Dense Subgraphs with Restrictions and Applications to Gene Annotation Graphs

Barna Saha*, Allison Hoch**, Samir Khuller***, Louiqa Raschid†, and Xiao-Ning Zhang‡

Abstract. In this paper, we focus on finding complex annotation patterns representing novel and interesting hypotheses from gene annotation data. We define a generalization of the densest subgraph problem by adding an additional distance restriction (defined by a separate metric) to the nodes of the subgraph. We show that while this generalization makes the problem NP-hard for arbitrary metrics, when the metric comes from the distance metric of a tree, or an interval graph, the problem can be solved optimally in polynomial time. We also show that the densest subgraph problem with a specified subset of vertices that have to be included in the solution can be solved optimally in polynomial time. In addition, we consider other extensions when not just one solution needs to be found, but we wish to list all subgraphs of almost maximum density as well. We apply this method to a dataset of genes and their annotations obtained from The Arabidopsis Information Resource (TAIR). A user evaluation confirms that the patterns found in the distance restricted densest subgraph for a dataset of photomorphogenesis genes are indeed validated in the literature; a control dataset validates that these are not random patterns. Interestingly, the complex annotation patterns potentially lead to new and as yet unknown hypotheses. We perform experiments to determine the properties of the dense subgraphs, as we vary parameters, including the number of genes and the distance.

1 Introduction

Biological knowledge is increasingly being represented using graphs, e.g., protein interactions, metabolic pathways, gene regulation, gene annotation, etc. Finding

* Research supported by NSF Award CCF-0728839. Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail : barna@cs.umd.edu.

** Research supported by NSF REU Supplement to Award CCF-0728839. Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail : allie@umd.edu.

*** Research supported by NSF Award CCF-0728839 and a Google Research Award. Department of Computer Science and UMIACS, University of Maryland, College Park, MD 20742. E-mail : samir@cs.umd.edu.

† Research supported by NSF Award IIS-0430915 and IIS-0960963. UMIACS and Robert H. Smith School of Business, University of Maryland, College Park, MD 20742. E-mail : louiqa@umiacs.umd.edu.

‡ Research supported by Department of Biology, St. Bonaventure University, St. Bonaventure, NY 14778 and Department of Cell Biology and Molecular Genetics, University of Maryland, College Park, MD 20742. Email: xzhang@sbu.edu

highly dense regions in graphs is a problem of both theoretical [17, 12, 3, 14] and practical importance. *Density* is one quantitative measure of the connectedness of a subgraph and is defined as the ratio of the number of induced edges to the number of vertices in the subgraph. Even though there are an exponential number of subgraphs, a subgraph of maximum density can be found in polynomial time [17, 12, 3]. In contrast, the *maximum clique* problem to find the subgraph of largest size having all possible edges is NP-hard; it is even NP hard to obtain any non-trivial approximation. Finding densest subgraphs with additional size constraints is NP hard [14]; yet, they are more amenable to approximation than the maximum clique problem. Moreover detecting only cliques can be somewhat restrictive, since interesting subgraphs missing a few edges are omitted by any such procedure.

In this paper, we apply the densest subgraph problem to the task of finding complex patterns in a *gene annotation graph* representing annotations of genes using terms from controlled vocabularies (CVs) or ontologies. We attempt to increase the biological meaning of subgraphs by favoring the inclusion of pairs of nodes that have a meaningful relationship within the ontology structure that was used to create the gene annotation graph; we do this by defining a distance metric d_H between pairs of nodes. The goal is to return dense subgraphs with vertices within the subgraph satisfying a distance threshold.

We introduce a new variant of densest subgraph problems in this paper, namely the *distance restricted* densest subgraph problem to capture this property. We are given a graph $G = (V, E)$ as well as a distance metric d_H defined over pairs of vertices $u, v \in V$. The goal is to return a maximum density subgraph $S \subseteq V(G)$, such that in S, any pair of vertices are within distance τ according to d_H.

Further, researchers may be interested in obtaining patterns containing prespecified nodes. We refer to this as the *subset maximum density problem*, and this is described in Section 3. Finding only *one* dense subgraph may not suffice since the researchers may wish to find many complex annotation patterns. Thus, we address the problem of *all* maximum and *nearly* maximum dense subgraphs with distance/ subset restrictions in Section 4. We are the first to introduce and study the problem of detecting distance and subset restricted densest subgraphs.

In computational biology, there has been a body of work closely related to detecting dense subgraphs. Most of these papers concentrate on protein-protein interaction networks, where the goal is to cluster the network to detect densely connected molecular modules [30, 15, 19, 24], that can possibly identify protein families and molecular complexes [4, 1], or even identify missing interactions [32] and annotations [21]. Work by Newman [22] studies community detection in metabolic and regulatory networks. Communities are characterized by high intra and sparse inter connectivity. Many of these works on community detection can benefit by application of distance restricted dense subgraphs problem and its extensions.

The works of [1, 19] consider *clustering coefficients* for a measure of density on the neighborhood of each node. It is defined as the ratio of edges among the

neighbors to the maximum possible number of edges. Thus an alternative measure for density can be to compute clustering coefficient of the entire subgraph. However it tends to find very small subgraphs and is not effective.

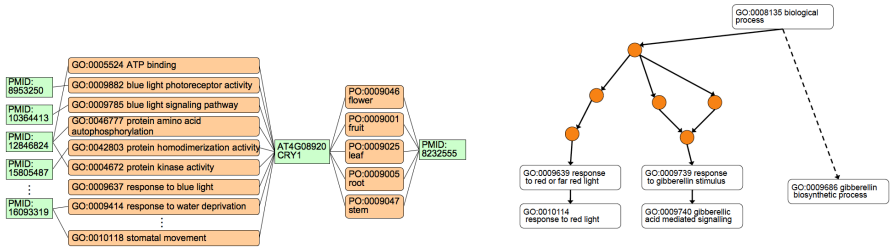
1.1 Gene Annotation Data

Knowledge about genes has been captured in publicly available bibliographic resources such as PubMed [27, 6] and PubMed Central [28, 6], general purpose resources such as Entrez Gene [5, 20], and in more focused model organisms or domain specific collections such as The Arabidopsis Information Resource (TAIR) [8, 16, 9]. In order to improve interoperability, various communities have created a number of ontologies such as the Gene Ontology (GO) [7, 11], the Plant Ontology (PO) [26], and the Unified Medical Language System (UMLS) [2, 31]. Data entries (records) in a resource are typically annotated with concepts or controlled vocabulary (CV) terms from one or more of these ontologies, creating a rich Web of annotation knowledge.

We focus on The Arabidopsis Information Resource (TAIR) [8, 16, 9]. A scientist can typically visit a page that provides a rich synopsis of a TAIR gene and then follow links to reach genotype and phenotype annotation data, publications, organism specific data, ESTs, pathway data, etc. Annotations in TAIR are associated with explanations or evidence codes reflecting the underlying methodology supporting the annotation. While TAIR is a valued and much visited portal that fuels the progress of scientific research, it also requires that scientists spend many hours manually clicking through web pages and following links, to create a subset of annotation knowledge for pattern discovery. Scientists often use simple tools such as a spreadsheet to maintain this subset of annotation knowledge. Increasingly, there is a need for more sophisticated tools to help the scientist integrate, analyze and visualize this knowledge.

We illustrate this using an example tool for integrating TAIR annotation data. The *LSLink* system [18] can be used to specify a protocol to create a background *LSLink* dataset of hyperlinked data records and their annotations. The protocol follows hyperlinks from each TAIR gene, and integrates the corresponding GO annotations, PO annotations, and the publications in PubMed that support the annotation. Some sample output of the integration protocol and the *LSLink* dataset is illustrated in Fig. 1(a) where we visualize the annotations for gene `CRY1`. The GO annotations are on the left side and the PO annotations on the right. Each includes the identifier and the label for the Controlled Vocabulary (CV) term. In addition, the figure includes the PubMed publications that support the annotations. As of January 2009, there were 17 GO annotations and 5 PO annotations for `CRY1`. The figure illustrates only some of the annotations (due to lack of space).

The *LSLink* annotation dataset of Fig. 1(a) represents knowledge culled from multiple research projects and their accompanying publications. The challenge for the scientist is to mine these datasets to discover important patterns. Consider the gene `GA3OX1` and a simple pattern of a pair comprising the GO CV term `gibberellic acid mediated signaling` and the PO CV term `germination`;



(a) Associations between GO and PO CV Terms Gene CRY1

(b) Semantic relationships in Gene Ontology GO

Fig. 1. TAIR

it is meaningful since `GA3OX1` regulates seedling growth. While these simple patterns are somewhat interesting, in order to capture biological knowledge, the scientist would be interested in finding a more complex pattern. Identifying a complex pattern in the annotations of a single gene may be non-trivial for a gene such as `CRY1` which has many annotations since the scientist has to consider many pairs of annotations and many groups of CV terms. However, the real challenge is even more difficult. While a pattern comprising a group of GO and PO terms annotating a single gene may correspond to a meaningful biological phenomenon, it may not be an interesting discovery. This is because it is annotating a single gene and the knowledge may be well known. A truly interesting discovery of knowledge that is as yet unknown, typically would require that the scientist solve the greater challenge of finding a pattern of a group of PO terms and GO terms that annotated *multiple genes*. Identifying such a co-occurrence pattern for a group of as yet unrelated genes can lead to the *gold standard* of an interesting discovery that would lead to actionable hypothesis, e.g., an experiment to verify the pattern.

The second challenge is that the GO and PO terms that form a pattern are not independent but they occur within a (hierarchical) ontology structure. Controlled vocabulary (CV) terms that are closer to each other in the hierarchy may be more closely related in meaning. Consider the fragment of the GO hierarchy of Fig. 1(b). This fragment illustrates some of the GO terms that annotate the TAIR gene `GA3OX1`. The labeled rectangular nodes annotate the gene while the circular nodes are placeholder GO CV terms in the ontology that do not annotate `GA3OX1`. We note that the following 2 terms, `response to gibberellin stimulus` and `gibberellic acid mediated signaling`, are more closely related whereas the pair of terms, `response to red light` and `gibberellin biosynthetic process` may appear to be unrelated. A complex pattern that included the first pair is more likely to be meaningful in comparison to a complex pattern that included the second pair. Two nodes in the ontology graph that have a smaller shortest path distance are assumed to be more closely

related and therefore more biologically meaningful, compared to a pair that are farther apart in the structure.

1.2 Gene Annotation Graph and Notion of Density

We can formalize our problem as follows: We are given two ontologies, GO and PO and a collection of genes $\mathcal{G}$, that are associated with some subsets of the CV terms in the two ontologies. In other words, each gene is annotated (associated) with a set of GO and PO nodes as seen in Fig. 1(a). We can represent this data in the form of a bipartite graph $G = (A, B, E)$ between the set of GO nodes and the set of PO nodes. The bipartite graph is a weighted graph where each edge is labeled with a set of gene names, such that each gene is annotated with the corresponding GO and PO nodes.

Each CV term in the GO (or PO) ontology has a vertex representing it in A (or B). If there are t genes $g_1, g_2, .., g_t \in \mathcal{G}$ containing the CV terms corresponding to vertices $u \in A$ and $v \in B$ in their annotations, then an edge is added between u and v in G, with weight $w'(u, v) = t$. We will often refer to this bipartite graph G as the annotation graph. We note that while we illustrate our algorithms using this GO PO bipartite graph, our algorithm works equally well for general graphs.

If the set of genes of interest are richly annotated with GO and PO terms, then the scientist has to examine a large annotation graph G. Even a simple yet meaningful visualization of the annotation graph is non trivial. Our high level objective is to discover complex patterns involving multiple genes that are co-annotated with the same subset of GO and PO terms. One way to do this is to identify large cliques in bipartite graphs. To be more flexible in finding interesting patterns, we instead look for densest subgraphs that find a large set of genes sharing a lot of common GO and PO terms; at the same time we would like the GO and PO terms to be closely related leading to the distance restriction.

Another formulation may consider the genes and their annotations by GO, PO nodes as a hypergraph. GO and PO nodes correspond to vertices as before, but now each gene is a hyperedge consisting of a set of GO and PO nodes. One related notion of density in a hypergraph is the ratio of hyperedges completely contained in a subgraph to the number of vertices present in that subgraph. Our algorithms for finding maximum density subgraphs work with hypergraphs as well. However this formulation may not be very useful in our context. A set of GO and PO nodes may be shared by a few genes; if there is a gene that in addition includes another GO or PO node that was not chosen, then it will not be included. The detection of this last gene might provide valuable information by discovering a missing annotation; but the hypergraph approach may not detect it.

We further consider two extensions to the problem that will be of interest to the scientist. Finding a single densest subgraph may not help the scientist explore all the interesting patterns in the annotation knowledge. One extension is to find all densest subgraphs. Further, there may be subgraphs that have

density close to the maximum density that are also interesting, e.g., they include a different set of genes, or a different set of GO or PO terms, in comparison to the densest subgraph. Such diversity of subgraphs may also help the scientist discover interesting patterns. Thus, a natural generalization is to find all the subgraphs of density close to the maximum. We refer to this as the *all almost maximum density subgraph problem*. Finally, scientists might be interested in filtering the densest subgraphs so that they contain a specified subset of GO or PO CV terms that are of interest to the scientist. We call this *subset maximum density problem*.

Our main contributions are as follows:

- In Section 2 we give a formal definition of the distance restricted dense subgraph problem. We show that for general metrics the problem is at least as hard to approximate as the well known independent set problem, and for special metrics such as trees and interval graphs it can be solved optimally in polynomial time. In addition, if we are willing to relax the distance threshold slightly we can solve it in polynomial time.
- In some cases there is a subset of GO and PO nodes that are to be studied, and we are specifically looking for subgraphs that contain these nodes. In Section 3 we show that the problem when a specified subset of vertices must be part of the subgraph can also be solved optimally in polynomial time as well.
- In Section 4 we show how Picard and Queyrannes's framework [25] (developed to find a compact encoding of all s-t min-cuts) can be adapted to find a collection of subgraphs whose density is close to the density of the maximum density subgraph. This framework can also be trivially extended for the generalizations we mentioned above (distance restricted subgraphs as well as the case when a subset of nodes must be part of the solution).
- Using a set of 10 photomorphogenesis genes and a set of 10 control genes, a user evaluation demonstrates that the *densest subgraph* for the photomorphogenesis genes returns many patterns that are validated by the literature. Further, the control genes validate that the results in the densest subgraph are not random patterns. Of more interest, we identified complex patterns of as yet not well known knowledge that could lead to new hypotheses. Results are reported in Section 5. We performed experiments on several other different set of genes and studied the properties of densest subgraphs and our algorithms on TAIR dataset. These additional results can be found in an extended version [29].

2 Distance Restricted Densest Subgraph Problem

In this section we are interested in the *distance restricted densest subgraph problem*. While our methods work for general graphs, in this framework we consider a bipartite graph $G = (A, B, E)$ with two disjoint sets of vertices A and B, and a set of edges E. We are also given a distance function (say a metric) d_A (d_B) that specifies distances between pairs of nodes in set A (B). In addition, we

are given distance thresholds τ_A, τ_B. The goal is to compute a densest subgraph $G_S = (S_A, S_B, E_S)$ by choosing subsets $S_A \subset A$ and $S_B \subset B$ to maximize the density of the subgraph, which is defined as $\frac{w'(E_S)}{|S_A|+|S_B|}$. Here $w'(E_S)$ denotes the weight of the edges in the subgraph induced by E_S. In addition, we require that for all pairs of vertices $u, v \in S_A$ we have $d_A(u, v) \leq \tau_A$, and the same condition holds for pairs of vertices in S_B, namely that for all $x, y \in S_B$ we have $d_B(x, y) \leq \tau_B$. Here G represents the annotation graph, A and B correspond to GO and PO nodes respectively. Distance function d_A (d_B) comes from the shortest path metric of the GO (PO) ontology graph.

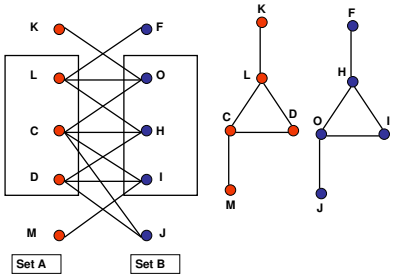


Fig. 2. An example of a bipartite graph $G = (A, B, E)$ (left). GO and PO graphs are shown on the right.

Consider the example in Fig. 2. We set $\tau_A = \tau_B = 1$ (distance is defined by the shortest path metric).The densest subgraph that satisfies the distance constraints is as shown - formed by the nodes $S_A = \{L, C, D\}$ and $S_B = \{O, H, I\}$. The number of edges is 7 in this induced graph giving a density of $\frac{7}{6}$. Note that the subgraph obtained by adding node J to G_S would have a higher density of $\frac{9}{7}$, but we cannot add J to the subgraph since $d(H, J) > \tau_B$. The proof of the following theorem is omitted for lack of space and can be found in the extended version [29].

Theorem 1. *When the distance function is an arbitrary metric, the problem is NP-hard and at least as hard to approximate as the maximum independent set problem [10].*

The relationship with the independent set problem explains why this problem is hard to approximate and it is not possible to develop approximation algorithms with good performance guarantee for this problem for general metrics. However, we next show that for many family of graphs this problem can be solved exactly. We identify a generic property of a metric, such that if the metric satisfies this property then we can solve the problem optimally in polynomial time.

2.1 Polynomial Time Algorithms for the Distance Restricted Densest Subgraph Problem

Let $G = (A \cup B, E)$ and d_A and d_B be the two metrics. Let $S_A \subseteq A$ and $S_B \subseteq B$ form a densest subgraph in G such that any two vertices $u, v \in S_A, d_A(u, v) \leq \tau_A$, for a given value τ_A (and similarly for $x, y \in S_B, d_B(x, y) \leq \tau_B$).

For our specific problem, we encode the distance function between pairs of nodes in A and B by the shortest path distance in a given graph H. H will have two components, one for A and one for B. Let n be the number of nodes in G. The high level idea is as follows: we wish to select a polynomial collection of subgraphs $G_i = (A_i, B_i, E_i)$ for $i = 1 \ldots p(n)$, and for each G_i compute the densest subgraph. The computed G_i's satisfy the following two properties:

- **Distance property**: All pairs of nodes in A_i (B_i) satisfy the pairwise distance constraint.
- **Subset property**: There exists some G_i that contains the true optimum solution G_S.

Since we find the densest subgraph within each G_i, we are guaranteed to find G_S. Thus a polynomial time algorithm for extracting G_i's give a sufficient condition for the existence of a polynomial time algorithm for the *distance restricted densest subgraph* problem on G. To obtain the densest subgraph within each G_i, we will use the procedure **Find-Dense-Subgraph**(G_i) (described later). The worst case running time involves $p(n)$ calls to an algorithm for computing the densest subgraph, which in turn requires $O(\log n)$ calls to a a min-cut/max-flow algorithm (worst case $O(n^3)$). This gives rise to a polynomial time algorithm, albeit with a rather high polynomial complexity. Luckily, in practice, the subgraphs we run the computation on are significantly smaller than the entire graph, so the algorithm runs fairly quickly.

We now show how to generate a polynomial collection of subgraphs G_i's satisfying the two properties: distance and subset property. Let z be a small constant. Consider every subset Y_p of A such that $|Y_p| = z$, and for all $t'_A \leq \tau_A$ let A_p be defined as $\{v \in A | \forall r \in Y_p \ d_A(v, r) \leq t'_A\}$. Similarly we define a collection of subsets B_q: Consider every subset Z_q of B such that $|Z_q| = z$, and $t'_B \leq \tau_B$, let B_q be defined as $\{v \in B | \forall r \in Z_q \ d_B(v, r) \leq t'_B\}$. We generate subgraphs defined by every A_p and B_q pair.

First note that, since we are considering every subset $Y_p \subseteq A$ and $Z_q \subseteq B$: the *subset* property holds; namely that one of these subgraphs is guaranteed to contain the optimal subset of nodes. The main difficulty is to show the distance property, that is to show the pairwise distance between every pair of nodes in A_p is at most t'_A and the pairwise distance between every pair of nodes in B_q is at most t'_B. Now, we exhibit some classes of graphs for which the distance property holds.

Tree Metric. Let T_A, T_B be the trees for A and B respectively in H. The distance in $T_A(T_B)$ induces the metric $d_A(d_B)$.

In this case we need $z = 2$ (recall that z is the cardinality of Y_p and Z_q). Choose two vertices a and b from T_A of distance, say t'_A. and two vertices c, d from T_B of distance t'_B. Define $Y_p = \{a, b\}$ and $Z_q = \{c, d\}$. Obtain the sets A_p and B_q as described in the previous subsection. Construct a subgraph induced on the vertex sets A_p, B_q and obtain the densest subgraph. Return the densest subgraph obtained from all of these subgraphs by making all possible choices for $\{a, b\}$ and $\{c, d\}$.

Now, we prove that the above algorithm (call it **Tree-Densest-Subgraph**) produces an optimum solution, by showing that the distance property holds.

Theorem 2. Tree-Densest-Subgraph *gives an optimum solution, when d_A and d_B form a tree metric.*

Proof. We only need to show the distance property, that is all the vertices chosen in A_p have pair-wise distance $\leq t'_A$ and all the vertices chosen in B_q have pair-wise distance $\leq t'_B$. Pick any two arbitrary vertices $x, y \in A_p$. Therefore they are both at distance at most t'_A from a and b. Let the path from x to a, $P_{x,a}$ intersect $P_{a,b}$ at c_1 and similarly the path from y to a intersect $P_{a,b}$ at c_2. Let $d(x, c_1) = d_1$ and $d(y, c_2) = d_2$. Without loss of generality, assume, c_1 is closer to a than c_2. Let $d(a, c_1) = r_1$, $d(c_1, c_2) = r_2$, $d(c_2, b) = r_3$. Hence the distance between x and y is $d(x, y) = d_1 + r_2 + d_2$. We have the following sets of equations, $d_1 \leq r_1$, otherwise x, b is a furthest pair. Similarly, $d_2 \leq r_3$, otherwise (a, y) is a furthest pair. Hence $d_1 + r_2 + d_2 \leq r_1 + r_2 + r_3 \leq t'_A$. Therefore, all the vertices chosen from A_p satisfies the distance threshold. Same argument works for vertices chosen from B_q. Thus the distance property is established. □

Some Other Distance Metrics. The same approach can be extended for graphs where each edge can participate in at most one or two cycles and the problem can be solved optimally in polynomial time. In general it may be possible to extend this approach to graphs where each edge participates in constant number of cycles. The proof technique is similar to the case of trees. Another class of graphs for which we can obtain polynomial time algorithm is *interval* graphs. The proofs of these results can be found in an extended version [29].

2.2 Generalization to arbitrary graphs

For general graphs, it is not possible to obtain an exact polynomial time algorithm. Here we describe two methods that we implemented. The first method guesses a vertex $a \in G_S$ from GO ($b \in G_S$ from PO) and selects all the vertices within distance say $\frac{t'_A}{2}$ ($\frac{t'_B}{2}$) of a(b). Suppose that the set of vertices are denoted by X_a(X_b). We now run the algorithm **Find-Dense-Subgraph**($X_a \cup X_b$). This ensures that the vertices are all close to each other, but we may not find the densest subgraph due to the shorter distance requirement.

The second method is identical except that we guess a node a from GO and a node b from PO and select all the vertices within distance say t'_A(t'_B) of a (b). Now clearly, $V(G_S) \subseteq X_a \cup X_b$ and any two vertices in X_a have distance at most

$2t'_A$ and any two vertices in X_b have distance at most $2t'_B$. Thus if the optimum solution has density d_S with distance threshold t, then we guarantee obtaining a subgraph with density at least d_S and distance at most $2t$.

3 Densest Subgraphs with a Specified Subset

In this section, we describe the densest subgraph algorithm, where a subset of GO and PO nodes are given apriori and must appear in the returned solution. A distance threshold may also be specified. In that case, we force the subset of nodes that must appear in the solution, into G_i and obtain the rest of the vertices by proper guessing as has been shown in the previous section. Thus in this section, we just consider the problem of finding a densest subgraph of a graph when a subset of nodes must appear in the solution.

Given a graph $G = (V, E)$ and a weight function on the edges w' and a weight function on the vertices w, and a subset C of vertices, we wish to compute a densest subgraph that contains C. The density of a subgraph is defined as the ratio of the total weight of the edges in the induced graph, to the weights of the nodes in the subgraph (in the unweighted case, all weights are 1). If S is a subset of nodes then $E(S)$ is the subset of edges in the subgraph induced by S. Let $w'(E(S)) = \sum_{e \in E(S)} w'(e)$. For a node c, let $w'(S, c) = \sum_{(x,c)|x \in S} w'(x, c)$. Let $\overline{E}(S)$ be the set of edges incident to nodes in S for any $S \subset V$.

We first contract all the nodes in C to a single node c. We define $w(c) = \sum_{i \in C} w(i)$. All the edges between nodes in C become a self loop on c with $w'(c) = \sum_{(i,j) \in E(C)} w'(i, j)$.

In other words, we wish to compute a subset $S \subset V \setminus \{c\}$ such that we maximize the following ratio: $\frac{w'(E(S))+w'(c)+w'(S,c)}{w(S)+w(c)}$.

3.1 Algorithm for Densest Subgraph without a specified subset

We first discuss the basic algorithm for finding a densest subgraph by a series of max-flow (min-cut) computations [17]. This is the procedure **Find-Dense-Subgraph** mentioned earlier. We guess α, the density of the maximum density subgraph and then refine our guess by doing a network flow computation. Suppose a subset S^* exists with density α^* and this is the maximum density subgraph. Suppose our guess is α. By appropriately defining a flow network and by examining its min-cut structure we are able to determine if $\alpha = \alpha^*$, or $\alpha < \alpha^*$ or $\alpha > \alpha^*$. It is very easy to start the binary search since we have upper and lower bounds on the optimal density α^* and since all densities are rational numbers, once the interval size drops to below $\frac{1}{|V|^2}$ we can stop.

We next describe the flow network that is constructed. Create a flow network G' with a source s and sink t. We have a node corresponding to each edge in G (call this set E') and a node corresponding to each node in G (call this set V'). Add edges from s to $e \in E'$ of capacity $w'(e)$ and an edge from $v \in V'$ to t with

capacity $\alpha w(v)$. Add edges from $e = (x, y) \in E'$ to both $x \in V'$ and $y \in V'$ with capacity ∞ [1].

If $C = \emptyset$ then the construction proceeds as follows (original problem). First note that there is a s-t min-cut of value $w'(E)$. Suppose the max density subset has density α^*. Suppose our guess $\alpha < \alpha^* = \frac{w'(S^*)}{w(S^*)}$, then it follows that $\alpha w(S^*) < w'(S^*)$.

Now consider an s-t cut $(s \cup V_1, t \cup V_2)$ in the flow network G', then let $S = V_1 \cap V'$. The cut includes all the edges from nodes in S to t of capacity $\alpha w(S)$ as well as edges from s to nodes in E' that are not in V_1. All edges $e = (x, y) \in E$ with one end in $V \setminus S$ must be in V_2 since otherwise there will be an edge of ∞ capacity across the cut. All the edges in the induced graph formed by S must be in V_1, since otherwise we can reduce the capacity of the cut. The weight of this cut is exactly $w'(E \setminus E(S)) + \alpha w(S)$. Note that $w'(E \setminus E(S))$ includes the weight of all edges that are incident on some node in $E \setminus S$. The weight of this cut is exactly $w'(E \setminus E(S)) + \alpha w(S) = w'(E) - w'(S) + \alpha w(S) = w'(E) - (w'(S) - \alpha w(S))$. But for the optimal subset S^*, we have $w'(S^*) - \alpha w(S^*) > 0$ thus there is a cut of value $< w'(E)$. So if this happens we know that our guess for α is $< \alpha^*$. Similarly, when our guess for α is $> \alpha^*$ then the (unique) min-cut has value $w'(E)$. When we make the correct guess, then there are multiple min-cuts of value $w'(E)$. Any min-cut other than the trivial gives the correct solution.

3.2 Algorithm for Densest Subgraph with a specified subset C

We now show how to modify this construction when $C \neq \emptyset$. We create a new source s' and add an edge to s with capacity $w'(E) - \alpha w(c)$. We also remove c from V'. Again suppose that $\alpha < \alpha^*$. In this case, a subset S^* exists such that $\frac{w'(E(S^*)) + w'(c) + w'(S^*, c)}{w(S^*) + w(c)} > \alpha$. Thus, $w'(E(S^*)) + w'(c) + w'(S^*, c) - \alpha(w(S^*) + w(c)) > 0$. Rreplace $w'(E) - w'(\overline{E}(V \setminus (S^* \cup c)))$ for the first term. (Note that $w'(\overline{E}(V \setminus (S^* \cup c)))$ includes all edges incident on nodes in $V \setminus (S^* \cup c)$, and not only the edges induced by those nodes). We now obtain:

$$w'(E) - w'(\overline{E}(V \setminus (S^* \cup c))) - \alpha(w(S^*) + w(c)) > 0.$$

$$(w'(E) - \alpha w(c)) - (w'(\overline{E}(V \setminus (S^* \cup c))) + \alpha w(S^*)) > 0.$$

$$(w'(E) - \alpha w(c)) > w'(\overline{E}(V \setminus (S^* \cup c))) + \alpha w(S^*).$$

This means that a min-cut exists (defined by the subset S^* for example) that is smaller than $w'(E) - \alpha w(c)$.

So again by looking at the min-cut structure we should be able to know that $\alpha < \alpha^*$. If $\alpha > \alpha^*$ then the trivial min-cut separating s' from the rest of the graph is unique. A binary search for α can be done.

Side Note: A simple method that will *not* work is to snap the edges to c as self loops and to then compute the densest subgraph in G with c removed. If

[1] If E' is a set of hyper-edges then we add such edges from e to all $x \in V'$ such that $x \in e$.

the density of the densest subgraph found is lower than $w'(c)/w(c)$ then we just return C as the answer. Otherwise we return $S \cup C$. The main problem is that the density of S could get lowered when we merge with C. The level of dilution depends on the size of the densest subgraph in G with c removed; hence a subgraph with slightly lower density than the optimal solution, but of much larger size could be a better choice.

4 Finding All almost Maximum Densest Subgraphs

In this section, we describe an algorithm for computing all densest subgraphs as well as all almost maximum densest subgraphs. It might not be sufficient just to find only one subgraph of highest density, and sometimes subgraphs having density close to the maximum might be interesting as well. If α^* is the highest density, our goal is to find all subgraphs that have density close to α^*. A subgraph S that has density $\alpha^*(1-\delta_S)$ is lacking by a factor of $(1-\delta_S)$ from the optimum. Thus if we want to detect it, we have to relax the density requirement of S by a $(1-\delta_S)$ factor. The amount of relaxation may differ depending on the size of the returned subgraph. We denote by $D(S)$ the density of the subgraph induced by S.

Formally, given a graph $G = (V, E)$, if $\alpha^* = \max_{S \subseteq V} D(S)$, then given an $\epsilon > 0$, we want to return $T = \{S \mid S \subseteq V, D(S) \geq \left(1 - \frac{\epsilon}{|S|}\right) \alpha^*\}$. Therefore, we have $\delta_S = \frac{\epsilon}{|S|}$. We consider the unweighted case, where vertices and edges all have unit weights. Extension to arbitrary weight is trivial. Also we can pose the distance restriction as in Section 2 easily and get the same approximation results as we obtained earlier.

Recall the construction of flow network from Section 3. We guess α as the value of density and create a flow network $N(G)$ for graph G. If $\{s \cup V_1, t \cup V_2\}$ is the minimum cut and $V_1 \wedge V = S$, then the value of min-cut is $K = |E| - (E(S) - \alpha|S|)$. Thus when, $\alpha = \alpha^*$, $K = |E|$. The algorithm searches for the value of α^* using a binary search. Since the gap between two consecutive density values, is at least $\frac{1}{|V|^2}$ [12], the value of α^* can be guessed accurately in $O(\log n)$ time.

We construct the flow network N_{α^*} with α^* as the guess, and compute all min-cuts having value $\leq |E| + \epsilon\alpha^*$. There are two questions, "how can we compute all min-cuts of value $\leq |E| + \epsilon\alpha^*$?" and "how can T be detected from these min-cut computations ?". While we address the first question in Subsection 4.1, following lemma answers the second.

Lemma 1. *Let* $M = \{V_1 \mid cut(s \cup V_1, t \cup (V \setminus V_1)) \leq |E| + \alpha^*\epsilon\}$, *then* $T = \{V_1 \cap V\}$.

Proof. Let $S' \in T$ and $S' = V_1' \wedge V$. Then the cut induced by $s \cup V_1'$ is $|E| - (E(S') - \alpha^*|S'|) = |E| - |S'|(D(S') - \alpha^*)$. Since $S' \in T$, $D(S') \geq \alpha^*(1 - \frac{\epsilon}{|S'|})$. Thus, the cut induced by $s \cup V_1'$ is at most $|E| - |S'|(\alpha^*(1 - \frac{\epsilon}{|S'|}) - \alpha^*) = |E| + \alpha^*\epsilon$. Hence, $V_1' \in M$. On the other hand, if $V_1' \in M$, then the cut value of $s \cup V_1'$

is, $|E| - (E(S') - \alpha^*|S'|) \leq |E| + \alpha^*\epsilon$. Thus, $\alpha^*|S'| - E(S') \leq \alpha^*\epsilon$, or $D(S') \geq \alpha^*(1 - \frac{\epsilon}{|S'|})$. □

Now we show how by modifying Picard and Queyranne's algorithm [25], we can compute all cuts of value $\leq |E| + \epsilon\alpha^*$ in N_{α^*}.

4.1 Finding All Almost Min-Cuts

In Picard's algorithm, we are given a finite directed network $N = (V, E, c)$, with vertex set V, including a source s and a sink t, arc sets E and positive capacities $c_{i,j}$ defined on every $(i, j) \in E$. The goal is to compute all s-t cuts having minimum value. Given a binary relation R on V, a subset $C \subseteq V$ is said to be closure for R, iff for all vertices $i, j \in V$, the conditions $i \in C$ and iRj imply $j \in C$. Picard showed that, if f is a maximum flow in N, $cres$ is the residual capacity and R is defined as, iRj, iff $cres(i, j) > 0$, then a cut $(S, \bar{S})$ separating s from t is a minimum cut iff S is a closure for R containing s and not t. By enumerating all closures of R, all the min-cuts can now be detected.

We define the relation R as, iRj iff $cres(i, j) > \epsilon\alpha^* = \delta$ instead of $cres(i, j) > 0$. The following lemma connects all almost s-t cuts with the closures for R.

Theorem 3. *All s-t cuts of value $\leq K + \delta$ are closures for R, where K is the value of minimum s-t cut.*

Proof. Consider a cut S of value $K' \leq K + \delta$. Let the edges across the cut $E(S, \bar{S}) = \{e_1, e_2, \ldots, e_l\}$, $e_i \in E(G)$ with capacities $\{c_1, c_2, \ldots, c_l\}$ and flow $\{f_1, f_2, \ldots, f_l\}$. Since maxflow is equal to min-cut, when we consider the max flow in the network, we must have, $f(S, \bar{S}) - f(\bar{S}, S) = K$. Hence $f(S, \bar{S}) > K$. Let if possible one of the residual capacity, say of e_1 be higher than δ, then that will imply S is not a closure for R. We have $K < f_1 + f_2 + \ldots + f_l < (c_1 - \delta) + c_2 + \ldots + c_l = K' - \delta$. So $K' > K + \delta$, giving a contradiction. □

Therefore, we again enumerate all the closures, and discard any closure for which cut value is $> K + \delta$. This last step is necessary, since there can be some closures for R that do not necessarily give a cut of value $\leq K + \delta$. The closures for R contain all the cuts of value $\leq K + \delta$ and some cuts of value $K + \delta(l + l')$.

5 Experiments on the TAIR Dataset

We briefly summarize the results of several experiments. In a first experiment (dataset SD_1) we analyze 10 photomorphogenesis genes. We use the literature to validate patterns identified in a dense subgraph. We highlight some interesting patterns that are novel and could lead to new hypotheses. A control experiment (dataset SD_2) includes the 10 photomorphogenesis genes and 10 additional control genes. The control experiment is used to confirm that all patterns identified in the dense subgraphs are true positives and are validated in the literature.

There were no false positive patterns identified in our experiment. In a subsequent experiment (dataset SD_3), we analyze 20 genes involved in different (and currently unrelated) biological pathways. We also perform experiments to study the properties of the dense subgraphs, for different experiment protocols and parameters such as the number of genes and the GO and PO distance thresholds. These results are in an extended report [29].

We execute a protocol to retrieve all TAIR genes, their GO and PO annotations, and the reference publications from PubMed. As of January 2009, the LSLink TAIR dataset contains 3540 GO CV terms, 350 PO CV terms, 18861 genes, 70128 GO annotations, 484261 PO annotations and 1873250 (GO, PO) pairs. The average number of GO annotations and PO annotations for the TAIR genes is 3.97 and 3.13, respectively. The maximum number of annotations for any gene is 22 GO annotations and 50 PO annotations.

5.1 Photomorphogenesis Case Study

We report on promising results of a photomorphogenesis case study on dataset SD_1 with the following 10 TAIR genes: `CRY1, CRY2, HFR1, CIB1, CIB5, SHB1, COP1, HY5, PHOT1, PHOT2`. These 10 genes are associated with 107 annotations (66 GO terms and 41 PO terms) and 2230 combinations of (GO, PO) terms. The edge weight from the corresponding bipartite graph ranged from a minimum of 1 (1368 edges) to a maximum of 7 (2 edges). We applied the *distance restricted dense subgraph* algorithm with GO distance threshold of 2 and PO distance threshold of 3, which identified a complex pattern involving the following subset: 9 genes, `CRY1, CRY2, HFR1, CIB5, COP1, HY5, PHOT1, PHOT2, SHB1` 3 GO terms, `5634: nucleus; cellular_component`, `5773: vacuole; cellular_component`, and `5794: Golgi apparatus; cellular_component`; and 13 PO terms. These obtained GO and PO terms are shown in Figure 3. Figure 3 also shows a subgraph chosen from the densest subgraph involving 2 GO terms, `5634: nucleus; cellular_compo- nent` and `5773: vacuole; cellular_component`; and 2 PO terms, `13: cauline leaf; plant_structure` and `37: shoot apex; plant_structure` are shown in Figure 3. This creates 4 (GO, PO) pairs as follows: (5634, 13); (5634, 37); (5773, 13); (5773, 37). Figure 3 also illustrates the genes that are annotated by these pairs.

We make the following observations:

- The combinations of (GO,PO) edges observed in this complex pattern are consistent with the literature and provides validation that the complex pattern is meaningful. Details of all the observations can be found in an extended version [29].
- Specific combinations of genes and (GO,PO) edges are interesting in that they can lead to further hypothesis. We identify 5 potentially interesting patterns from the subgraph. We elaborate on two patterns. Details can be found in an extended version [29].
- `HFR1` is not annotated with the following GO and PO combination: (`5634: nucleus; cellular_component` and `37: shoot apex; plant_structure`).

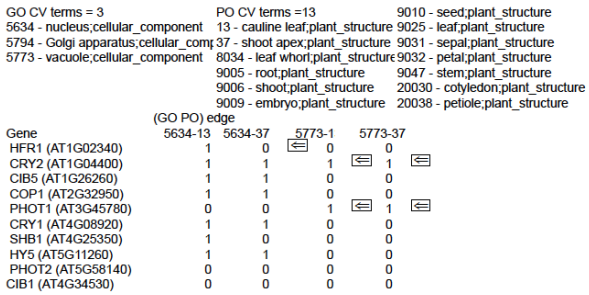


Fig. 3. Potential Complex Pattern of Photomorphogenesis Genes

This is indicated by an arrow in Figure 3. A review of the literature suggests that this is a novel observation about the mechanism controlling this gene that should be pursued further.

– The next observation confirms the potential benefits of our approach to finding complex patterns in the annotated *LSLink* datasets. Consider the pattern of annotation that includes the 2 genes `CRY2` and `PHOT1`. Both are annotated with the following 2 GO and PO combinations: (`5773: vacuole; cellular_component` and `13: cauline leaf; plant_structure`) and (`5773: vacuole; cellular_component` and `37: shoot apex; plant_structure`). These annotations are also marked with an arrow in Figure 3. We observe that there are only 2 papers in the literature, [23] published in 2004, and [13] published in 2008, that postulate that some members of the `CRY` and `PHOT` families may be functionally interactive in vacuoles. Indeed, these two papers came to this conclusion only after significant experimental research.

To summarize, our user evaluation confirmed the benefit of using the dense subgraph approach of identifying complex patterns based on the underlying patterns of annotation, without having to completely digest the scientific literature and/or complete an experiment protocol.

We performed a control experiment using SD_2; SD_2 included the 10 genes of SD_1 and 10 additional genes that were chosen randomly from genes that had some common annotations with the genes in SD_1. The goal of this experiment is to verify that the pattern emerged from experimenting on SD_1 alone still persists and thus to confirm that it was not a random pattern. The dense subgraph for SD_1 included 9 genes, 3 GO terms, 13 PO terms and 39 (GO, PO) edges. The dense subgraph for SD_2 included 14 genes, 4 GO terms, 11 PO terms and 44 (GO, PO) edges. The genes included the 8 photomorphogenesis genes (HFR1 CRY2 CIB5 COP1 PHOT1 CRY1 SHB1 HY5) and 6 control genes (GAPC2 FT ARF3 AG ARF4 REV). The gene PHOT2 is not included. Further, the GO term `Golgi apparatus` and 3 PO terms `cauline leaf`, `leaf whorl` and `petiole` were not present. Two GO terms `mitochondrion` and `cytosol` and 1 PO term `inflorescence meristem` were introduced. Detailed observations

from the control dense subgraph for SD_2 can be found in an extended version [29].

While the control dense subgraph for SD_2 does show some variations in terms of photomorphogenesis genes, GO and PO terms from that obtained using SD_1, we verified that none of these variations are significant, i.e., the variations do not contradict any of the patterns of annotation of the dense subgraph for SD_1. Further the patterns of SD_1, that were found validating the literature or can lead to potentially new hypothesis are unchanged. For example, `PHOT2` which is excluded from the control subgraph, as well as the GO and PO terms that are excluded were not included in any of the SD_1 patterns. An unexpected benefit is that the control densest subgraph for SD_2 was itself able to yield some interesting patterns that could lead to new hypotheses.

We note that developing a NULL hypothesis to test the significance of the dense subgraphs that we generate is non trivial since there are many metrics to compare the similarity of two graphs. One option is to add control genes as described. Other alternatives include comparing the density distribution of dense subgraphs from a random graph versus the dense subgraphs from our datasets. Another would generate *all almost dense subgraphs* to determine if they are different with respect to both metrics as well as the observed patterns. One may also consider random labeling of the GO and PO terms in the datasets. We will explore these alternatives in future work.

Additional experiments on different set of genes and empirical study on the properties of the densest subgraph algorithms can be found in an extended version [29].

Acknowledgments: We thank Carl Kingsford and Mihai Pop for useful discussions about our results and their feedback has been invaluable.

References

1. G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4, 2003.
2. O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database issue):267–270, 1 Jan 2004.
3. M Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, pages 84–95, 2000.
4. A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. 30(7):1575–1584, Apr 2002.
5. Entrez: the life sciences search engine. http://www.ncbi.nih.gov/gquery/gquery.fcgi.
6. E. W. Sayers et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 37(Database issue):D16–D18, 1 Jan 2009.
7. M. Ashburner et al. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.
8. Margarita et al. TAIR: a resource for integrated Arabidopsis data. *Functional and Integrative Genomics*, 2(6):239, 2002.
9. S. Y. Rhee et al. The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to arabidopsis biology, research materials and community. *Nucleic Acids Research*, 31(1):224–228, 1 Jan 2003.

10. U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
11. Gene Ontology (GO). http://www.geneontology.org/.
12. A. V. Goldberg. Finding a maximum density subgraph. Technical report, 1984.
13. B. Kang, N. Grancher, V. Koyffmann, D. Lardemer, S. Burney, and M. Ahmad. Multiple interactions between cryptochrome and phototropin blue-light signalling pathways in arabidopsis thaliana. *Planta*, 227(5):1091–1099, 2008.
14. Samir Khuller and Barna Saha. On finding dense subgraphs. In *ICALP '09*, pages 597–608, 2009.
15. A. D. King, N. Przulj, and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, Nov 2004.
16. S. Y. Rhee L. Reiser. *Using The Arabidopsis Information Resource (TAIR) to Find Information About Arabidopsis Genes*. Current Protocols in Bioinformatics. 2005.
17. E. Lawler. *Combinatorial optimization - networks and matroids*. Holt, Rinehart and Winston, New York, 1976.
18. W. Lee, L. Raschid, H. Sayyadi, and P. Srinivasan. Exploiting ontology structure and patterns of annotation to mine significant associations between pairs of controlled vocabulary terms. In *DILS 08*, pages 44–60, 25-27 June 2008.
19. X. Li, C. Foo, and S. Ng. Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. 6:157–68, 2007.
20. Donna R. Maglott, James Ostell, Kim D. Pruitt, and Tatiana Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 35(Database issue):26–31, 1 Jan 2007.
21. S. Navlakha, J. White, N. Nagarajan, M. Pop, and C. Kingsford. Finding biologically accurate clusterings in hierarchical tree decompositions using the variation of information. In *RECOMB*, pages 400–417, 2009.
22. M. E. J. Newman. Modularity and community structure in networks. 103(23):8577–8582, June 2006.
23. M. Ohgishi, K. Saji, K. Okada, and T. Sakai. Functional analysis of each blue light receptor, cry1, cry2, phot1, and phot2, by using combinatorial multiple mutants in arabidopsis. *PNAS*, 1010(8):2223–2228, 2004.
24. J. B Pereira-Leal, A. J Enright, and C. A Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*, 54(1):49–57, Jan 2004.
25. J.-C. Picard and M. Queyranne. On the structure of all minimum cuts in a network and applications. In *Mathematical Programming Study13*, pages 8–16, 1980.
26. Plant Ontology (PO). http://www.plantontology.org/.
27. PubMed. http://www.ncbi.nih.gov/entrez/.
28. PubMed Central. http://www.pubmedcentral.nih.gov/.
29. B. Saha, A. Hoch, S. Khuller, L. Raschid, and X. Zhang. Dense subgraph with restrictions and applications to gene annotation graphs. *http://www.cs.umd.edu/∼samir/grant/recomb-full.pdf*, 2010.
30. V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. 100(21):12123–12128, Oct 2003.
31. Unified Medical Language System (UMLS). http://www.nlm.nih.gov/research/umls/.
32. H. Yu, A. Paccanaro, V. Trifonov, and M. Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, Apr 2006.