Connectivity Measures for Signaling Pathway Topologies

Nicholas Franzese*[1,2,3], Adam Groce[2], T. M. Murali[3,4], and Anna Ritz†[1]

[1]Department of Biology, Reed College, Portland, OR, US
[2]Department of Computer Science, Reed College, Portland, OR, US
[3]Department of Computer Science, Virginia Tech, Blacksburg, VA, US
[4]ICTAS Center for Systems Biology of Engineered Tissues, Virginia Tech, Blacksburg, VA, US

1 Abstract

Characterizing cellular responses to different extrinsic signals is an active area of research, and curated pathway databases describe these complex signaling reactions. Here, we revisit a fundamental question in signaling pathway analysis: are two molecules "connected" in a network? This question is the first step towards understanding the potential influence of molecules in a pathway, and the answer depends on the choice of modeling framework. We examined the connectivity of Reactome signaling pathways using four different pathway representations. We find that Reactome is very well connected as a graph, moderately well connected as a compound graph or bipartite graph, and poorly connected as a hypergraph (which captures many-to-many relationships in reaction networks). We present a novel relaxation of hypergraph connectivity that iteratively increases connectivity from a node while preserving the hypergraph topology. This measure, B-relaxation distance, provides a parameterized transition between hypergraph connectivity and graph connectivity. B-relaxation distance is sensitive to the presence of small molecules that participate in many functionally unrelated reactions in the network. We also define a score that quantifies one pathway's downstream influence on another, which can be calculated as B-relaxation distance gradually relaxes the connectivity constraint in hypergraphs. Computing this score across all pairs of 34 Reactome pathways reveals two case studies of pathway influence, and we describe the specific reactions that contribute to the large influence score. Our method lays the groundwork for other generalizations of graph-theoretic concepts to hypergraphs in order to facilitate signaling pathway analysis.

2 Introduction

A major effort in molecular systems biology is to identify signaling pathways, the networks of reactions that link extracellular signals to downstream cellular responses. Computational representations of signaling pathways have increased in complexity, moving from gene sets to pairwise interactions in the past two decades [1]. Graphs are common representations of protein networks, where nodes are proteins and edges represent pairwise interactions between two proteins. While graph representations have been useful for pathway analysis [2–5] and disease-related applications [5–7], the limitations of graphs for representing biochemical reactions are well recognized [8–12].

Many pathway databases [13–20] have adopted reaction-centric signaling pathway formats such as the Biological Pathway Exchange (BioPAX) [21], which provides more mechanistic information about the interactions. As reaction-centric information has become available, many modeling frameworks have been proposed to overcome the limitations of graphs for analyzing signaling pathway structure [8, 9, 22, 23]. Compound graphs [24, 25] and metagraphs [8] aim to represent protein complexes and hierarchical relationships among molecular entities in the cell. Factor graphs [26] have been used to infer pathway activity from heterogeneous data types. Hypergraphs [27, 28] are generalizations of directed graphs that allow multiple inputs and outputs, and their realization as a model for signaling pathways is

*Current Affiliation: Department of Computer Science, University of Maryland College Park, College Park, MD, US
†Corresponding Author: aritz@reed.edu

emerging [9, 11, 29]. Other models such as Petri nets [30] and logic networks [31, 32] move away from structural network analysis and towards discrete dynamic modeling. Many of these modeling frameworks have an underlying bipartite graph structure.

These new representations have improved fidelity to the underlying biology of signaling reactions but also exhibit increased mathematical and algorithmic complexity. In this light, we examine a fundamental topological concept: when are two molecules "connected" in a signaling pathway? Defining and establishing connectivity is the first step to determining downstream or upstream elements of a molecule, which may indicate the influence of its activity or the effect of its perturbation. Connectivity is also central to computational methods for identifying potential off-target effects, determining pathway crosstalk, and computing portions of pathways that may be altered in disease.

We first begin by considering existing connectivity measures on four distinct representations of the Reactome pathway database [13, 14]. We demonstrate that these measures range from highly permissive (e.g., path-based connectivity in graphs) to very restrictive (e.g., connectivity in directed hypergraphs), depending on the representation. Thus, two molecules may be "connected" in one representation of a pathway and "disconnected" in another representation. We then introduce B-relaxation distance, a parameterized relaxation of connectivity that offers a tradeoff between the permissive and restrictive representations. We show that this new version of connectivity uncovers more subtle structures within the pathway topologies than previous measures, and is sensitive to the presence of small molecules that that participate in many reactions. We then consider 34 Reactome signaling pathways and use B-relaxation distance to capture the downstream influence of one pathway on another. B-relaxation distance allows us to gradually relax the connectivity constraints in hypergraphs, calculating pathway influence at each step. The graph representation of Reactome is too highly connected to enable the discovery of such relationships. We also show that the bipartite graph representation, although not as highly as connected as the graph, does not support this type of result. We describe two case studies of pathway influence that we recovered, and describe the specific reactions that contribute to the large influence score.

3 Results

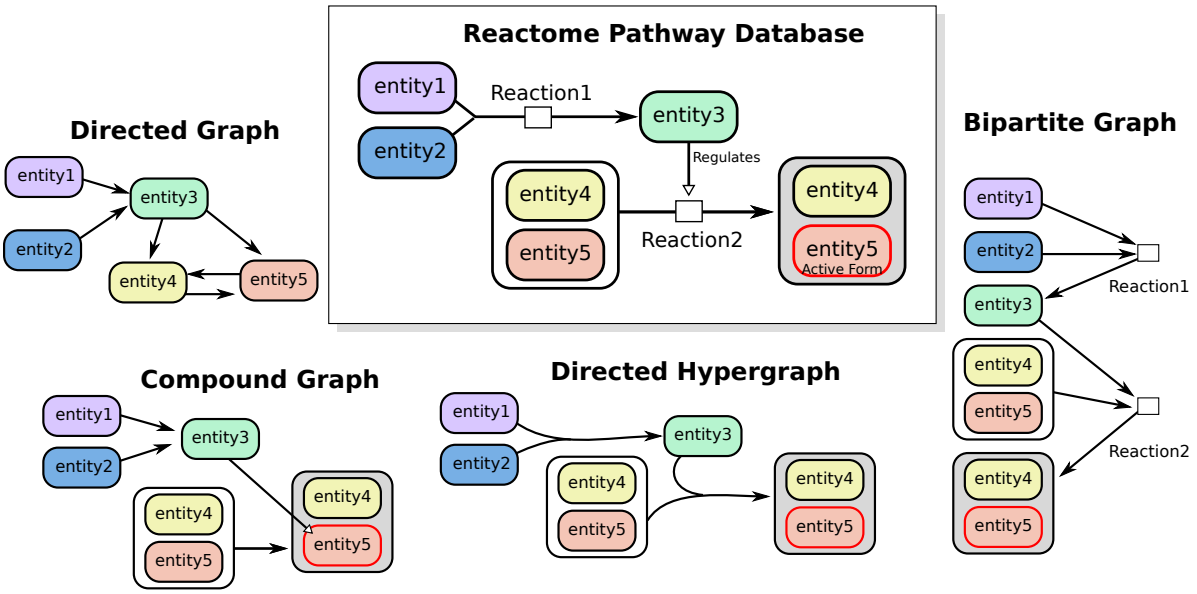

Fig 1. Representations of two toy reactions as directed graphs, compound graphs, directed hypergraphs, and bipartite graphs. In this work, we use "directed hypergraphs" and "hypergraphs" interchangeably.

3.1 Connectivity analysis using established traversal algorithms

We considered four established directed representations of signaling pathway topology and their associated measures of connectivity (Fig. 1). Directed graphs describe relationships among molecules (proteins, and small molecules),

while the other models describe relationships among entities that include proteins and small molecules, their modified forms, protein complexes, and protein families. Please refer to the Methods for full details about these representations, including how they are built.

1. **Directed graphs** represent molecules as nodes and interactions as pairwise edges. Interactions may be directed (such as regulation) or bidirected (such as physical binding). We use a Breadth First Search (BFS) traversal to find connected nodes.

2. **Compound graphs** represent interactions between pairs of nodes, which may be molecules or groups of molecules (e.g., protein complexes or protein families). We use a previously-established algorithm that traverses the BioPAX structure as a compound graph according to biologically meaningful rules [25].

3. **Bipartite graphs** contain two types of nodes: entity nodes and reaction nodes. Each biochemical reaction has an associated reaction node, whose incoming edges are connected to reactants and whose outgoing edges are connected to products. For each entity node, we use BFS to compute the set of connected entity nodes.

4. **Directed hypergraphs** represent reactions with many-to-many relationships, where each hyperedge $e = (T_e, H_e)$ has a set of entities in the tail T_e and a set of entities in the head H_e. We adopt a definition of connectivity called B-connectivity that requires all the nodes in the tail of a hyperedge to be visited before it can be traversed [28]. This definition has a natural biological meaning in reaction networks: B-connectivity requires that all reactants of a reaction must be present in order for any product of that reaction is reachable [11, 28].

	Directed Graph	Compound Graph	Bipartite Graph	Hypergraph
# Nodes	12,086	19,650	30,775	19,650
# Edges/Hyperedges	285,556	38,218	45,155	11,125

Table 1. Representations of the Reactome database.

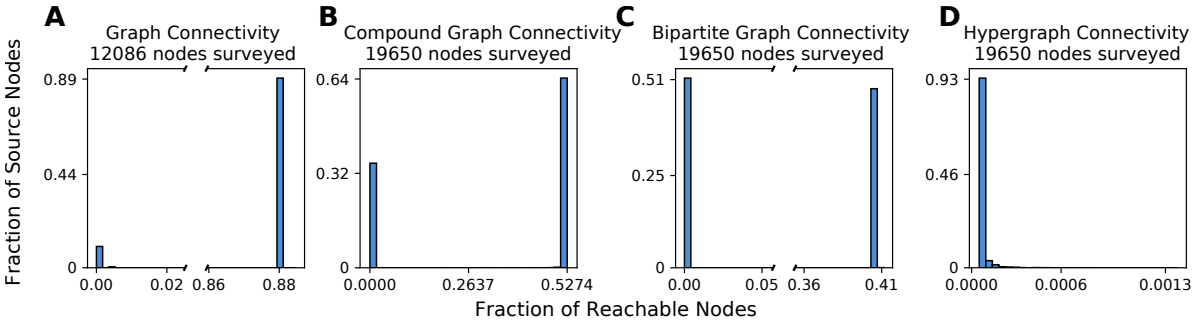

Fig 2. Reactome connectivity across pathway representations. Histograms of the fraction of nodes reached by each source node in the **(A)** directed graph BFS, **(B)** compound graph traversal [25], **(C)** bipartite graph BFS on molecule nodes, and **(D)** hypergraph B-connectivity [28].

We converted the Reactome pathway database to each of the four representations in an effort to determine if they agreed on connectivity (Table 1). The directed graph has far more edges than the other representations since it represents protein complexes as complete graphs (cliques). The hypergraph has more nodes than the graph since it represents protein complexes, families and modified forms as distinct entities. However, since each hyperedge is a multi-way relationship, the number of hyperedges is smaller than the number of edges in directed graphs. The compound graph and bipartite graph have the same node set as the hypergraph, but contain more edges since they describe relationships among entities using pairwise edges. Note that the number of nodes in the bipartite graph is the sum of the number of nodes and the number of hyperedges in the hypergraph, by definition.

In the directed graph representation, nearly 90% of the nodes reached over 80% of the network due to the large number of edges (Fig. 2A). For the other representations, we surveyed the same 19,650 entities representing proteins, small molecules, complexes, and families. We observed two sharp peaks for both the compound and bipartite graph representations: nodes that reach a large portion of the network and nodes that reach very few nodes in the network.

Two-thirds of the nodes in the compound graph representation reach 50% of the network while half the nodes in the bipartite graph representation reach about 40% of the network (Fig. 2B–C). In the hypergraph representation, only five of the nodes are connected to more than 20 others in terms of B-connectivity, and most of the nodes cannot reach any others (Fig. 2D). In hypergraphs, the B-connectivity requirement of visiting all nodes in the tail of a hyperedge before traversal is overly strict for Reactome's topology.

3.2 B-relaxation distance on hypergraphs

Connectivity in four different representations of Reactome largely exhibits an all-or-nothing behavior: nodes are either connected to very few or a large fraction of all other nodes. We introduce B-relaxation distance, a parameterized relaxation of hypergraph B-connectivity that naturally bridges the gap between B-connectivity in directed hypergraphs and connectivity in bipartite graphs. When we consider the connectivity from a node v in the hypergraph, nodes with a B-relaxation distance of 0 from v, denoted B_0, are exactly the nodes that are B-connected to v. Nodes with a B-relaxation distance of 1 (B_1) allows one hyperedge to be freely traversed, lifting the restriction that all nodes in the tail must be visited in order to traverse the hyperedge. In general, nodes with a B-relaxation distance of k (B_k) require k hyperedges to be freely traversed. For shorthand, we will denote $B_{\leq k}$ to be the set of nodes with a B-relaxation distance from a source node of at most k. A formal definition and efficient algorithms for computing B-relaxation distance appear in the Methods).

We computed the B-relaxation distance from every node in the hypergraph to every other node and plotted $|B_{\leq k}|$ for different values of k (Fig. 3A). The first column ($k = 0$) is the number of B-connected nodes for each source, a histogram of which is shown in Fig. 2D. The last column ($k = 49$) corresponds to the other extreme: for each source node, we display the number of nodes that are B-connected to the source while requiring that only one node in the tail of a hyperedge needs to be connected to the source for us to determine that every node in the head of the hyperedge is reachable from the source. The nodes reached for such a large value of k for each source are exactly the nodes that are connected to the source in the bipartite graph representation (Fig. 2C). As in Fig. 2C, we observe the nodes are divided into two sets: the top blue half are nodes that are not connected to very few others and the bottom yellow half are the nodes that are connected to about 40% of the bipartite graph.

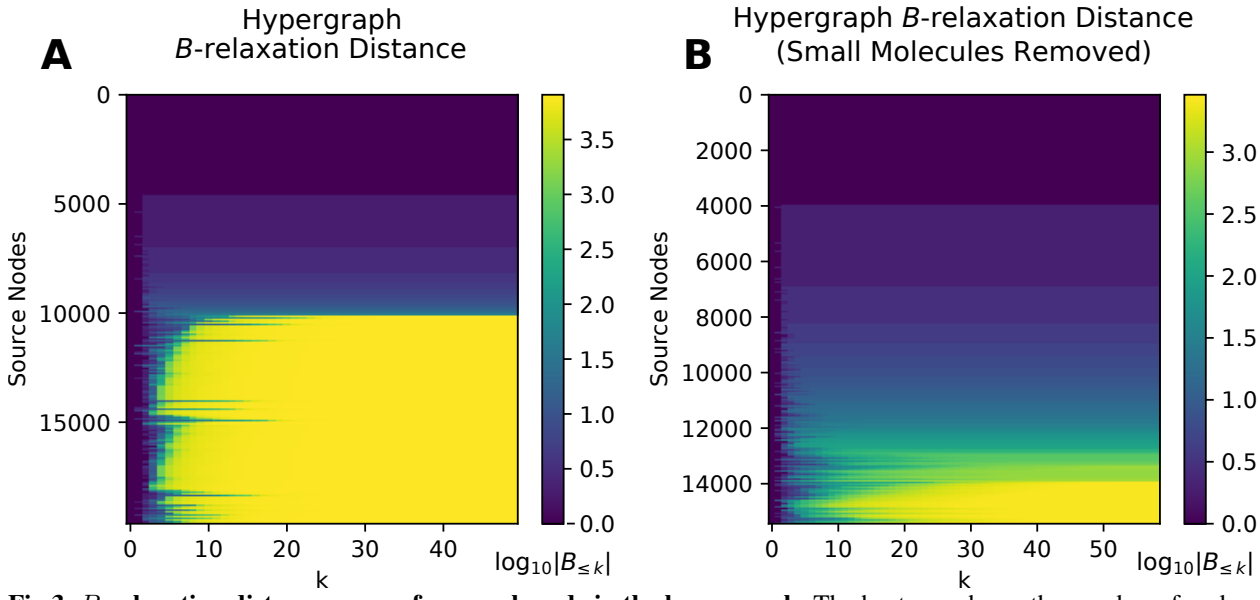

Fig 3. B-relaxation distance survey from each node in the hypergraph. The heatmap shows the number of nodes in the set $B_{\leq k}$ from each source node (rows) for different values of k (columns) in **(A)** the Reactome hypergraph and **(B)** the hypergraph after removing small molecules and three other entities with large connectivity (see text).

The nodes in the bottom half of Fig. 3A exhibited a transition from reaching very few nodes (blue) to reaching many nodes (yellow). The rapidity of this transition suggested that a small number of nodes may be responsible for it. We hypothesized that these nodes may be small molecules, e.g., ATP, water, sodium and potassium ions, that participate in a vast number of reactions that are functionally unrelated. Consequently, we pruned the hypergraph by removing

the 2,778 nodes labeled as small molecules by Reactome, as well as three other highly-connected entities (cytosolic Ubiquitin, nuclear Ubiquitin, and the Nuclear Pore Complex). We also removed hyperedges with an empty tail or head. In total, we altered 5,180 hyperedges by removing these entities, resulting in a filtered hypergraph with 15,440 nodes and 8,773 hyperedges. In this hypergraph, fewer nodes are connected to many others, and further the transition from low-to-high connectivity is more gradual across different source nodes (Fig. 3(B)). In contrast, removing small molecules from the directed graph changed the distribution very little, suggesting that small molecules played only a minor role in the the high level of connectivity in directed graphs (Supplementary Fig. S1). Others have noted that small molecules increase pathway connectivity through reactions that are not intended to be sequential, so we repeated the B-relaxation distance survey after removing 155 ubiquitous small molecules flagged by PathwayCommons [19] from the full hypergraph. As expected, the B-relaxation distance survey on this hypergraph reveals a pattern between the full hypergraph and the hypergraph with small molecules removed (Supplementary Fig. S2).

From these results, we concluded that we had a promising definition of parameterized distance that allowed us to relax the strict assumptions posed by B-connectivity, and a hypergraph where reachability was not affected by ubiquitous molecules that participate in many reactions. For the remainder of this study, we use the hypergraph with all small molecules removed (Fig. 3B).

3.3 Pathway influence across Reactome

While the entire Reactome pathway database appears to be poorly connected in the hypergraph representation, this determination comes from treating individual nodes as sources. We wished to leverage Reactome's pathway annotations to understand how *pathways* are connected in the hypergraph according to B-relaxation distance. We identified 34 signaling pathways in Reactome (Supplementary Table S1) and considered the relationship between pairs of pathways within the hypergraph. When we computed the overlap of the members within each pair of pathways, we found that some pathway pairs already shared nearly all their members (Fig. 4A). For example, the normalized overlap between DAG/IP3 signaling and GPCR signaling is 0.9; DAG and IP3 are second messengers in the phosphoinositol pathway, which is activated by GPCRs. The next largest scores are 0.62 and 0.73 between Insulin Receptor signaling and Insulin-like Growth Factor 1 Receptor (IGF1R) signaling. Other growth factor pathways have moderate overlap (e.g., the overlaps among EGFR, ERBB2, and ERBB range from 0.24 to 0.32).

Our aim is to quantify how well a source pathway S can reach a target pathway T by finding pathway pairs where T is "downstream" of S. Since we wish to find a directed relationship between pathways, we should ignore the initial overlap between their member sets P_S and P_T. Thus, we developed a score that measures how many additional members of T may be reached when computing the B-relaxation distance from S, after accounting for the initial overlap and the total of number of elements that are reached from S. We defined the *influence score* $s_k(S,T)$ of the source pathway S on target pathway T for B-relaxation distance up to k as follows:

$$s_k(S,T) = \frac{|\,(B_{\leq k}(P_S) \cap P_T) \setminus (P_S \cap P_T)\,|}{|B_{\leq k}(P_S) \setminus (P_S \cap P_T)\,|}. \tag{1}$$

This score makes use of the *pathway overlap* between S and T ($P_S \cap P_T$). The numerator counts the number of nodes in T that are reached in the set $B_{\leq k}(P_S)$ that are not already in P_S. The denominator counts the total number of nodes that are reached in $B_{\leq k}(P_S)$ that are not in the pathway overlap. Pathway pairs with a large initial overlap are penalized in this score, allowing more subtle patterns to emerge. Moreover, this score penalizes a pathway P_S that reaches many nodes indiscriminately.

We computed s_k for every pair of Reactome signaling pathways for $k = 0, 1, 2, \ldots$ (Fig. 4B). As k increases, a few pathway pairs exhibit a peak influence score around $k = 3$, including the largest computed influence score across all values of k (red box). There are three pairs that exhibit a large influence score for $k = 3$ (Fig. 4B): (a) the Mst1 pathway's influence on MET signaling ($s_3 = 0.79$), (b) the Activin pathway's influence on TGFβ signaling ($s_3 = 0.54$), and (c) the BMP pathway's influence on TGFβ signaling ($s_3 = 0.48$). We discuss these pathway pairs in two case studies: Mst1 and MET signaling followed by Activin/BMP, and TGFβ signaling.

3.3.1 Mst1 pathway influence on MET signaling

Using Macrophage-stimulating Protein 1 (Mst1) as the source pathway S, we computed the overlap of the other 33 pathways with $B_{\leq k}$ as k increases (Fig. 5). The largest influence score that we observed across all pathway pairs was 0.79 at $k = 3$ for Mst1 to MET signaling, which indicates that almost all the nodes downstream of Mst1 for $k = 3$

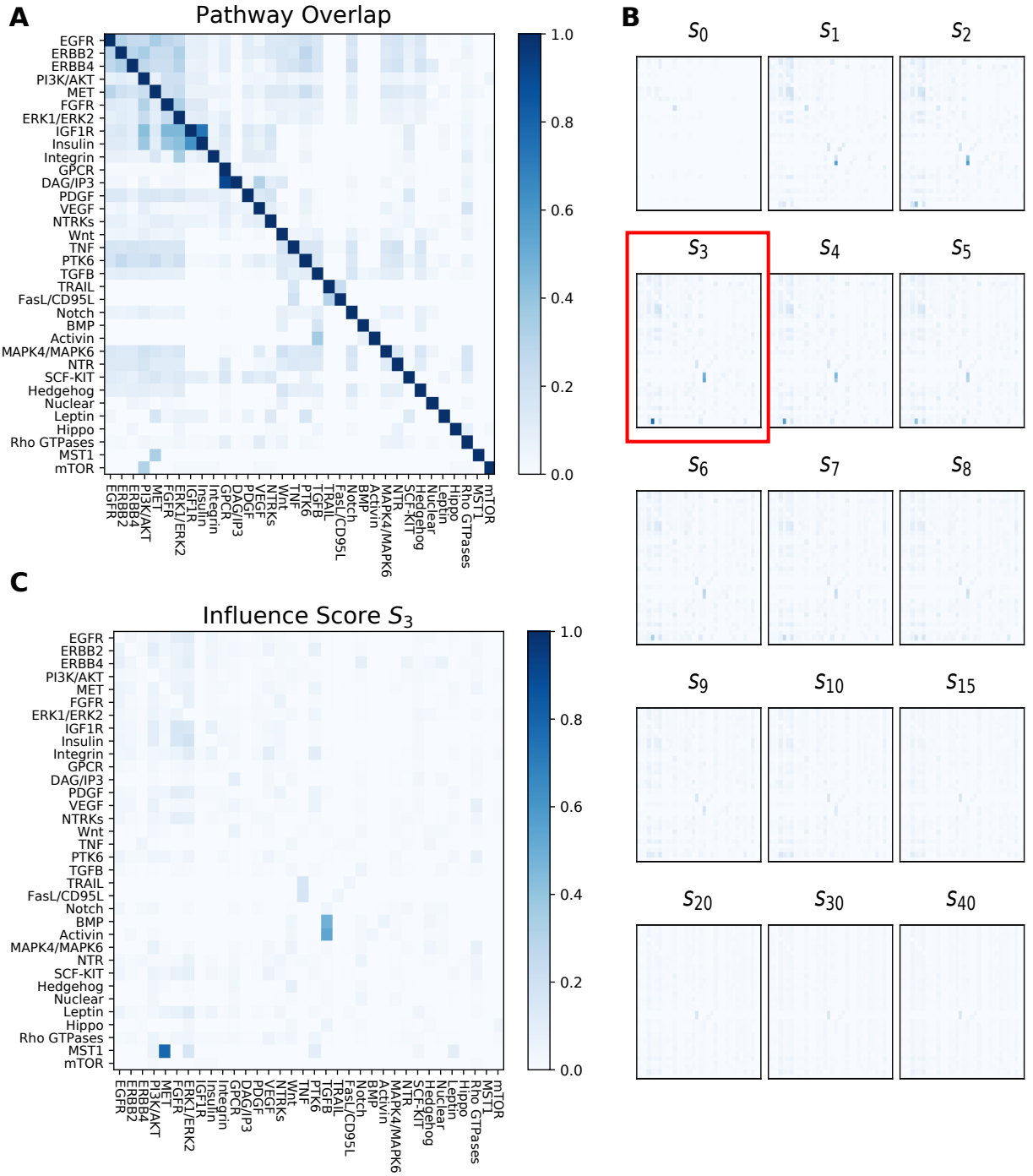

Fig 4. Overlap and influence of 34 Reactome signaling pathways. Rows indicate the source pathway P_S and columns indicate the target pathway P_T. **(A)** Node overlap of pathway members (normalized by the size of P_S). **(B)** Influence scores for different values of k, with S_3 enlarged in Panel **(C)**.

are MET pathway members. For $k = 10$, the set $B_{\leq k}$ contains many ERK1/ERK2 or PI3K/AKT pathway members; however, they comprise a relatively small portion of the total number of nodes in $B_{\leq k}$.

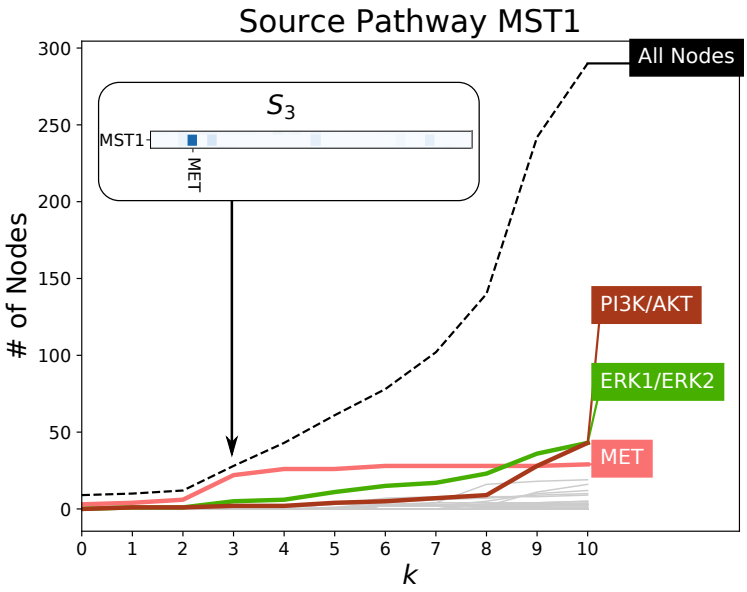

Fig 5. The Mst1 pathway's influence on downstream pathways. The dashed black line indicates the number of nodes in $B_{\leq k}$ using the Mst1 pathway as the source set for different values of k. There is one line for each of the 33 other target pathways denoting the number of members that appear in $B_{\leq k}$; the MET, ERK1/ERK2, and PI3K/AKT pathways are highlighted. The inset, which is a row from Fig. 4C, illustrates the large influence score of Mst1 on MET signaling.

Fig. 5 suggested that the Mst1 pathway may influence the MET pathway. An inspection of the literature and the topology of the nodes in $B_{\leq k}$ from the Mst1 pathway as the source lent support to this hypothesis. Mst1 is produced in the liver and is involved in organ size regulation [33, 34]. Mst1 acts like a hepatocyte growth factor and has been established as a tumor suppressor gene for heptacellular carcinoma [34]. MET, also known as hepatocyte growth factor (HGF) receptor, is a receptor tyrosine kinase that promotes tissue growth in developmental, wound-healing, and cancer metastasis [35]. Mst1, on the other hand, binds to Mst1R (also known as RON), which is a member of the MET family. Both MET and Mst1R have been shown to have similar downstream effects and can trans-phosphorylate when active [36]. Upon inspection of the reactions that involved the nodes $B_{\leq 3}$, we found that Hepsin (HPN) was involved in forming both the Mst1 dimer and HGF dimer (Fig. 6). This protease is known to cleave both pro-Mst1 and pro-HGF into active Mst1 and HGF [37]. The hypergraph also emphasizes the fact that the nodes that in $B_{\leq k}$ but are not in the MET pathway involve STAT regulation in different cellular compartments. The computed pathway influence (observed as an enrichment of stars in Fig. 6 in the regions named B_0, B_1, B_2, and B_3) is due to HPN's role in activating the ligands responsible for both Mst1 signaling and MET signaling. Fig. 6 also displays the nodes in B_4. The high prevalence of nodes that are not in the Met pathway (circles) in this region reinforces the fact that the influence of the Mst1 pathway on the Met pathway is the largest for $k = 3$.

3.3.2 Activin and BMP influence on TGFβ signaling

Following the influence score for Mst1 and MET pathways, the next three largest scores across all pathway pairs and all values of k were for the Activin pathway on TGFβ signaling ($s_2 = 0.58, s_3 = 0.54$) and the Bone Morphogenic Protein (BMP) pathway on TGFβ signaling ($s_3 = 0.48$). The pattern of s_k values for Activin and TGFβ were strikingly similar to the trends for BMP and TGFβ pathways; for both Activin and BMP, TGFβ was the only target pathway that received a large influence (Fig. 7). Even though Activin, BMP, and TGFβ are all known ligands of the TGFβ superfamily, our analysis demonstrates that the Activin and BMP pathways are upstream of the TGFβ pathway. The TGFβ superfamily regulates processes involved in proliferation, growth, and differentiation through both SMAD-dependent and SMAD-

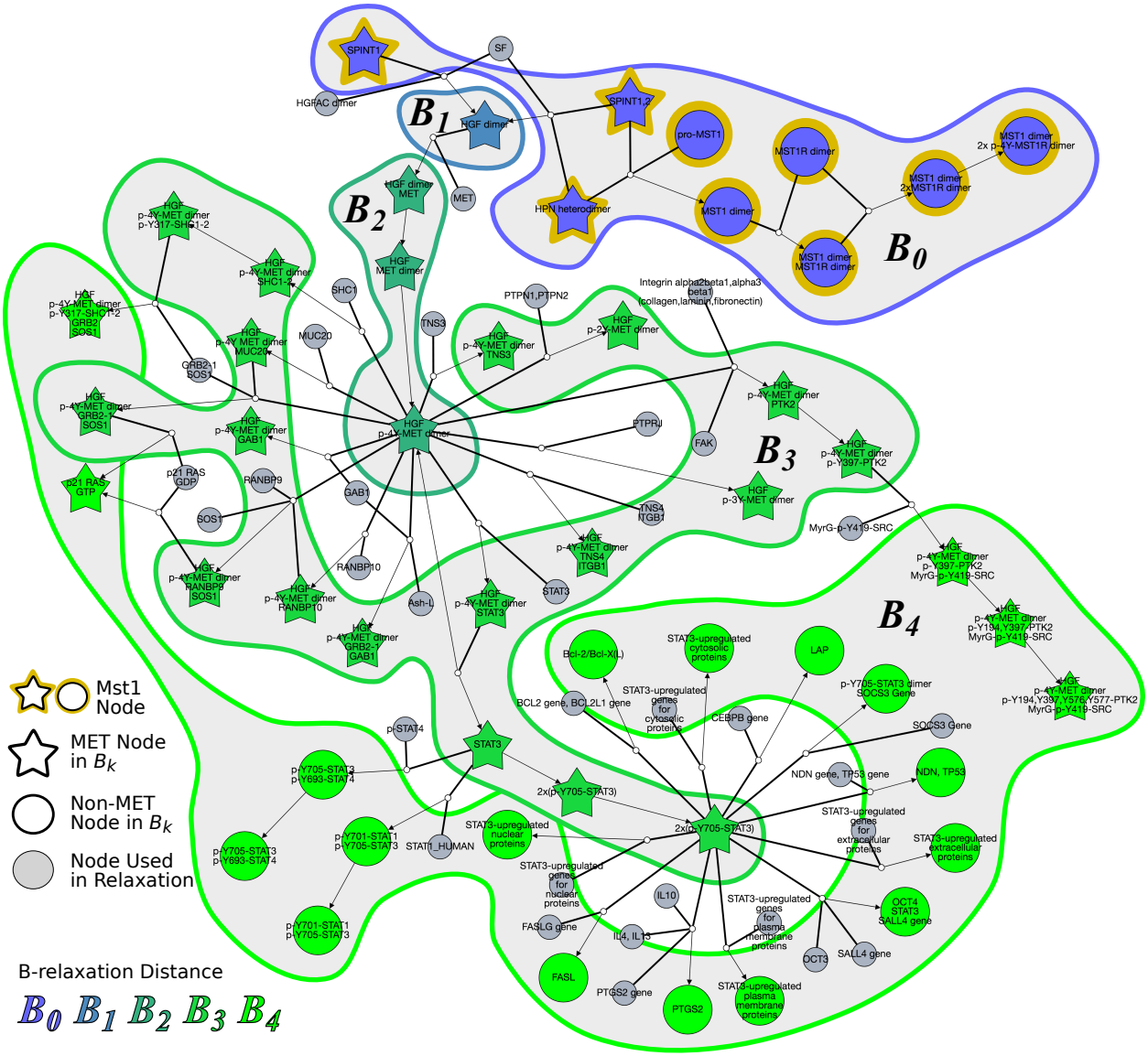

Fig 6. Hyperedges traversed to compute $B_0, B_1, \ldots, B_4$ from source pathway Mst1. Node colors represent B-relaxation distance from $k = 0$ (B_0, blue) to B_4 (bright green). Gray nodes are entities that are not in B_k but are involved in traversed hyperedges. Star-shaped nodes are members of the MET pathway. This network is available on GraphSpace at `http://graphspace.org/graphs/26755?user_layout=6707`.

independent signaling [38]. TGFβ, Activin, and BMP phosphorylate different SMAD proteins by forming dimers and binding to receptor serine/threonine kinases. TGFβ binds to TGFβ Receptor II (TGFBR2), which forms a homeodimer with TGFBR1 and activates SMAD2 and SMAD3. Activin also phosphorylates SMAD2 and SMAD2 through binding and activation of the Activin A receptor (ACVR). BMP, on the other hand, phosphorylates SMAD1, SMAD5, and SMAD8 through BMP receptor activation. The hypergraph that shows the nodes in $B_{\leq 3}$ from Activin consists of different components and many cycles that denote reuse of SMADs (Supplementary Fig. S3). The hypergraph suggests that the influence of Activin on TGFβ does not begin at the ligand, but rather at the activation of SMAD proteins.

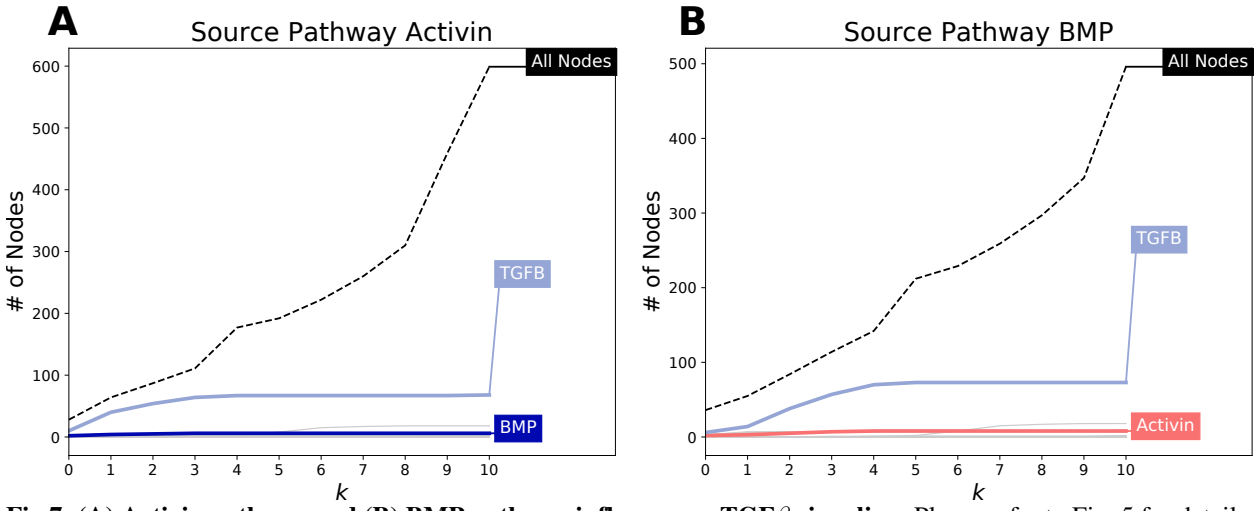

Fig 7. (A) Activin pathway and (B) BMP pathway influence on TGFβ signaling. Please refer to Fig. 5 for details.

4 Discussion

Connectivity is a foundational concept in cellular reaction networks, since it lies at the heart of determining the effect of one molecule upon another. The formal definition of connectivity is familiar and straightforward in directed graphs, the most common mathematical representation of reaction networks. However, precisely capturing this concept is challenging in more sophisticated and biologically accurate representations such as compound graphs, bipartite graphs, and directed hypergraphs. In recent years, scientists have developed these definitions independently for each of these representations.

This work is the first to systematically compare the relevant formulations of connectivity in four different models of reactions in signaling pathways. We study their impact on the Reactome database. Our striking finding is that the directed graph representation of Reactome is very highly connected (90% of the nodes reach over 80% of the graph), the compound and directed graph versions are somewhat less connected (two thirds of the nodes in the compound graph are connected to about half the nodes and half the nodes in the bipartite graph reach about 40% of the nodes), whereas the directed hypergraph model exhibits very poor connectivity (only five nodes are connected to more than 20 nodes).

We attribute this trend to multiple, related factors. The SIF format for Reactome, from which we construct the directed graph, does not distinguish between modified forms of a protein and represents complexes as cliques. Compound graphs, bipartite graphs, and directed hypergraphs create a node for each form of a protein and for each protein complex. However, compound and bipartite graphs are much more connected than hypergraphs since they record multi-way reactions using multiple, independent edges. Directed hypergraphs accurately represent reactions, but their biologically-meaningful definition of connectivity (B-connectivity) is very restrictive in practice.

Motivated by these findings, we have provided a relaxed version of hypergraph connectivity, B-relaxation distance, that is tailored for the analysis of signaling pathways. B-relaxation distance takes the intuitive mechanical significance of B-connectivity and grants it the leeway necessary to deal with the challenges presented by the topologies of biomolecular hypergraphs. We show that B-relaxation distance elegantly bridges the gap between bipartite graphs and hypergraphs.

We use B-relaxation distance to identify downstream influence between annotated pathways in Reactome, defining an influence score s_k that suggests how much a target pathway T might be influenced by the downstream effects of a

source pathway S. After performing an all-vs-all comparison across 34 Reactome pathways, we demonstrate the ability of B-relaxation distance to capture points of influence in two case studies: (a) the effect of the Mst1 pathway on MET signaling and (b) the role of Activin and BMP pathways on TGFβ signaling. Visualizing the hypergraph that contains nodes with small B-relaxation distance can pinpoint the exact reaction or reactions responsible for the influence of one pathway on another. While our findings are not biologically novel, they demonstrate how researchers may explore Reactome in a systematic, unbiased manner to identify possible points of influence among pathways. As pathway databases such as Reactome continue to expand, B-relaxation distance will become a useful measure for systematically characterizing connectivity and relationships among annotated pathways.

Our algorithm for B-relaxation distance runs in polynomial time, and is efficient in practice. However, using directed hypergraphs to solve other computational problems can come with additional algorithmic challenges. For example, the shortest path problem on graphs is widely known to be solvable in polynomial time, while the analogous problem on directed hypergraphs is NP-complete [11, 28], even when bounding the number of nodes in the tail and head sets [29]. These challenges invite the generalization of other classic graph algorithms that have been used in biological applications to directed hypergraphs; in fact, random walks [39] and spectral clustering [40] have already been developed for directed hypergraphs with applications to other fields.

5 Methods

5.1 Connectivity measures

Given a pathway and two entities, we wish to ask a very fundamental connectivity question: "is a downstream of b"? The answer to this question in directed graphs can be efficiently computed using a traversal algorithm such as breadth first search. Established connectivity measures on compound graphs [25] and hypergraphs [28] generalize breadth-first traversal. We begin with hypergraph connectivity and then describe our proposed relaxation to this measure, which is the main computational contribution in this work. We then describe another version of connectivity for compound graphs, which lies conceptually between graph connectivity and hypergraph connectivity.

5.1.1 Hypergraph connectivity

A directed hypergraph $\mathcal{H} = (V, \mathcal{E})$ contains a set V of nodes and a set $\mathcal{E}$ of *hyperedges*, where a hyperedge $e = (T_e, H_e) \in \mathcal{E}$ consists of a tail set $T_e \subseteq V$ and a head set $H_e \subseteq V$ of nodes [28]. The *cardinality* of hyperedge e is the sum of the nodes in the tail and head, i.e., $|T_e| + |H_e|$. Note that directed graphs are a special case of directed hypergraphs where $|T_e| = |H_e| = 1$ for each hyperedge e. In a directed graph, the set of nodes connected to some source s is simply all nodes that are reachable via a path from s. The equivalent notion in a directed hypergraph is B-connectivity. Given a set of nodes $S \subseteq V$, B-connectivity ensures the property that traversing a hyperedge $\in \mathcal{E}$ requires that all the nodes in T_e are connected to S. The following definition is adapted from Gallo et al. [28]:

Definition 1. Given a directed hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a source set $S \subseteq V$, a node $u \in V$ is **B-connected** to S if either (a) $u \in S$ or (b) there exists a hyperedge $e = (T_e, H_e)$ where $u \in H_e$ and each element in T_e is B-connected to S. We use $B(\mathcal{H}, S)$ to denote the set of nodes that are B-connected to S in $\mathcal{H}$.

We can compute $B(\mathcal{H}, S)$ using a hypergraph traversal [28]. This traversal works by finding hyperedges that have tails whose nodes are all B-connected to S, augmenting the set of B-connected nodes with the nodes in the heads of these hyperedges, and repeating this process until it does not discover any new nodes. The running time of this algorithm is linear in the size of $\mathcal{H}$.

5.1.2 Parameterized hypergraph connectivity

While B-connectivity is a biologically useful notion of connectivity, it is overly restrictive for the purpose of assessing the connectivity of pathway databases. We establish a relaxation of B-connectivity which works around such restrictions. Before we formally define B-relaxation distance, we distinguish different sets of hyperedges based on their association with the source set S (Fig. 8).

1. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a source set $S \subseteq V$, a hyperedge $e = (T_e, H_e)$ is **reachable** from S if at least one element of T_e is B-connected to S.

2. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a source set $S \subseteq V$, a hyperedge $e = (T_e, H_e)$ is **traversable** from S if all elements of T_e are B-connected to S.

3. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a source set $S \subseteq V$, a hyperedge e is **restrictive** (with respect to S) if it is reachable but not traversable from S. We use $R(\mathcal{H}, S)$ to denote the set of restrictive hyperedges.

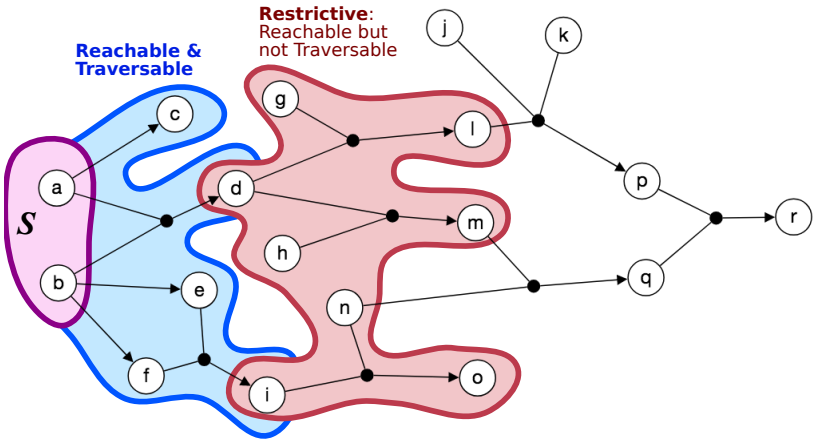

Fig 8. Reachable, traversable and restrictive hyperedges. This hypergraph has eight reachable hyperedges with respect to S: five traversable hyperedges (blue) and three restrictive hyperedges (red).

We modify the `b_visit()` algorithm from [28] to return the B-connected set $B(\mathcal{H}, S)$ and the restrictive hyperedges $R(\mathcal{H}, S)$ (Algorithm 1). The main difference between this traversal and a typical BFS is that a hyperedge is traversed only when all the nodes in the head have been visited. We also return the set of traversed hyperedges to avoid redundant computation in the relaxation algorithm that we describe later.

Algorithm 1 `b_visit`$(\mathcal{H} = (V, \mathcal{E}), S \subset V)$

$\mathsf{c}[e] \leftarrow 0$ for each hyperedge $e \in \mathcal{E}$ *// counter of reached nodes in e's tail*
$B \leftarrow S$ *// set of B-connected nodes*
$X \leftarrow \emptyset$ *// set of traversed hyperedges*
$Q \leftarrow S$ *// queue of nodes to traverse*
while Q is nonempty **do**
  select and remove some node $v \in Q$
  for each hyperedge $e \in \mathcal{E}$ where $v \in T_e$ **do**
    $\mathsf{c}[e] \leftarrow \mathsf{c}[e] + 1$
    if $\mathsf{c}[e] = |T_e|$ **then**
      $Q \leftarrow Q \cup [H_e \setminus B]$ *// add unvisited heads of e to queue*
      $B \leftarrow B \cup H_e$ *// add heads of e to B-connected set*
      $X \leftarrow X \cup \{e\}$ *// add e to traversed hyperedges*
$R \leftarrow \emptyset$ *// set of restrictive hyperedges*
for each hyperedge $e \in \mathcal{E}$ **do**
  if $\mathsf{c}[e] \geq 1$ and $\mathsf{c}[e] < |T_e|$ **then**
    $R \leftarrow R \cup \{e\}$ *// hyperedge e reached but not traversed*
return B, R, X

We iteratively relax the notion of B-connectivity by allowing restrictive hyperedges to be traversed; to do so, at each iteration k we need to keep track of $B_k(\mathcal{H}, S)$, the connected nodes, and $R_k(\mathcal{H}, S)$, the restrictive hyperedges. We initialize these sets to be the outputs of `b_visit()`:

$$B_0(\mathcal{H}, S) = B(\mathcal{H}, S) \tag{2}$$
$$R_0(\mathcal{H}, S) = R(\mathcal{H}, S). \tag{3}$$

In the kth iteration of this relaxation process, we consider the heads of each restrictive hyperedge e from the previous iteration. $B_k(\mathcal{H}, S)$ is the set of B-connected nodes and $R_k(\mathcal{H}, S)$ is the set of restrictive hyperedges for each head set from $R_{k-1}(\mathcal{H}, S)$:

$$B_k(\mathcal{H}, S) = \bigcup_{e \in R_{k-1}(\mathcal{H},S)} B(\mathcal{H}, H_e) \tag{4}$$

$$R_k(\mathcal{H}, S) = \bigcup_{e \in R_{k-1}(\mathcal{H},S)} R(\mathcal{H}, H_e). \tag{5}$$

Note that computing $R_k(\mathcal{H}, S)$ using this definition requires $|R_{k-1}(\mathcal{H}, S)|$ different `b_visit()` calls, which is necessary to ensure that only one restrictive hyperedge is used to establish connectivity. With these definitions in hand, we are now ready to define our relaxation of B-connectivity.

Definition 2. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, a source set $S \subseteq V$, and an integer $k \geq 0$, a node $v \in V$ is **B_k-connected** to S if $v \in B_i(\mathcal{H}, S)$ for $i = 0, 1, \ldots, k$.

The *B-relaxation distance* of a node v from a source set S is the smallest value of k such that v is B_k-connected to S in $\mathcal{H}$. In the main text, we use $B_{\leq k}$ to denote the B_k-connected set. An example of computing B-relaxation distance for all nodes in a hypergraph is shown in Fig. 9.

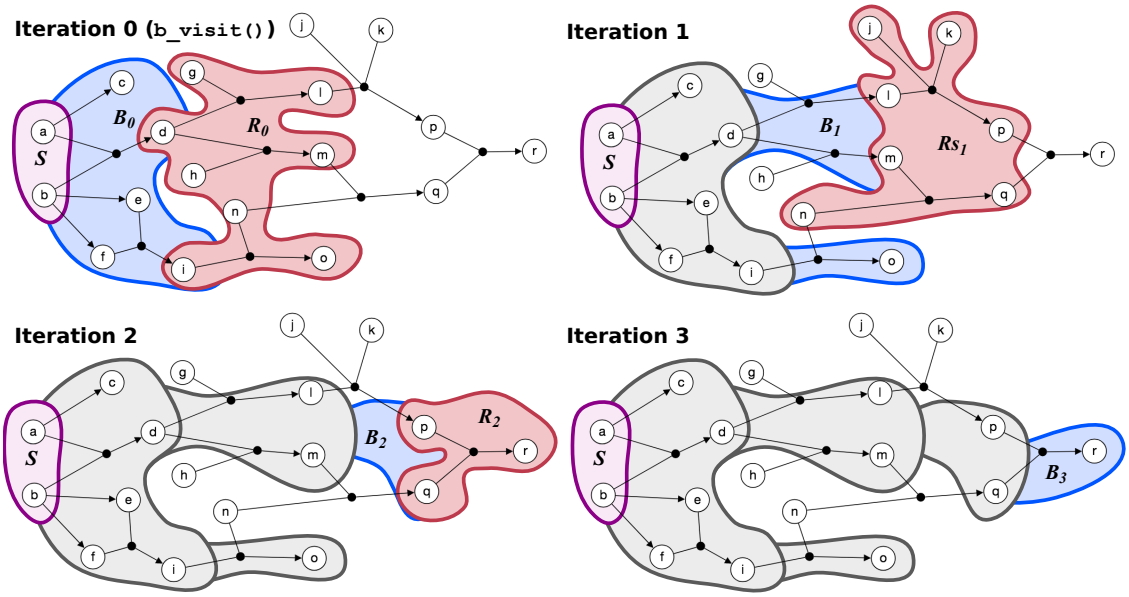

Fig 9. Computing B-relaxation distance. Connected nodes are in blue and restrictive hyperedges are in red for each iteration k. In this example, all nodes are B_3-connected to $S = \{a, b\}$ and node r has B-relaxation distance of three.

We calculate the B-relaxation distance from S to every node in the hypergraph by calling `b_visit()` on restrictive hyperedges for $k = 0, 1, 2, \ldots$ (Algorithm 2). The algorithm first calls `b_visit()` from S to get the B-connected set B_0,[1] the restrictive hyperedges Q_0, and the traversed hyperedges X (line 1). The B-relaxation distance dictionary `dist` is initialized to 0 for nodes in B_0 and infinity otherwise, and the `seen` dictionary of hypereges set to `True` if they have been traversed and `False` otherwise. While there are unseen restrictive hyperedges to traverse, the algorithm computes B_k and R_k by calling `b_visit()` on the heads of each restrictive hyperedge from iteration $k-1$ (lines 7–11). We update the `seen` dictionary with all traversed hyperedges from each `b_visit()`, since these hyperedges may be restrictive with respect to another set of nodes and would be recomputed at a later iteration (lines 12–13, Supplementary Fig. S5). Finally, the algorithm updates the `dist` dictionary for all nodes that are reached in the k-th iteration and increments k (lines 14–17). This implementation keeps track of $B_0, B_1, \ldots, B_k$ and $R_0, R_1, \ldots, R_k$, which may be returned for other purposes.

[1] In the algorithm we drop the parameterization of the hypergraph and source set to declutter notation.

Algorithm 2 `b_relaxation` $(\mathcal{H} = (V, \mathcal{E}), S \subseteq V)$

1: B_0, R_0, $X \leftarrow$ `b_visit`($\mathcal{H}$,S)
2: `dist`$[v] \leftarrow 0$ if $v \in B_0$ else ∞ for each node $v \in V$
3: `seen`$[e] \leftarrow$ `True` if $e \in X$ else `False` for each hyperedge $e \in \mathcal{E}$
4: $k \leftarrow 1$
5: **while** there exists some $e \in R_{k-1}$ where `seen`$[e] =$ `false` **do**
6: $\quad B_k \leftarrow \emptyset$, $R_k \leftarrow \emptyset$
7: $\quad$ **for** $e \in R_{k-1}$ where `seen`$[e] =$ `False` **do**
8: $\quad\quad$ `seen`$[e] \leftarrow$ `True`
9: $\quad\quad B$, R, $X \leftarrow$ `b_visit`($\mathcal{H}$,H_e)
10: $\quad\quad B_k \leftarrow B_k \cup B$
11: $\quad\quad R_k \leftarrow R_k \cup R$
12: $\quad\quad$ **for** e' in X **do**
13: $\quad\quad\quad$ `seen`$[e'] \leftarrow$ `True`
14: $\quad$ **for** v in B_k **do**
15: $\quad\quad$ **if** `dist`$[v] = \infty$ **then**
16: $\quad\quad\quad$ `dist`$[v] \leftarrow k$
17: $\quad k \leftarrow k + 1$
18: **return** `dist`

Runtime analysis. The original `b_visit()` from Gallo et al. runs in $\mathcal{O}(size(\mathcal{H}))$ time where $size(\mathcal{H})$ refers to the sum of the hyperedge cardinalities in $\mathcal{H}$ [28]. The modified `b_visit` incurs no additional asymptotic runtime cost since the timing of the additional operations it conducts (Algorithm 1, lines 13-16) is trivially bounded by $|\mathcal{E}|$, which is bounded by $size(\mathcal{H})$.

In Algorithm 2, initializing the `dist` and `seen` dictionaries takes $|V|$ and $|\mathcal{E}|$ time, respectively. The while loop (line 5) contains two for loops. The first loop in line 7 iterates over all restrictive hyperedges, performing work only when that hyperedge has not been previously traversed. Thus, the code in the first loop will be executed at most $|\mathcal{E}|$ times over the full course of the algorithm, corresponding to the case where every hyperedge in $\mathcal{H}$ appears in some restrictive set. The first loop calls `b_visit()` in line 9 at each iteration, which runs in $\mathcal{O}(size(\mathcal{H}))$ time as previously mentioned. The second loop in line 14 updates the B-relaxation distance of each node exactly once, when it is first discovered by the algorithm. It will be executed at most $|V|$ times over the full course of the algorithm. The running time of the first loop (line 7) dominates those of the initialization steps and the distance update loop; thus, the runtime of Algorithm 2 is $\mathcal{O}(|\mathcal{E}| \cdot size(\mathcal{H}))$.

Pre-processing speedup. When we ran `b_relaxation()` on each source node on the Reactome hypergraph, the algorithm took an average of 31.6 seconds per node on a Linux machine with quad Intel Core i7-4790 processors. The quadratic runtime is tractable for a handful of calls, but calling `b_relaxation()` from every vertex in V (as we do in this work) will result in a cubic runtime. We formulated an optimized version of `b_relaxation()`, which we initialized by calling `b_visit()` on H_e for each $e \in \mathcal{E}$ and recording the resulting connected nodes and restrictive hyperedges. This initialization step incurs a cost of $|\mathcal{E}| \cdot size(\mathcal{H})$ time, but replaces the call to `b_visit()` in line 9 with a constant-time lookup operation. Thus the sole quadratic term in the runtime of Algorithm 2 becomes linear in the optimized version. The optimized version, when applied to each source node on the Reactome hypergraph, gave an average running time of 0.310 seconds per node, giving an improvement of two orders of magnitude.

5.1.3 Compound graph connectivity

There are multiple definitions of compound graphs [8, 25]. Here we describe *compound pathway graphs* $CP = (G, I)$ that consist of two graphs [25]. The pathway graph $G = (V, E_G)$ is a mixed graph where V denotes the set of nodes and E_G denotes the interaction and regulation edges among nodes, some of which may be directed.[2] The inclusion graph $I = (V, E_I)$ is on the same node set V and E_I denotes the undirected edges for defining compound structure membership (e.g., complexes and abstractions). To traverse a compound pathway graph, we need, for each compound

[2]Edges may also denote inhibition/activation; here, we ignore this aspect of the compound graph.

structure, two flags: (a) `compound`: if a compound structure is reached, are all its members also reached? and (b) `member`: if a member of a compound structure is reached, are all other members in the compound structure also reached? During the traversal, once a node u is reached, the algorithm determines if any other nodes are "equivalent" to u based on these flags. Note that while compound graphs handle traversals through entities such as protein complexes and families, the edges only connect pairs of these entities. Thus, the requirements imposed by B-connectivity on hypergraphs cannot be implemented on compound graphs as they are currently defined.

A *compound path* between two nodes consists of edges that are either from the pathway graph E_G or represent a link between nodes that are equivalent for traversal based on the `compound` and `structure` flags. These compound paths are used to establish the set of nodes that are downstream of a source node. For comparison with other measures, we modify the definition from [25] to ignore activation/inhibition effects and remove a restriction on path lengths:

Definition 3. Given a compound pathway graph $CP = (G, I)$ and a source set $S \subseteq V$, a node $u \in V$ is **downstream** of S in CP if there exists some compound path from any node $s \in S$ to u in CP.

We run the `DOWNSTREAM` algorithm implemented in the PaxTools software [25, 41] on each source node in S, ignoring activation/inhibition sign and the path length limit.

5.2 Data formats and representations

We automatically generate the four Reactome representations – directed graph, compound graph, bipartite graph, and hypergraph – using a suite of tools (Fig. 10). We use PathwayCommons, a unified collection of publicly-available pathway data [19], to collect BioPAX and SIF files representing the entire Reactome database (`http://www.pathwaycommons.org/archives/PC2/v10/`). The SIF files are generated by PathwayCommons by converting BioPAX relationships to binary relations; more details are available at `http://www.pathwaycommons.org/pc2/formats`. We convert the SIF files to a directed graph by converting each binary relation to a directed or bidirected graph (Supplementary Table S2).

We use the PaxTools java parser to work with BioPAX files [41]. PaxTools offers querying algorithms such as `DOWNSTREAM` that operates on the compound graph representation [25]. We use PaxTools to construct hypergraphs by traversing the BioPAX files. For each biochemical reaction in BioPAX, we construct a hyperedge with the reactants and control elements in the tail and the products in the head. We use the algorithms provided in the Hypergraph Algorithms Package (HALP, `http://murali-group.github.io/halp/`) to work with hypergraphs. The B-relaxation distance algorithm is provided in a developmental branch of HALP. Finally, we build the bipartite graph directly from the hypergraph, converting each hyperedge e into a reaction node r and connecting the tails of e to r and then r to the heads of e. Thus, the number of nodes in the bipartite graph is exactly the number of nodes plus the number of hyperedges in the hypergraph, and large B-relaxation distance corresponds to traversing the bipartite graph.

We visualize hypergraphs using GraphSpace [42], a web-based collaborative network visualization tool. The hypergraphs are available as interactive networks on GraphSpace using the with the `GLBio2019` tag (`http://graphspace.org/graphs/?query=tags:glbio2019`).

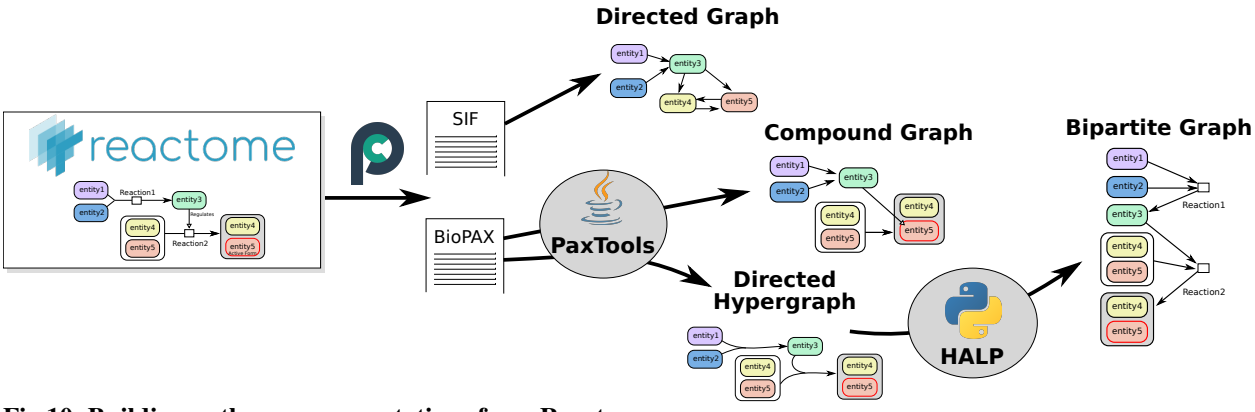

Fig 10. Building pathway representations from Reactome.

Acknowledgments

We thank Brendan Avent for his initial work on the hypergraph algorithms library and Ozgun Babur for discussions about BioPAX and PaxTools. This work is supported by the National Science Foundation under grants DBI-1750981 (to PI AR) and CCF-1617678 (to PI TMM).

References

[1] Purvesh Khatri, Marina Sirota, and Atul J Butte. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS computational biology*, 8(2):e1002375, 2012.

[2] W. Winterbach, P. Van Mieghem, M. Reinders, H. Wang, and D. de Ridder. Topology of molecular interaction networks. *BMC Syst Biol*, 7:90, Sep 2013.

[3] Cristina Mitrea, Zeinab Taghavi, Behzad Bokanizad, Samer Hanoudi, Rebecca Tagett, Michele Donato, Calin Voichita, and Sorin Draghici. Methods and approaches in the topology-based analysis of biological pathways. *Frontiers in physiology*, 4:278, 2013.

[4] Vladimir Gligorijević and Nataša Pržulj. Methods for biological data integration: perspectives and challenges. *Journal of the Royal Society Interface*, 12(112):20150571, 2015.

[5] Marc Vidal, Michael E Cusick, and Albert-László Barabási. Interactome networks and human disease. *Cell*, 144(6):986–998, 2011.

[6] Michael Caldera, Pisanu Buphamalai, Felix Müller, and Jörg Menche. Interactome-based approaches to human disease. *Current Opinion in Systems Biology*, 3:88–94, 2017.

[7] Pau Creixell, Jüri Reimand, Syed Haider, Guanming Wu, Tatsuhiro Shibata, Miguel Vazquez, Ville Mustonen, Abel Gonzalez-Perez, John Pearson, Chris Sander, et al. Pathway and network analysis of cancer genomes. *Nature methods*, 12(7):615, 2015.

[8] Zhenjun Hu, Joe Mellor, Jie Wu, Minoru Kanehisa, Joshua M Stuart, and Charles DeLisi. Towards zoomable multidimensional maps of the cell. *Nature biotechnology*, 25(5):547–554, 2007.

[9] S. Klamt, U. U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Comput. Biol.*, 5(5):e1000385, May 2009.

[10] T. S. Christensen, A. P. Oliveira, and J. Nielsen. Reconstruction and logical modeling of glucose repression signaling pathways in Saccharomyces cerevisiae. *BMC Syst Biol*, 3:7, Jan 2009.

[11] Anna Ritz, Allison N Tegge, Hyunju Kim, Christopher L Poirel, and TM Murali. Signaling hypergraphs. *Trends in biotechnology*, 32(7):356–362, 2014.

[12] W. Zhou and L. Nakhleh. Properties of metabolic graphs: biological organization or representation artifacts? *BMC Bioinformatics*, 12:132, May 2011.

[13] David Croft, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R Kamdar, et al. The reactome pathway knowledgebase. *Nucleic acids research*, 42(D1):D472–D477, 2013.

[14] Antonio Fabregat, Florian Korninger, Guilherme Viteri, Konstantinos Sidiropoulos, Pablo Marin-Garcia, Peipei Ping, Guanming Wu, Lincoln Stein, Peter D?Eustachio, and Henning Hermjakob. Reactome graph database: Efficient access to complex pathway data. *PLoS computational biology*, 14(1):e1005968, 2018.

[15] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, 45(D1):D353–D361, Jan 2017.

[16] Huaiyu Mi and Paul Thomas. Panther pathway: an ontology-based pathway database coupled with data analysis tools. In *Protein Networks and Pathway Analysis*, pages 123–140. Springer, 2009.

[17] K. Kandasamy, S. S. Mohan, R. Raju, S. Keerthikumar, G. S. Kumar, A. K. Venugopal, D. Telikicherla, J. D. Navarro, S. Mathivanan, C. Pecquet, S. K. Gollapudi, S. G. Tattikota, S. Mohan, H. Padhukasahasram, Y. Subbannayya, R. Goel, H. K. Jacob, J. Zhong, R. Sekhar, V. Nanjappa, L. Balakrishnan, R. Subbaiah, Y. L. Ramachandra, B. A. Rahiman, T. S. Prasad, J. X. Lin, J. C. Houtman, S. Desiderio, J. C. Renauld, S. N. Constantinescu, O. Ohara, T. Hirano, M. Kubo, S. Singh, P. Khatri, S. Draghici, G. D. Bader, C. Sander, W. J. Leonard, and A. Pandey. NetPath: a public resource of curated signal transduction pathways. *Genome Biol.*, 11(1):R3, Jan 2010.

[18] R. Elkon, R. Vesterman, N. Amit, I. Ulitsky, I. Zohar, M. Weisz, G. Mass, N. Orlev, G. Sternberg, R. Blekhman, J. Assa, Y. Shiloh, and R. Shamir. SPIKE–a database, visualization and analysis tool of cellular signaling pathways. *BMC Bioinformatics*, 9:110, Feb 2008.

[19] Ethan G Cerami, Benjamin E Gross, Emek Demir, Igor Rodchenkov, Özgün Babur, Nadia Anwar, Nikolaus Schultz, Gary D Bader, and Chris Sander. Pathway commons, a web resource for biological pathway data. *Nucleic acids research*, 39(suppl_1):D685–D690, 2010.

[20] Martina Kutmon, Anders Riutta, Nuno Nunes, Kristina Hanspers, Egon L Willighagen, Anwesha Bohler, Jonathan Mélius, Andra Waagmeester, Sravanthi R Sinha, Ryan Miller, et al. Wikipathways: capturing the full diversity of pathway knowledge. *Nucleic acids research*, 44(D1):D488–D494, 2015.

[21] Emek Demir, Michael P Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'eustachio, Carl Schaefer, Joanne Luciano, et al. The biopax community standard for pathway data sharing. *Nature biotechnology*, 28(9):935–942, 2010.

[22] R. Samaga and S. Klamt. Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Commun. Signal*, 11(1):43, Jun 2013.

[23] Luis Sordo Vieira and Paola Vera-Licona. Computing signal transduction in signaling networks modeled as boolean networks, petri nets and hypergraphs. *bioRxiv*, 2018.

[24] Ken-ichiro Fukuda and Toshihisa Takagi. Knowledge representation of signal transduction pathways. *Bioinformatics*, 17(9):829–837, 2001.

[25] Ugur Dogrusoz, Ahmet Cetintas, Emek Demir, and Ozgun Babur. Algorithms for effective querying of compound graph-based pathway databases. *BMC bioinformatics*, 10(1):376, 2009.

[26] Charles J Vaske, Stephen C Benz, J Zachary Sanborn, Dent Earl, Christopher Szeto, Jingchun Zhu, David Haussler, and Joshua M Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. *Bioinformatics*, 26(12):i237–i245, 2010.

[27] Claude Berge. Graphs and hypergraphs. 1973.

[28] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2):177 – 201, 1993.

[29] A Ritz, B Avent, and TM Murali. Pathway analysis with signaling hypergraphs. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(5):1042, 2017.

[30] Derek Ruths, Melissa Muller, Jen-Te Tseng, Luay Nakhleh, and Prahlad T Ram. The signaling petri net-based simulator: a non-parametric strategy for characterizing the dynamics of cell-specific signaling networks. *PLoS computational biology*, 4(2):e1000005, 2008.

[31] Bree B Aldridge, Julio Saez-Rodriguez, Jeremy L Muhlich, Peter K Sorger, and Douglas A Lauffenburger. Fuzzy logic analysis of kinase pathway crosstalk in tnf/egf/insulin-induced signaling. *PLoS computational biology*, 5(4):e1000340, 2009.

[32] Camille Terfve, Thomas Cokelaer, David Henriques, Aidan MacNamara, Emanuel Goncalves, Melody K Morris, Martijn van Iersel, Douglas A Lauffenburger, and Julio Saez-Rodriguez. Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC systems biology*, 6(1):133, 2012.

[33] M-H Wang, Y-Q Zhou, and Y-Q Chen. Macrophage-stimulating protein and ron receptor tyrosine kinase: Potential regulators of macrophage inflammatory activities. *Scandinavian journal of immunology*, 56(6):545–553, 2002.

[34] Dawang Zhou, Claudius Conrad, Fan Xia, Ji-Sun Park, Bernhard Payer, Yi Yin, Gregory Y Lauwers, Wolfgang Thasler, Jeannie T Lee, Joseph Avruch, et al. Mst1 and mst2 maintain hepatocyte quiescence and suppress hepatocellular carcinoma development through inactivation of the yap1 oncogene. *Cancer cell*, 16(5):425–438, 2009.

[35] Livio Trusolino, Andrea Bertotti, and Paolo M Comoglio. Met signalling: principles and functions in development, organ regeneration and cancer. *Nature reviews Molecular cell biology*, 11(12):834, 2010.

[36] A Follenzi, S Bakovic, P Gual, MC Stella, P Longati, and PM Comoglio. Cross-talk between the proto-oncogenes met and ron. *Oncogene*, 19(27):3041, 2000.

[37] Rajkumar Ganesan, Ganesh A Kolumam, S Jack Lin, Ming-Hong Xie, Lydia Santell, Thomas D Wu, Robert A Lazarus, Amitabha Chaudhuri, and Daniel Kirchhofer. Proteolytic activation of pro-macrophage-stimulating protein by hepsin. *Molecular Cancer Research*, 9(9):1175–1186, 2011.

[38] Erine H Budi, Dana Duan, and Rik Derynck. Transforming growth factor-β receptors and smads: regulatory complexity and functional versatility. *Trends in cell biology*, 27(9):658–672, 2017.

[39] Aurélien Ducournau and Alain Bretto. Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Computer Vision and Image Understanding*, 120:91–102, 2014.

[40] Tom Michoel and Bruno Nachtergaele. Alignment and integration of complex networks by hypergraph-based spectral clustering. *Physical Review E*, 86(5):056111, 2012.

[41] Emek Demir, Özgün Babur, Igor Rodchenkov, Bülent Arman Aksoy, Ken I Fukuda, Benjamin Gross, Onur Selçuk Sümer, Gary D Bader, and Chris Sander. Using biological pathway data with paxtools. *PLoS computational biology*, 9(9):e1003194, 2013.

[42] Aditya Bharadwaj, Divit P Singh, Anna Ritz, Allison N Tegge, Christopher L Poirel, Pavel Kraikivski, Neil Adames, Kurt Luther, Shiv D Kale, Jean Peccoud, et al. Graphspace: stimulating interdisciplinary collaborations in network biology. *Bioinformatics*, 33(19):3134–3136, 2017.