

Struct-net-align

currently a very rough incoherent and partially incorrect sketch

Introduction

Using structure to improve a noisy PPI network.

Algorithm

Definitions

Let $G = (V, Int(G) \cup Hom(G))$ be a network whose vertices represent domains.

Let $Int(G)$ is a set of edges indicating interactions.

Let $Int(v)$ for any vertex $v \in Int(G)$ is the set of edges in $Int(G)$ that are incident to v . Also define $Hom(v)$ in the same way.

$Hom(G)$ is a set of edges indicating homology (similarity) between vertices.

An edge $e \in Int(G)$ is associated with a probability $Prob(e)$.

An edge $(a, b) \in Hom(G)$ is associated with a score denoted $s((a, b))$ ranging from 0 to ∞ . Any edge with score $s \geq 1$ is defined as significant.

Let $\rho(a, b)$ denote the score of the most specific homology relation r identified between nodes a and b . For example, if a and b are from the same SCOP fold but not the same SCOP superfamily, $\rho(a, b)$ will be given by this relation.

Let $\alpha(a, b)$ denote the weighted score of an alignment that has been performed between vertices a and b .

Let $\iota(v)$ denote the number of edges (v, v') in $Int(G)$, $\forall v \in V$, and let $\eta(v)$ denote the number of edges (v, v') in $Hom(G)$, $\forall v \in V$.

Let $\iota^* = \max_{v \in V} \{\iota(v)\}$, $\eta^* = \max_{v \in V} \{\eta(v)\}$.

Overview

The algorithm consists of three major steps.

Given an interaction network $G = (V, Int(G))$, build a network $G' = (V, Int(G') \cup Hom(G'))$ by using any available information (e.g. homology databases or alignment) to add edges to $Hom(G')$ and to increase the weights of those edges. At the end of this process, remove any edge from $Hom(G')$ with weight less than 1.

Second, generate a network $G'' = (V, Int(G''))$ by using edges in $Hom(G')$ to update edge weights (probabilities) in $Int(G)$.

Third, define an equivalence relation between vertices, H , such that $H(u) \equiv H(v)$ iff v is reachable from u by traversing $Hom(G')$. Generate a network $G''' = (V', Int(G''') \cup Hom(G'''))$ by merging nodes contained in cliques of $(V, Hom(G'))$ in which every vertex in the clique is associated with the same set of interaction edges in $(V, Int(G''))$, with respect to the equivalence relation H .

Algorithm

1. Parse network.
2. For every pair of vertices (a, b) , add an edge (a, b) to $Hom(G)$, with label $\rho(a, b)$.
3. For every pair of vertices (a, b) (optionally: not joined by an edge in $Hom(G)$), align a against b and add an edge to $Hom(G)$, with label $\alpha(a, b)$.
4. For every edge $e \in G$ such that $s(e) < 1$, remove e .
5. For every edge $x = (u, v) \in Int(G)$:
 - (a) Run BFS on $(V, Hom(G))$ from u and v concurrently.

- (b) Set $P := 0$.
 - (c) For every edge $y = (s, t) \in \text{Int}(G)$ such that s is reachable from u and t is reachable from v :
 - (d) Let $\epsilon_x := \text{Prob}(i, j) \cdot \exp \left(\sum_{\pi, \rho} \sum_{i, j} \log S(\pi_i, \rho_i) \right)$, where π is a path from u to s and ρ is a path from v to t
 - (e) Let $P := P + \epsilon_x$
 - (f) Let $x = x + \epsilon_x - P$ FIXME
6. Discard every vertex v such that $\iota(v) = 0$ and remove every edge incident to v .
 7. Run Bron–Kerbosch’s algorithm on G'_i to enumerate cliques $C = \{c_1 \dots c_x\}$, for all cliques of size $\gamma(C_j) > 1$.
 8. Partition C into sets $D = d_1 \dots d_z$ such that $\forall v \in d_i, u \in d_j, \text{Int}(v) \neq \text{Int}(u), \forall i, j$, and $\text{Int}_H(v) = \text{Int}_H(u) \forall u, v \in d_i \forall i$.
 9. For each $d \in D$, merge every vertex in d in the reference graph G .
 10. Output resulting network. The network should contain no homology edges.

Defining homology by relations

Defining homology by alignment

Using homology to score interactions

$$\text{Define } R_{i,j} = \begin{cases} 0 & i = 0 \vee j = 0 \\ Q(v_i)Q(v_j)\text{Prob}(v_i, v_j) - \sum_{k < i} R_{i,j} & \text{otherwise} \end{cases}$$

$$\text{Define } Q_i(j) = \begin{cases} 0 & i = 0 \\ \sum_{k < i} \sum_j Q_k(j)S(v_{k,j}, v_{k,i}) & \text{otherwise} \end{cases}$$

Merging equivalent domains

To merge, we require:

1. Two subgraphs G_1, G_2 of G'' with edges in $\text{Int}(G')$

2. $\exists H \subseteq \text{Hom}(G)$ s.t. H defines an isomorphism between G_1 and G_2 (TODO: rigor).

Unfortunately, MAX-CLIQUE is NP-HARD. However, we can bound the time-complexity of a naive algorithm (such as Bron–Kerbosch) in terms of the maximum number of homology edges of any vertex in G . Let γ be the maximal clique size in $\text{Hom}(G)$. Then $\gamma^* \leq \eta^*$.

We can readily solve MAX-CLIQUE(G) in $\mathcal{O}(|V|^{\eta^*}(\eta^*)^3)$ time by enumerating all $|V|^\gamma$ subgraphs of size γ for every $\gamma = 1, 2, \dots, \eta^*$.

Note that a subset of $\text{Hom}(G'')$ nearly defines an isomorphism between two subgraphs of $(V', \text{Int}(G''))$. For a pair (u, v) where $(u, v) \in \text{Hom}(G')$, $\text{Int}(u)$ may not exactly equal $\text{Int}(v)$, but $\text{Int}(u)/H = \text{Int}(v)/H$, where $\text{Int}(u)/H$ is some kind of factor graph in the algebraic sense that I'll need to define (look up factor group or quotient ring).

Extending to PPIs

Scoring of interactions

Implementation

Results

Overview

Accuracy

Speed

Appendix 1. Formal proofs