in experiment E :

Algorithm 1 Probability Based Microbial Signature Algorithm

```
procedure MakeSignature($P_i$,$S$,$O$,$percent$)
    $i \leftarrow 0$
    $j \leftarrow 0$
    $count \leftarrow 0$
    $sig \leftarrow []$
    for $i = 1 \rightarrow length(O)$ do
        for $j = 1 \rightarrow length(S)$ do
            $sample \leftarrow S[j]$
            if $O[i] \in sample$ then
                $count \leftarrow count + 1$
            $totalPercent \leftarrow count \setminus length(S)$
            if $totalPercent \geq percent$ then
                $sig \leftarrow sig.append(O[i], 1)$
            else
                $sig \leftarrow sig.append(O[i], 0)$
            $count \leftarrow 0$
procedure RemoveOTUS($P$, $Sigs$, $O$, $percent$)
    $i \leftarrow 0$
    $j \leftarrow 0$
    $k \leftarrow 0$
    $count \leftarrow 0$
    $OTUs \leftarrow []$
    for $i = 1 \rightarrow length(O)$ do
        for $j = 1 \rightarrow length(P)$ do
            if  $O[i] \in Sigs(P_j)$ then
                $count \leftarrow count + 1$
            $totalPercent \leftarrow count \setminus length(S)$
            if $totalPercent \geq percent$ then
                $OTUs \leftarrow OTUS.append(O[i])$
        $count \leftarrow 0$
    for  $k = 1 \rightarrow length(Sigs)$  do
        $Sigs[k] \leftarrow Sigs[k].remove(OTUs)$
procedure MatchSignature($s_1$,$s_2$)
    $i \leftarrow 0$
    $count \leftarrow 0$
    for $i = 1 \rightarrow length(S_1)$ do
        if  $S_1[i] xor S_2[i]$ then
            continue
        else
            $count \leftarrow count + 1$
    return count
procedure MatchSignature($Sigs$,$aSig$)
    $i \leftarrow 0$
    $samps \leftarrow []$
    for $i = 1 \rightarrow length(Sigs)$ do
        $value \leftarrow MatchSignature(Sigs[i], aSig])$
        $samps \leftarrow samps.append((Sigs[i]), value)$
    sort samps in ascending order return samps
```