Hyperpath Relaxations for Signaling Pathway Analysis

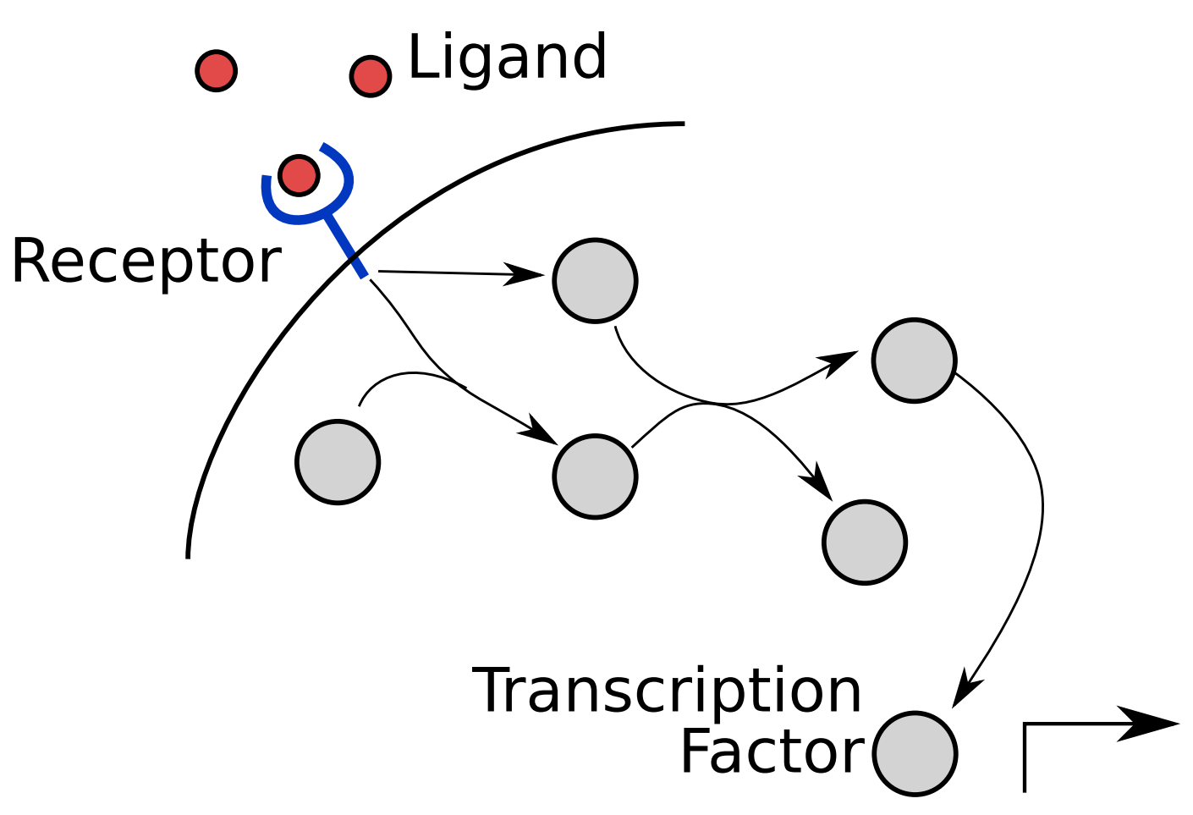
Nicholas Franzese[1,2], Barney Potter[1,2], Adam Groce[1], James Fix[1], and **Anna Ritz**[2]

[1]Mathematics Department, Reed College, Portland, OR, USA

[2]Biology Department, Reed College, Portland, OR, USA

REED COLLEGE

Motivation



Signaling pathways are series of reactions that are typically initiated by an extracellular ligand to a membrane-bound receptor, culminating in altered expression of a set of target genes. Pathways are commonly represented as graphs, which offer elegant algorithms for analyzing signaling pathways but fail to capture many-to-many relationships among molecules in signaling reactions [4]. We recently presented a **shortest path formulation** posed on **directed hypergraphs**, which capture many aspects of signaling hypergraphs [5]. However, it offered a strict and restrictive definition of connectivity that limited applicability to real-world signaling pathways.

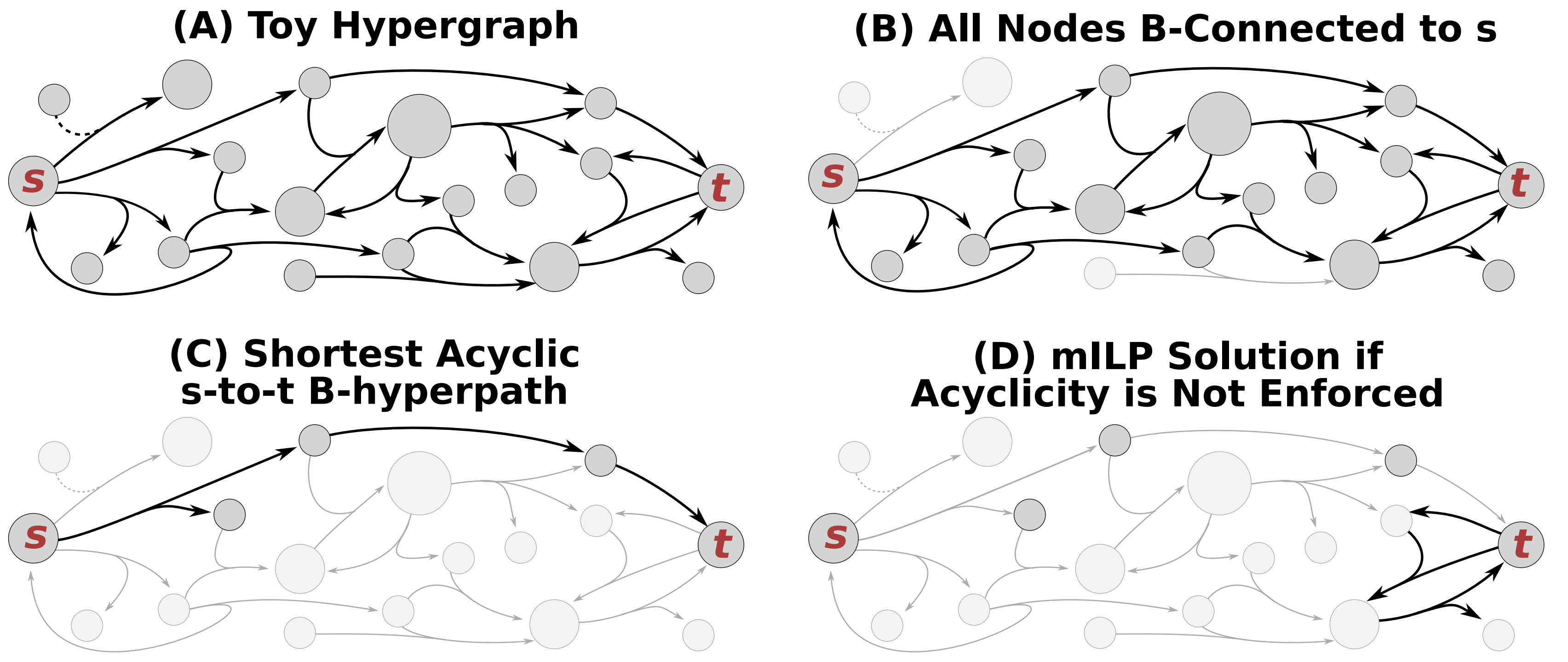
Contributions

We extend a mixed Integer Linear Program (mILP) that achieve hyperpath relaxations in the following ways:

1. We allow **simple cycles** in shortest hyperpaths that capture feedback loops.
2. We allow **plausible "source" nodes** that are not specified in advance.

We apply these relaxations to hypergraphs built automatically from pathway databases [1-3].

Related Work & Hypergraph Notation


(A) Toy Hypergraph

(B) All Nodes B-Connected to s

(C) Shortest Acyclic s-to-t B-hyperpath

(D) mILP Solution if Acyclicity is Not Enforced

A directed hypergraph $\mathcal{H} = (V, E)$ consists of a set V of nodes and a set E of hyperedges, where a hyperedge $e = (T_e, H_e)$ is an ordered pair of node sets $T, H \subseteq V$ (Panel A). Nodes may represent proteins, small molecules, complexes, or other pathways. A node $s \in V$ is B-connected to a node $t \in V$ in a recursive fashion, with the intuition that t is reachable from s only if it can be "produced" by a series of reactions starting with s (Panel B). Previous work designed a mILP to solve the NP-hard shortest s-to-t B-hyperpath that contains no cycles (Panel C) [5]. Introduce binary variables "active" α_v for every node and α_e for every hyperedge.

$$\min \sum_{e \in E} \alpha_e \text{ s.t.} \qquad \textit{Minimize the number of active hyperedges}$$

$$\forall\, v \in V \backslash \{s\} : \alpha_v \leq \sum_{e: v \in H_e} \alpha_e \qquad \textit{Active nodes must have an active incoming hyperedge}$$

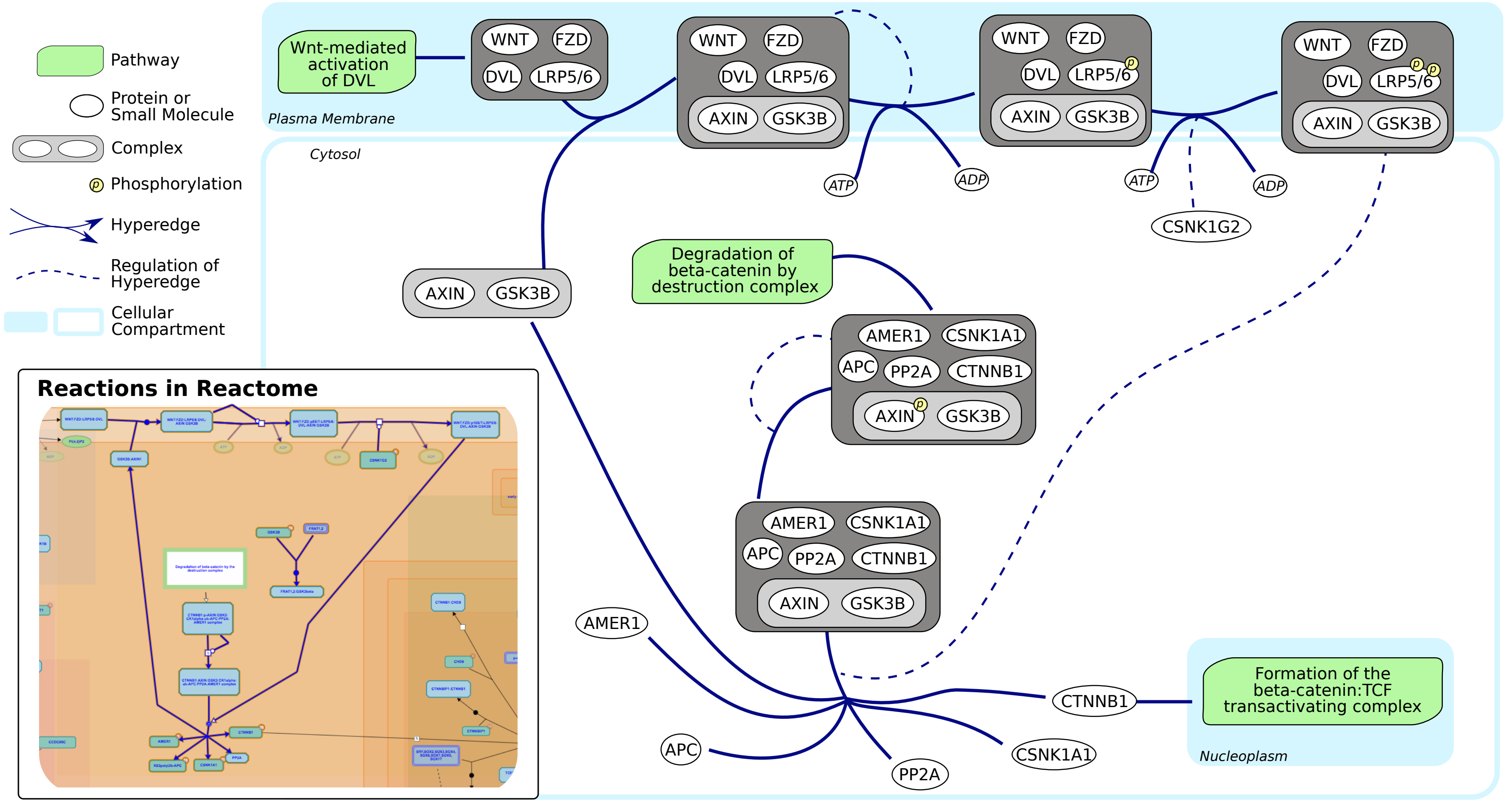
$$\forall\, e \in E : \sum_{v \in T_e} \alpha_v \geq |T_e| \alpha_e \qquad \textit{All nodes in the tail of an active hyperedge must be active}$$

$$\forall\, e \in E : \sum_{v \in H_e} \alpha_v \geq |H_e| \alpha_e \qquad \textit{All nodes in the head of an active hyperedge must be active}$$

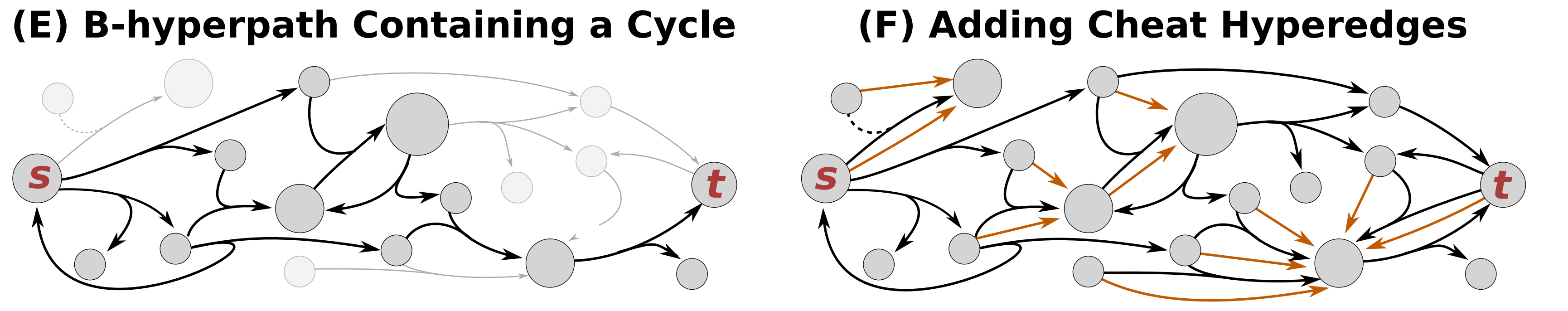
$$\alpha_t = 1 \qquad \textit{The target node must be in the solution.}$$

This formulation, along with a topological ordering constraint to prohibit simple cycles (Panel D), provably returns the shortest s-to-t acyclic B-hyperpath.

Hypergraph Example



Algorithms


(E) B-hyperpath Containing a Cycle

(F) Adding Cheat Hyperedges

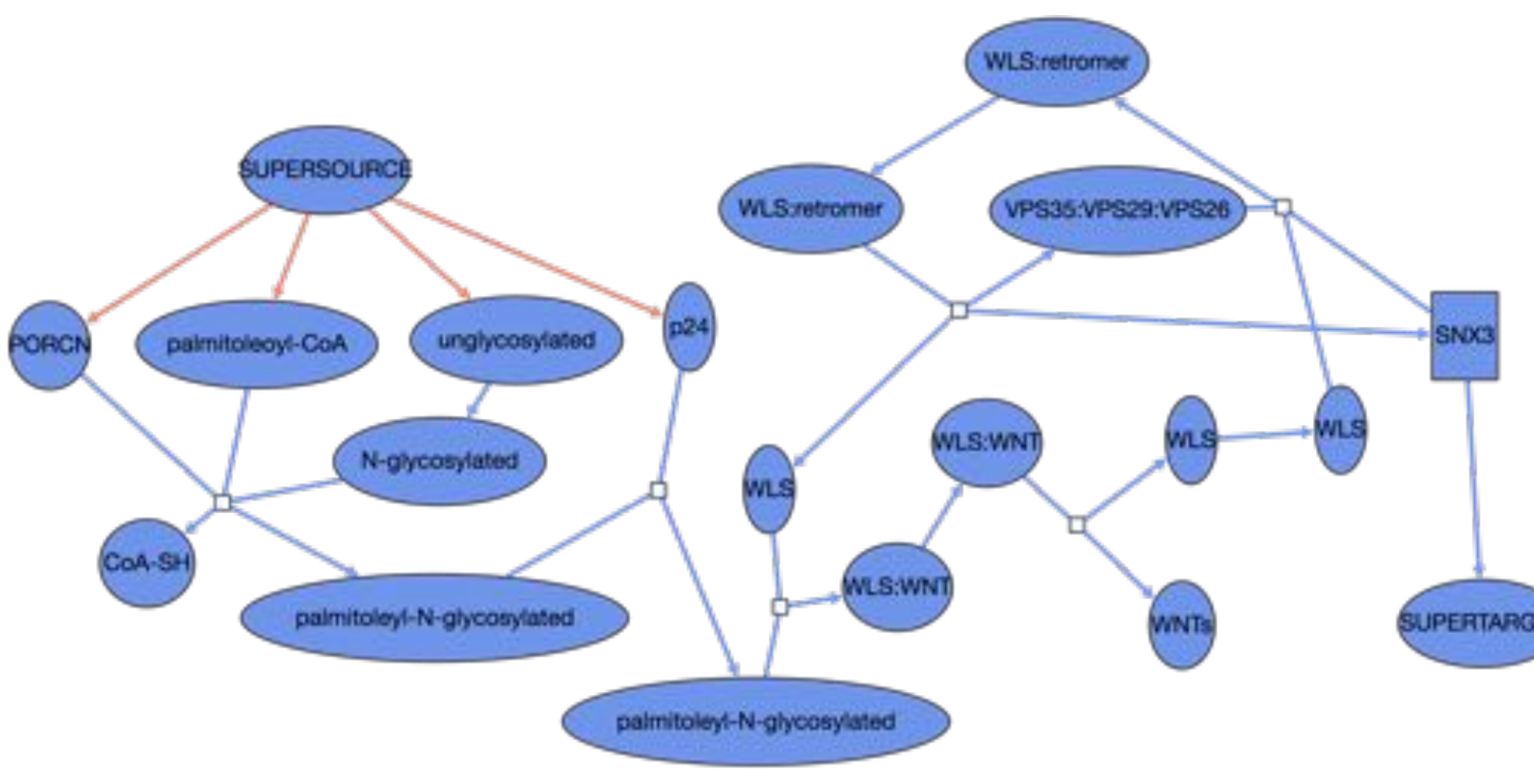
1. **Allowing simple cycles** using a flow-based series of linear constraints (Panel E). Intuitively, we need to ensure that there exists *some* simple path that connects s and t. Introduce flow variables $f_e \in [0, 1]$ for every hyperedge, and add the following constraints:
   1.1 Outgoing flow from s and incoming flow to t must equal one.
   1.2 The flow f_e for hyperedge e must be less than or equal to the incoming flow from hyperedges into the tail nodes T_e. These balance-like constraints handle the fact that a hyperedge flow f_e travels through many head nodes H_e
   1.3 Only active edges (e.g. where $\alpha_e = 1$) can have positive flow f_e.
2. **Adding plausible "source" nodes** by augmenting the original hypergraph $\mathcal{H}$ (Panel F). Define $\mathcal{H}' = (V, E \cup C)$ from the original hypergraph $\mathcal{H}$ by adding a set C of "cheat" hyperedges that contain a single node in the tail:

$$C = \left\{ (\{v\}, H_e)\ \forall\, v \in T_e;\ e \in E \text{ where } |T_e| > 1 \right\}$$

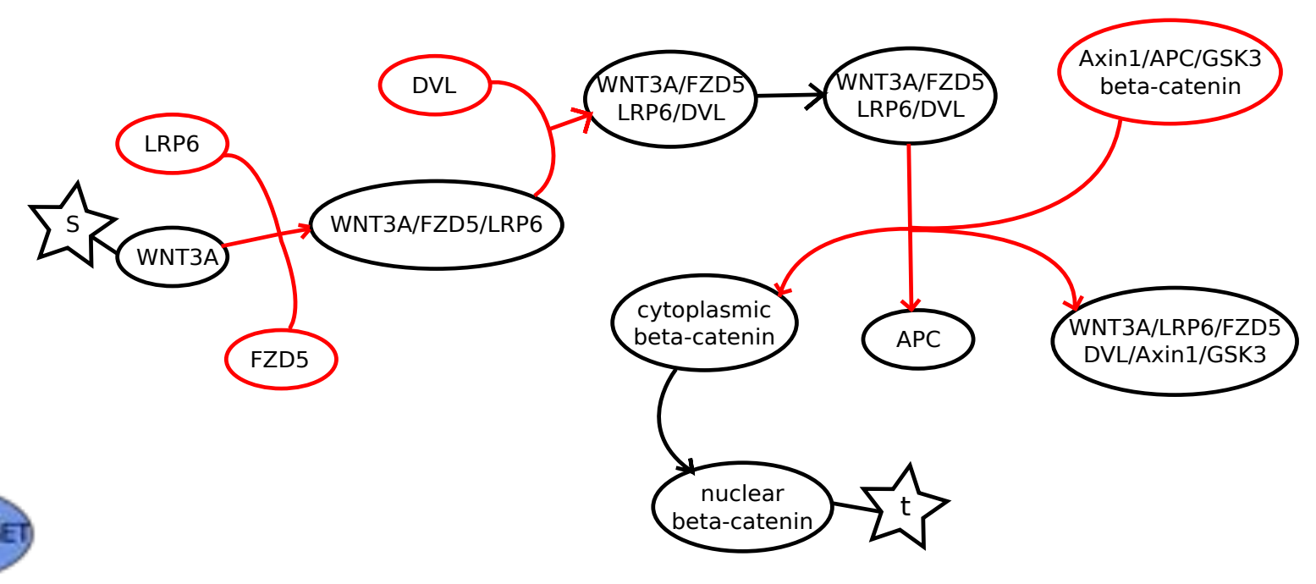
These "cheat" hyperedges are penalized in the objective, but allow nodes in the tail of hyperedges to be ignored in the B-connectedness constraint. We bound the number of cheats allowed in the solution.

Preliminary Results

WLS recycling loop in the shortest B-Hyperpath from a super source to SNX3.



Shortest hyperpath from WNT3A to nuclear β-catenin from NCI-PID [6] using three cheats (red).

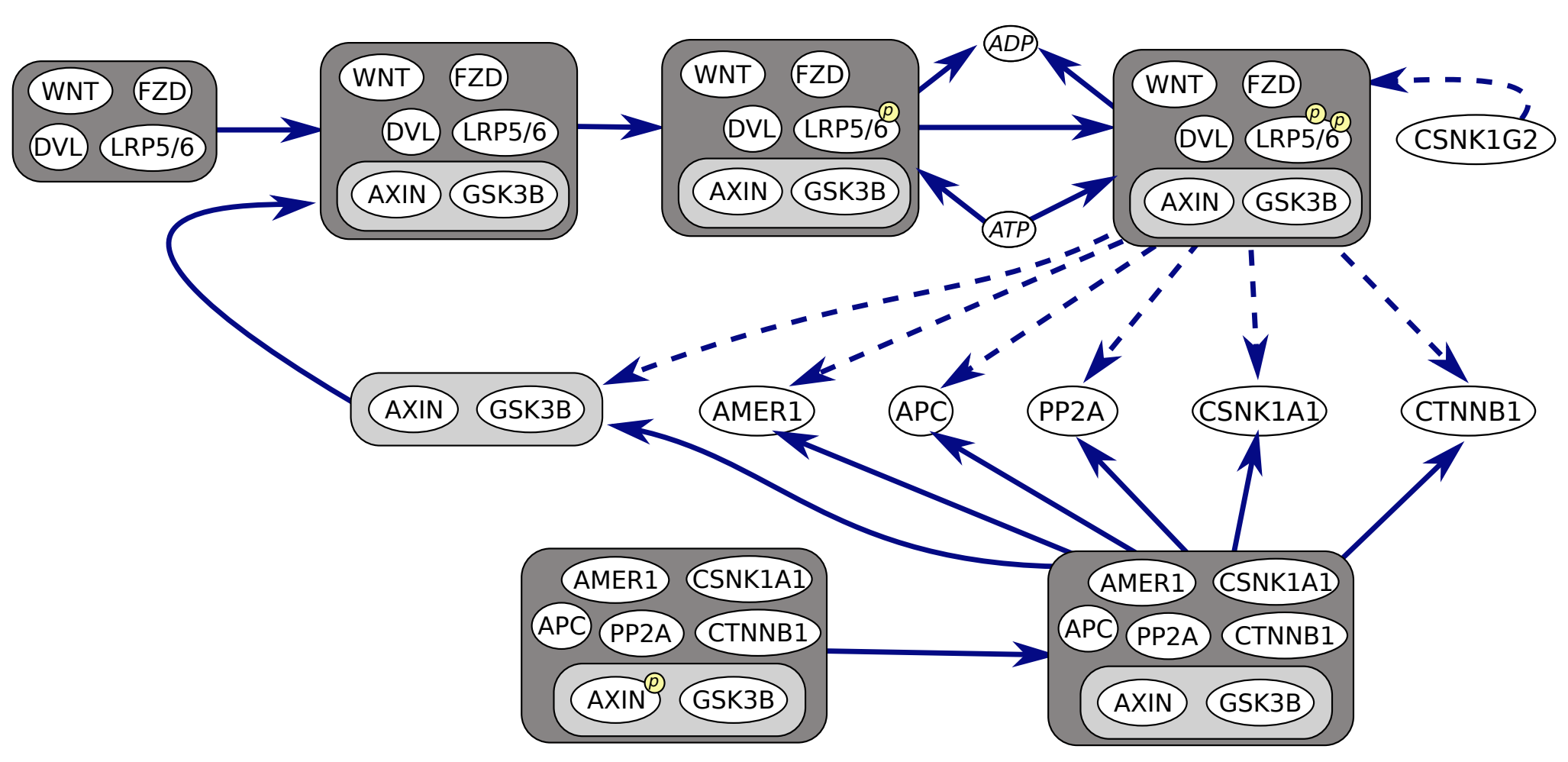


Comparison to Graph Representations

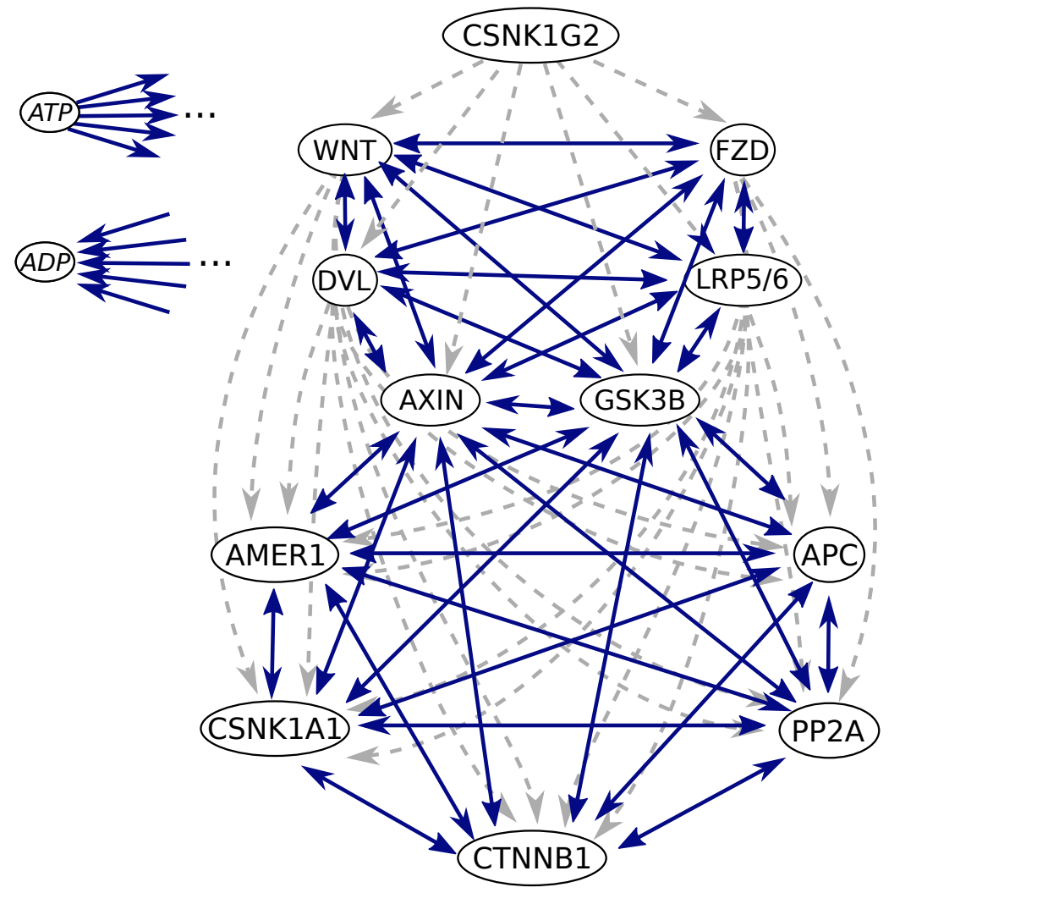
Reactome Signaling Pathways [3] (via PathwayCommons [1]

	Search Term	WNT5A	CTNNB1	WNT	ALL
	# Pathway Hits	16	41	125	2,148
Directed Hypergraph	# Nodes	225	401	850	12,559
	# Hyperedges	91	168	407	6,398
Graph with Complexes	# Nodes	225	401	850	12,559
	# Edges	298	559	1,327	22,815
Standard Graph	# Nodes	578	1,034	1,886	19,281
	# Edges	44,785	126,022	148,578	1,266,285

Graph-With-Complexes Representation



Graph Representation



Future Work

In future work, we plan to **integrate gene expression data** from cancer studies to identify dysregulated sub-hypergraphs. While the returned structures are not necessary hyperpaths, they will be critical for exploring dysregulated signaling.

References & Acknowledgements

[1] Cerami et al. *Pathway Commons, a web resource for biological pathway data.* Nucleic Acids Research, 2011.

[2] Demir et al., *The BioPAX community standard for pathway data sharing.* Nat. Biotech. 2010.

[3] Fabregat et al., *The Reactome pathway knowledgebase.* Nucleic Acids Research, 2016.

[4] Ritz et al., *Signaling Hypergraphs.* Trends in Biotechnology, 2014.

[5] Ritz et al., *Pathway analysis with signaling hypergraphs.* IEEE/ACM TCBB, 2015.

[6] Schafer et al., *PID: the Pathway Interaction Database.* Nucleic Acids Research, 2009.

We thank T. M. Murali at Virginia Tech for insightful discussions about signaling hypergraphs.

Reed College, Portland, Oregon   aritz@reed.edu   http://www.reed.edu/biology/ritz/