
ALGORITHM: Algorithm for measuring term similarities across ontologies.

Input: term set $T_1 = \{t_{1,1}, t_{1,2}, \dots, t_{1,n}\}$, 'IS_A' relationships of T_1 , term set $T_2 = \{t_{2,1}, t_{2,2}, \dots, t_{2,m}\}$, 'IS_A' relationships of T_2 , gene set $G = \{g_1, g_2, \dots, g_k\}$, term-gene pairs set $TG = \{(t_1, g_1), (t_1, g_2), \dots, (t_m, g_k)\}$, weights of gene-gene pair $\{wg_{1,2}, wg_{1,3}, \dots, wg_p\}$.

Output: The similarity of all the term pairs.

- 1 For each term-gene pair (t_i, g_j) in TG
 - 2 Calculate weights of each pair $(w(t_i, g_j))$ using equation 1.
 - 3 For each term-gene pair (t_i, g_j) in TG
 - 4 Normalize weights of each pair $(nw(t_i, g_j))$ using equation 2.
 - 5 For each term t_i in T_1 or T_2
 - 6 Calculate weights of each pair $(w(t_i))$ using equation 3.
 - 7 For each term t_i in T_1 or T_2
 - 8 Normalize weights of each pair $(nw(t_i))$ using equation 4.
 - 9 For each gene-gene pair (g_i, g_j) in T_1 or T_2
 - 10 Normalize weights of each pair $(w(g_i, g_j))$ using equation 5.
 - 11 For each term t_i in T_1 or T_2
 - 12 Calculate the vector of each term (WV_t) using ITM probe.
 - 13 For each t_i in T_1
 - 14 For each t_j in T_2
 - 15 Calculate the similarity of t_i and t_j ($Sim(t_i, t_j)$) using equation 7&8
 - 16 add $Sim(t_i, t_j)$ to $Sim(T_1, T_2)$.
 - 17 return $Sim(T_1, T_2)$.
-