Algorithm 1: Clustermatch algorithm

Function `get_partitions(`$\mathbf{v}$, k`)`**:**

Input:

$\mathbf{v}$: feature values on n objects

k: maximum number of internal clusters

Output:

Ω_r: clustering with r clusters over n objects

if $\mathbf{v} \in \mathbb{R}^n$ **then**

  for $r \leftarrow 2$ **to** $\min\{k, |\mathbf{v}| - 1\}$ **do**

    $\boldsymbol{\rho} \leftarrow (\rho_\ell \mid \Pr(v_i < \rho_\ell) \leq (\ell - 1)/r), \forall \ell \in [1, r+1]$

    $\Omega_{r\ell} \leftarrow \{i \mid \rho_\ell < v_i \leq \rho_{\ell+1}\}, \forall \ell \in [1, r]$

else

  $\mathcal{C} \leftarrow \cup_j \{v_i\}$

  $r \leftarrow |\mathcal{C}|$

  $\Omega_{rc} \leftarrow \{i \mid v_i = \mathcal{C}_c\}, \forall c \in [1, r]$

`// TODO: remove singletons`

return Ω

Function `clustermatch(`$\mathbf{x}$, $\mathbf{y}$, k`)`**:**

Input:

$\mathbf{x}$: feature values on n objects

$\mathbf{y}$: feature values on n objects

k: maximum number of internal clusters

Output:

c: similarity value for $\mathbf{x}$ and $\mathbf{y}$ ($c \in [0, 1]$)

$\Omega^{\mathbf{x}} =$ get_partitions($\mathbf{x}$, k)

$\Omega^{\mathbf{y}} =$ get_partitions($\mathbf{y}$, k)

`// get the maximum adjusted Rand index (A) across all clusterings`

$c \leftarrow \max\{\mathcal{A}(\Omega^{\mathbf{x}}_p, \Omega^{\mathbf{y}}_q)\}, \forall p, q$

return c