# Discovering Pathways by Orienting Edges in Protein Interaction Networks

## Supporting Information

Anthony Gitter, Judith Klein-Seetharaman, Anupam Gupta, and Ziv Bar-Joseph

Here we present additional theoretical and experimental results, algorithm pseudocode, and details regarding both the datasets used and our implementations. Source code for our algorithms is available at http://www.sb.cs.cmu.edu/OrientEdges.

# Supporting Methods

### Calculating the average path length in gold standard pathways

In the Introduction we stated that on average, pathways in KEGG and the *Science Signaling* Database of Cell Signaling contain only 5 edges between a target and its closest source. To calculate this value, we used nodes without parents as sources and nodes without children as targets. We considered the KEGG MAPK pathways' targets, *Science Signaling* pheromone pathway's targets, and *Science Signaling* high osmolarity glycerol (HOG) pathway's targets separately. For each of these three pathways, or groups of pathways in the case of the KEGG MAPK network, we searched for the shortest path from any source to each target using only edges in that pathway or group of pathways. Because of overlapping nodes in the KEGG MAPK pathways, the closest source to a given target was sometimes not a source in the same functional pathway as that target (e.g. Sho1, a source in the KEGG HOG pathway, was the closest source to Far1, a target in the KEGG pheromone pathway).

### Extended MAX-DI-CUT reduction

We begin our proof that reduction from MAX-DI-CUT (1) shows MEO cannot be approximated within 12/13 for any $k \geq 2$ by recapitulating the transformation described in the main text. To reduce a MAX-DI-CUT instance $G = (V, E)$ to MEO, we add a new node $C$ and construct an undirected graph $H = (V', E')$, where $V' = V \cup \{C\}$ and $E' = (v', C)$ for all $v' \in V$. All edges and vertices in $H$ are given a weight of 1 so that for all $p$, $w(p) = 1$. For every directed edge $(u, v)$ in the MAX-DI-CUT instance, we create a source-target pair $< u, v >$ in the MEO instance.

Any orientation of the MEO instance that achieves a score $m$ can be used to construct a solution to the MAX-DI-CUT problem that places $m$ directed edges across the cut. In the orientation, if an edge $(v', C)$ is oriented toward $C$, then place the corresponding vertex $v$ in the set $A$. For all edges $(v', C)$ oriented away from $C$, include $v$ in the set $B$. All paths in the MEO instance consist of two edges $(v'_1, C), (C, v'_2)$. Thus, if a path is satisfied the orientation of these edges must be directed $v'_1 \rightarrow C$ and $C \rightarrow v'_2$. As a result, in every satisfied path the vertex $v_1$ in $G$ corresponding to the source $v'_1$ will be in the set $A$ and every vertex $v_2$ in $G$ corresponding to the target $v'_2$ will be in the set $B$. Furthermore, because source-target pairs were derived from the directed edges in $G$, we know that there is a unique directed edge $(v_1, v_2)$ in $G$ that corresponds to the source-target pair. In addition, there is only one path connecting a particular source-target pair in *H*. It follows that for every satisfied source-target path, the corresponding directed edge will begin in *A* and end in *B* so that if there are *m* satisfied paths in *H* there will be *m* edges across the cut in *G*.

Similarly, any partitioning of the vertices in *G* will yield a unique orientation in *H*. For every vertex *v* in *A*, orient $(v', C)$ toward *C*. For every vertex *v* in *B*, orient $(v', C)$ toward $v'$. Using this procedure, any cut of *m* edges will produce an orientation with *m* satisfied paths because each directed edge across the cut will correspond to a source-target pair and the path connecting that pair will have its first edge oriented toward *C* and its second edge oriented away from *C* toward the target. Consequently, the number of edges across the cut in the optimal solution to the MAX-DI-CUT problem is equal to the objective function score of the optimal MEO orientation.

Because the problems have the same optimal solution and an orientation that achieves a score *m* can be used to construct a vertex partitioning that places *m* directed edges across the cut, an algorithm that achieves an *r*-approximation for MEO can achieve an *r*-approximation for MAX-DI-CUT as well. MAX-DI-CUT cannot be approximated within 12/13 (2), therefore MEO is inapproximable within 12/13. The reduction only requires paths of length 2 so this result holds for any $k \geq 2$.

## MIN-k-SAT approximation algorithm

As described in the main text, MIN-k-SAT is an optimization version of the traditional SAT problem in which weighted disjunctive clauses of at most *k* literals are given and the objective is to find the assignment to all variables that minimizes the sum of the weights of the satisfied clauses. We now describe how to use MIN-k-SAT to approximate MEO.

For each edge $(u, v)$ in the MEO graph, the MIN-k-SAT instance will have a corresponding edge variable $x_{uv}$. The goal is to orient the edge by assigning a value of 1 (*u*→*v*) or 0 (*v*→*u*) to that edge. We first enumerate all simple paths of length at most *k* via depth first search. Then for each path, we construct a disjunctive clause that has the same weight as the path. The edge variables in the clause are given by the edges used by the path. If a path uses an edge in its canonical positive orientation (*u*→*v*), the negation of the edge variable appears in the clause. Otherwise the edge variable appears in the clause but is not negated. Observe that there is a one-to-one mapping between clauses that are satisfied and paths that contain at least one edge oriented in the wrong direction and will not be satisfied. The constructed MIN-k-SAT instance therefore aims to minimize the sum of the weights of the paths that are not satisfied (which, of course, maximizes the sum of those satisfied).

Figure S1 illustrates the transformation for an instance with two paths: $p_1 = (1,3),(3,4),(4,6)$ with $< s_1, t_1 >=< 1,6 >$ and $p_2 = (5,4),(4,3),(3,2)$ with $< s_2, t_2 >=< 5,2 >$. All vertices have been assigned an index, and the canonical positive orientation of each edge is the orientation toward the vertex with the larger index. Because $p_1$ uses all edges in the positive direction, all edges variables in clause 1 are negated. Thus, if any of these three edges are oriented in the negative direction (toward the lesser index), clause 1 will be satisfied and the objective function will be penalized by $w(p_1)$.

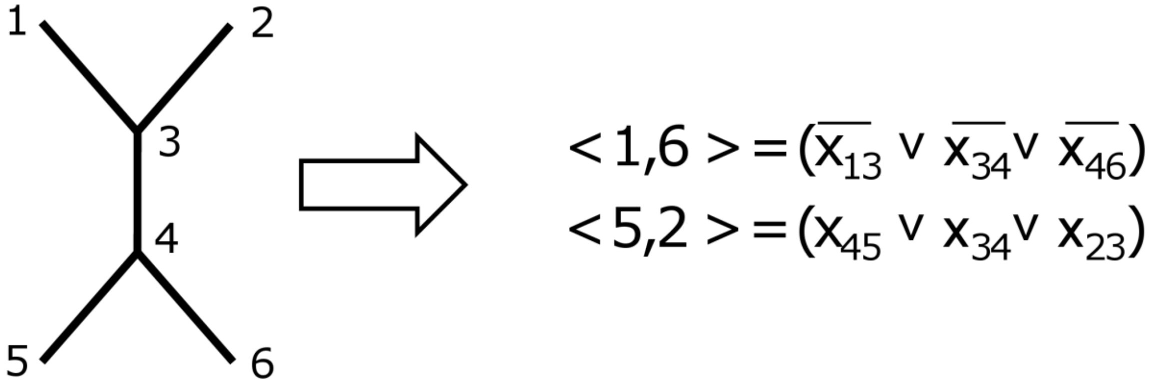

1
2
3
4
5
6
$<1,6> = (\overline{x_{13}} \vee \overline{x_{34}} \vee \overline{x_{46}})$
$<5,2> = (x_{45} \vee x_{34} \vee x_{23})$

**Figure S1.** Formulating an MEO instance as a MIN-k-SAT problem. Each path connecting a source-target pair becomes a disjunctive clause. The literals in the clause are given by the edges in the path.

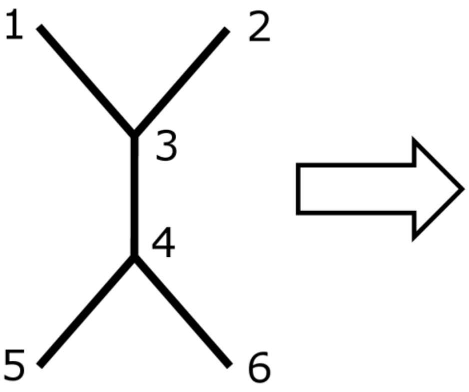
The constructed MIN-k-SAT instance can be solved using an algorithm by Bertsimas *et al.* (3). The MIN-k-SAT instance is formulated as an integer program and then relaxed as a linear program (LP). The authors present a dependent randomized rounding scheme for transforming the LP solution into variable assignments for the MIN-k-SAT problem. We use lp_solve (http://lpsolve.sourceforge.net/5.5/), an open-source LP solver based on the revised simplex method, in our implementation of the MIN-k-SAT-based approximation algorithm.

While the optimal solution for the weighted MIN-k-SAT problem will provide the optimal solution to our problem, the $\frac{1}{2} * \left(\frac{2^k}{2^k - 1}\right)$-approximation ratio for the specific algorithm by

Bertsimas *et al.* does not hold for MEO. This is due to our transformation of the MEO maximization problem into a minimization problem; the optimum of the weighted MIN-k-SAT instance is the sum of the weights of all paths *minus* the optimum of the MEO instance. Pseudocode for the complete MIN-k-SAT-based approximation algorithm can be found in Figure S3. Note that the MIN-k-SAT instance is simplified before running lp_solve by fixing the values of those edge variables that are used in the same direction by all paths and removing those clauses in which all edge variables' values are already fixed.

**MAX-k-CSP approximation algorithm**

Rather than minimizing the weights of paths that are not satisfied as in the MIN-k-SAT-based approximation, it is more straightforward to maximize the weights of satisfied paths by using *conjunctive* clauses. The transformation is similar that used in the MIN-k-SAT-based algorithm except that edge variables used in the positive canonical direction by a path are positive in the conjunctive clause and vice versa. Figure S2 shows the transformation using the previously introduced MEO example.



$$<1,6> = (x_{13} \wedge x_{34} \wedge x_{46})$$
$$<5,2> = (\overline{x_{45}} \wedge \overline{x_{34}} \wedge \overline{x_{23}})$$

**Figure S2.** Transforming an MEO instance into MAX-k-CSP. Each path connecting a source-target pair is mapped to a conjunctive clause. As in the MIN-k-SAT transformation, the literals in the clause are given by the edges in the path.

Optimizing the weights of the satisfied conjunctive clauses is an instance of MAX-k-AND, which is also referred to as MAX-k-CSP (constraint satisfaction problem) because the more general MAX-k-CSP can be approximated as well as MAX-k-AND (4). The state of the art MAX-k-CSP approximation (5) does not yield an explicit approximation ratio. However, previous work by Charikar *et al.* (6) provides a $O\left(\frac{k}{2^k}\right)$-approximation ratio for general *k*, and even better special case solutions for *k* equal to 2, 3, and 4 exist as well (7-9). Because the MAX-k-CSP reduction is approximation-preserving, these general and special case theoretical guarantees apply directly to the MEO problem as well, improving the $\frac{1}{2^k}$-approximation guarantee obtained via random orientation.

Although they provide theoretical guarantees, the above MAK-k-CSP approximations are all based on semidefinite programming. Consequently, they do not scale well on large instances (e.g. genome-wide protein-protein interaction networks) and are not typically used in practice. Therefore, to solve the MAX-k-CSP reduction we use toulbar2 (http://mulcyber.toulouse.inra.fr/projects/toulbar2) (10), a branch and bound-based solver, which was by far the best performing solver in the MAX-CSP portion of the Third International CSP Solver Competition (http://www.cril.univ-artois.fr/CPAI08/).

### Approximation algorithm pseudocode

Below we provide pseudocode for the MIN-k-SAT-based approximation algorithm (Figure S3) and local search technique (Figure S4) described in the main text. While it is not necessary to provide pseudocode for the random orientation algorithm, we note that when running this algorithm with local search we initiated a search from multiple random initial orientations (typically 20) and kept the highest scoring result. When running the MIN-k-SAT-based algorithm, we performed the randomized rounding step 100 times and kept the highest scoring result. The MAX-k-CSP-based approximation algorithm is similar to MIN-k-SAT-based algorithm except for the form of the clauses and the solver used so we do not provide separate pseudocode.

```
MinSatApprox(G,ST,R)
        G – the graph
        ST – source-target pairs
        R – the number of times to perform the randomized rounding procedure

Perform a depth first search to enumerate simple paths connecting source-target pairs
Randomly orient edges not on any path
Construct the MIN-k-SAT instance as described previously
Identify edges that are used in the same direction by all paths and fix their values
Identify paths whose edges are all fixed and remove the corresponding MIN-k-SAT clauses
Construct the LP instance from the MIN-k-SAT instance as in Bertsimas et al.
Solve the LP with lp_solve
for(R iterations)
        Use the dependent randomized rounding procedure to obtain an integer solution
        Assign edge directions according to the integer solution
Return the oriented network with the highest objective function value
```

**Figure S3.** Pseudocode for the approximation algorithm based on MIN-k-SAT, which relies on the MIN-k-SAT algorithm by Bertsimas *et al*. (3).

```
LocalSearch(G,ST,O)
        G – the graph
        ST – source-target pairs
        O – the initial orientation

while(there exists an edge that will improve the objective function score if it is flipped)
        for(each edge)
                delta(edge) := score after flipping edge –  score using edge's current orientation
        Flip the edge for which delta(edge) is largest
```

**Figure S4.** Pseudocode for the local search algorithm that is run after executing the random orientation, MIN-k-SAT, or MAX-k-CSP algorithm.

## BioGRID network and experimental types

The BioGRID (11) network used was based on the version 2.0.51 release. Only physical interactions were used, and all interactions inferred from co-localization were ignored. The confidence scores for the 15 types of experiments are below in Table S1.

**Table S1.** Confidence scores for each type of experiment used to compile the PPI network.

| Experiment type | Confidence score |
|---|---|
| Affinity Capture-Luminescence | 0.5 |
| Affinity Capture-MS | 0.5 |
| Affinity Capture-RNA | 0.7 |
| Affinity Capture-Western | 0.5 |
| Biochemical Activity | 0.5 |
| Co-crystal Structure | 0.99 |
| Co-fractionation | 0.7 |
| Co-purification | 0.7 |
| Far Western | 0.5 |
| FRET | 0.7 |
| PCA | 0.3 |
| Protein-peptide | 0.7 |
| Protein-RNA | 0.3 |
| Reconstituted Complex | 0.3 |
| Two-hybrid | 0.3 |

All PPI edges with confidence scores less than 0.6 were removed from the version of the network used in our primary evaluations, referred to as the high-confidence BioGRID interaction network. The MEO algorithms are not dependent on our method for calculating PPI edge reliability scores, thus other types of PPI confidence scores (12-14) could be used.

## Gold standard signaling pathways

All gold standard pathways were obtained from KEGG (15) and the *Science Signaling* Database of Cell Signaling (http://stke.sciencemag.org/cm/), and all available yeast signaling pathways in these databases were used to construct the gold standard. Although other yeast signaling pathways such as the TOR pathway have been described in the literature, we chose only pathways from established signaling databases. From KEGG, we used the yeast MAPK signaling pathway (http://www.genome.jp/kegg-bin/show_pathway?sce04011), which is composed of the pheromone, hypotonic shock, high osmolarity, and starvation pathways, as well as the phosphatidylinositol signaling system (http://www.genome.jp/kegg/pathway/sce/sce04070.html). From the Database of Cell Signaling, we obtained the HOG (16), filamentous growth (17), and pheromone signaling (18) pathways. When creating the gold standard, we kept all of the above individual pathways separate (even those that appeared in both databases but described the same path) so that when validating our algorithms' predicted pathways in the gold standard network, matches had to come entirely from one single pathway.

To explore the overlap between the gold standard network and the PPI network, we calculated how many simple paths of a given length could be enumerated by beginning a depth first search at every node in the gold standard network. We then determined how many of those paths were also present in the PPI network (Table S2). Paths of length 1 are a single edge, thus less than 50% of the edges in the gold standard pathways are also in our PPI dataset. This relatively low overlap indicates that our algorithms will not be able to recover certain paths in the gold standard simply because the requisite edges are not in the PPI network. The longest signaling pathways in the gold standard network contain more than 5 edges. However, we are only interested in calculating the overlap for paths with 5 or fewer edges because we set $k = 5$ when running our algorithms due to the exponential growth in the number of paths as $k$ increases (see Table S7). Therefore, even though the final oriented network will contain longer directed paths, the evaluation in Table 2 is based on paths containing 5 edges.

**Table S2.** The number of gold standard paths with the specified length and the number of those that appear in the PPI network. Paths may use undirected edges in either direction and may start and end at any vertex.

| Length | Gold standard paths | Gold standard paths in PPI network |
|---|---|---|
| 1 | 286 | 134 |
| 2 | 650 | 163 |
| 3 | 1549 | 176 |
| 4 | 3814 | 195 |
| 5 | 8653 | 198 |

Sources and targets were selected by inspecting the pathway diagrams on the databases' websites as described in Materials and Methods. Using those criteria, we chose the following sources and targets (Table S3) for our gold standard pathway evaluation in the main text. Some of the targets, such as Hog1 and Fus3, are not actually the terminal points of the signaling pathway, but rather vital members of the response that are considerably downstream from the sources. Proteins in Table S3 marked with an asterisk were present in the larger PPI networks, but not the high-weight BioGRID network, effectively making our evaluations on that network use 15 sources and 14 targets. For the timing evaluation in Table 1 of the main text, we replaced the source YPL187W with YBR097W (Vps15) and the target YIR019C with YLR452C (Sst2).

**Table S3.** Gold standard sources and targets

| Source standard name | Source systematic name | Target standard name | Target systematic name |
|---|---|---|---|
| SLN1 | YIL147C | CDC42 | YLR229C |
| YCK1 | YHR135C | HOG1 | YLR113W |
| YCK2 | YNL154C | STE7 | YDL159W |
| SHO1 | YER118C | STE20 | YHL007C |
| MF(ALPHA)2 | YGL089C | DIG2 | YDR480W |
| MID2 | YLR332W | DIG1 | YPL049C |
| RAS2 | YNL098C | PBS2 | YJL128C |
| GPR1 | YDL035C | FUS3 | YBL016W |
| BCY1 | YIL033C | STE5 | YDR103W |
| STE50 | YCL032W | GPA1 | YHR005C |
| MSB2 | YGR014W | MSN1* | YOL116W* |
| SIN3 | YOL004W | FKS2 | YGR032W |
| RGA1 | YOR127W | FUS1 | YCL027W |
| RGA2 | YDR379W | STE12 | YHR084W |
| ARR4 | YDL100C | SWI4 | YER111C |
| MF(ALPHA)1* | YPL187W* | FLO11* | YIR019C* |

## Implementation details for comparisons with previous work

When implementing the MTO algorithm for general trees described by Medvedovsky *et al.* (19), we used the randomized version of StarMTO, as described in the paper, rather than derandomizing the algorithm via the method of conditional expectation. A precondition of the MTO algorithm is that all cycles in a graph have been contracted, but to compare with our orientation algorithms, which restrict path length, cycles had to be expanded after running MTO.

After restoring the original cycles in the graph, all edges in a cycle were oriented in the same direction (chosen arbitrarily) such that reachability was maintained.

Local search for MTO is performed before expanding cycles and strives to maximize the MTO objective function, the number of reachable source-target pairs. The search is very similar to the MEO local search depicted in Figure S4. At each iteration the edge flip that yields the greatest improvement in the objective function is performed until there are no flips that will increase the score.

The unoriented edge selection algorithm was implemented as described by the authors (20) except for the few minor differences noted here. Instead of using depth first search to prune all edge that are not on paths of length 6 to 9 edges, our implementation uses breadth first search to prune all edges that are more than 9 edges away from a source. One of the linear program constraints is that all known members of the signaling network must be present in the subset of selected edges, and we take the only the sources and targets to be the known members. Zhao *et al.* do not specify how to round the linear program solution to obtain a feasible integer solution when the linear program solution does not happen to be integer. Like the authors, we find that nearly all variables in the linear program do have integer values in the optimal solution, thus the choice of a rounding scheme will have low impact. Therefore, we choose a trivial rounding scheme and set all non-integer variables to 1. Note that strictly speaking this could lead to an infeasible integer program solution. In particular, the constraint that each selected vertex must be connected to at least 2 selected edges could be violated, but in our evaluation there were at most 12 non-integer variables in the linear program solutions. As suggested by the authors, we used the graph density (average edge weight) to select λ, searching over all values of λ from 0 to 1 with a step size of 0.05.

For unoriented edge selection, local search seeks edges to add or remove edges from the set of selected edges instead of flipping directed edges. In order to make the unoriented edge selection local search have similar complexity to the MEO local search, we constrain it to only add edges whose endpoints are already in the set of selected vertices and only remove edges whose endpoints will still be of degree 2 or more in the selected subnetwork after the removal. Without these restrictions, the local search becomes substantially more complex because an addition or removal could violate the constraint that each selected vertex has degree of at least 2 in the selected subnetwork. While it is possible to extend the search to find a feasible solution after violating this constraint, such a search would no longer correspond to making small, local changes to the linear program solution. The local search is performed for each possible value of λ, and the graph density for each λ incorporates the results of the search.

Physical Network Models (PNM) (21) software was downloaded from http://www.cellcircuits.org/Yeang2005/index.shtml. This method requires that the last interaction in a pathway is a directed protein-DNA edge. Therefore, in order to consider paths consisting only of PPIs, pseudo protein-DNA edges were added from each target vertex to a new unique target vertex. The max path length parameter was set to 6 in order to find pathways of at most 5 PPI edges plus a pseudo protein-DNA edge. The PNM software requires knockout data to specify cause-effect pairs. Because we wish to find pathways using only PPI data, we constructed a small simulated knockout dataset that only contained our sources and targets of interest.

When running a test case with 16 sources and 16 targets, the software was quite far from termination after 10 days. To give a rough sense of its progress after this much time elapsed, we checked its log to see how many of the variables in the factor graph were still not fixed. After the first iteration, 2591 variables were not fixed and after 10 days 2126 remained unfixed. The time to fix a single variable remained roughly constant over this time, suggesting the algorithm may have taken nearly 2 months to terminate had we let it continue.

The supplemental material of Yeang *et al.* (21) provides the insight into PNM's strategy for resolving cases where there are multiple MAP configurations of the factor graph. If the initial application of the max product algorithm does not unique fix all variables, PNM will recursively select one variable that has not yet been fixed, fix the variable to one of its multiple optimal values, and rerun the max product algorithm conditioned on the new fixed state of this variable. As we observed, each successive call to the max product algorithm can be quite time consuming, which is especially detrimental if max product is unable to fix additional variables at each recursive call.

Therefore, we added a new base case to this recursive process. If not all variables have been fixed after running the recursive algorithm for some time *t*, we arbitrarily fix all remaining variables to one of their optimal values. PNM initiates this recursive process separately for the variables that correspond to the regulatory effect of the physical interactions and the variables tied to the direction of PPI edges, and we use the same timeout *t* for both sets of variables such that PNM is allowed to spend up to 2*t* altogether making recursive calls to the max product algorithm. The time needed to enumerate source-target paths, construct the factor graph, and run the initial application of the max product algorithm does not count toward the timeout.

Nevertheless, even when we set *t* = 12 hours, PNM was unable to generate predictions we could compare against. After running for over 24 hours, when the timeout forced the algorithm to terminate, it predicted that there were no active source-target paths in the network and thus no active edges in the network. PNM not only directs edges, but also determines whether the edge is truly present (active) or not. Because it did not place any edges in the network, there was nothing to evaluate. We note that our implementation of the timeout was the not the problem because when running PNM with smaller test cases, it was able to find active paths and edges after the timeout forced early termination. Thus, we believe the complex factor graph representation used in PNM is simply unable to scale to a genome-wide analysis if paths are allowed to contain 5 PPI edges.

### Linear program upper bound

A linear program relaxation of the following integer program was used to obtain an upper bound during testing with simulated sources and targets:

**maximize** $\sum_{p_j \in P} w(p_j) * p_j$

**subject to** $p_j \le e_i \qquad \forall e_i \in E_j^+$

$p_j \le 1 - e_i \qquad \forall e_i \in E_j^-$

$p_j, e_i \in \{0,1\}$

where $P$ is the set of all simple source-target paths with length at most *k*, $e_i$ are the edge variables, $w(p)$ is the weight of a path, $E_j^+$ is the set of all edges used in their positive canonical direction in path $p_j$ (as defined in the MIN-k-SAT algorithm description), and $E_j^-$ is the set of all edges used in their negative canonical direction in path $p_j$. If any edge in the set $E_j^+$ has the value 0, which corresponds to being oriented in the negative direction, the path cannot be satisfied and must have the value 0 as well. Likewise, if any edge in the set $E_j^-$ has the value 1 the path cannot be satisfied and must have the value 0.

This formulation provides an exact representation of MEO, and consequently the integer program's optimal solution is equal to the maximum MEO objective function value. The optimal solution to the LP relaxation of this integer program provides an upper bound of the optimal MEO score. This is because the optimal orientation corresponds to an integer solution to the integer program.  That integer solution is a valid solution to the LP, which means the maximum LP value cannot be lower than the value obtained by using that solution.

As an aside, we note that we tried using lp_solve (http://lpsolve.sourceforge.net/5.5/) to solve this integer program formulation of MEO directly, but it was not possible for instances involving the real biological network.

## Supporting Results

### Comparison of different PPI databases

To determine which PPI network to use in our analysis and whether the weighting scheme helps, we looked at three popular PPI databases: BioGRID, IntAct (22) (downloaded March 29, 2010) and MINT (23) (downloaded March 29, 2010). Table S4 shows the number of interactions in these databases and the overlap between them.

**Table S4.** The overlap between the BioGRID, IntAct, and MINT PPI-databases was calculated for both the full set of BioGRID interactions and the subset of high-weight interactions.

| Database(s) | Number of interactions (high-weight BioGRID interactions) | Number of interactions (all BioGRID interactions) |
|---|---|---|
| IntAct only | 31626 | 27204 |
| MINT only | 6458 | 2465 |
| IntAct and MINT | 11390 | 8177 |
| BioGRID only | 3325 | 16429 |
| BioGRID and IntAct | 2150 | 6572 |
| BioGRID and MINT | 785 | 4778 |
| BioGRID, MINT, and IntAct | 4685 | 7898 |
| Total unique interactions | 60419 | 73523 |

Although BioGRID only contains a fraction of the total yeast PPI and our high-weight network contains only a subset of all BioGRID edges, we found that the high-weight BioGRID network was nevertheless the best at recovering known gold standard pathways, and so it was the network used in our analysis described in the main text. In Table S5, we show the number of top-ranked pathways from the random algorithm with local search that correspond to gold standard pathways. 20 random restarts were used, and the results for the search that gave the highest objective score are reported. One network contained all unweighted interactions from all three databases. Another used only the edges appearing in multiple databases and assigned a weight of 0.95 to edges in all three databases and a weight of 0.75 to edges in any two of the three databases. The full BioGRID network contains all unweighted BioGRID interactions, and the high-weight BioGRID network is the weighted network used in the main text. Because these networks were much larger than the high-weight BioGRID network, we used $k = 4$ and only 7 sources and 7 targets (the first 7 in Table S3). Thus, a predicted path with exactly 5 proteins was considered to match a gold standard if at least 3 of the 5 proteins appear consecutively in a gold standard pathway. Note that this requirement is weaker than the requirement in the main text (4 of 6 matching proteins) and so the results differ from the ones presented in Table 2. Still, as a way of comparing different networks this is a useful setting. It is not possible to rank paths by the path weight or edge weight metrics when using unweighted networks. The weighted BioGRID network enables the recovery of many more gold standard paths than any of the other the networks. In fact, the full unweighted BioGRID network and the network that consists of interactions in any of the three databases do not lead to any gold standard matches.

**Table S5.** The high-weight BioGRID network is the best choice for recovering known signaling pathways. Note that the analysis in this table requires only 3 correct proteins on a path for a match. Thus, while it is suitable for comparison between different PPI databases, it is not directly comparable to the results presented in Table 2 in which we require at least 4 matches.

| Network | Path weight | Max. edge weight | Avg. edge weight | Min. edge weight | Max. edge use | Avg. edge use | Min. edge use | Max. degree | Avg. degree | Min. degree |
|---|---|---|---|---|---|---|---|---|---|---|
| Interactions in any database | - | - | - | - | 0 | 0 | 0 | 0 | 0 | 0 |
| Interactions in multiple databases | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| Full BioGRID | - | - | - | - | 0 | 0 | 0 | 0 | 0 | 0 |
| High-weight BioGRID | 16 | 16 | 16 | 16 | 18 | 8 | 8 | 18 | 0 | 4 |

## Evaluating the PPI network and its weights

To validate our formula for calculating PPI weights and the weights assigned to the experimental types in Table S1, we repeated the evaluation summarized in Table 2 of the main text using the same set of BioGRID edges without edges weights (i.e. all weights were set to 1). Table S6 shows that the top-ranked predicted paths correspond to a greater or equal number of gold standard pathways when using the weighted network instead of the unweighted network. Results from Table 2 are redisplayed here to facilitate the comparison between the weighted and unweighted results. When using the unweighted network, it is not possible to rank paths by the path weight or edge weight metrics because all paths and edges have the same weight of 1. As in the main text, MTO results are averaged over 20 runs, and oriented baseline results (random orientations without local search) are averaged over 1000 runs. The random orientation with search algorithm used 20 random restarts and the run that yielded the highest objective function value after search was used in the evaluation. The MIN-SAT evaluation was performed in the same way, although only 5 restarts (executions of the MIN-SAT solver) were used for the unweighted network. For some of the MTO runs, there were less than 100 paths with exactly 6 vertices so the top-ranked paths were taken to be the entire set of paths of this length.

**Table S6.** The algorithms always recover more or the same number of known signaling pathways when using the weighted PPI network.

| Network | Algorithm | Max. edge use | Avg. edge use | Min. edge use | Max. degree | Avg. degree | Min. degree |
|---|---|---|---|---|---|---|---|
| Weighted | Random + search | 0 | 0 | 40 | 10 | 0 | 0 |
| Unweighted | Random + search | 0 | 0 | 30 | 0 | 0 | 0 |
| Weighted | MIN-SAT | 0 | 0 | 0 | 1 | 0 | 0 |
| Unweighted | MIN-SAT | 0 | 0 | 0 | 0 | 0 | 0 |
| Weighted | MIN-SAT + search | 0 | 0 | 40 | 10 | 0 | 0 |
| Unweighted | MIN-SAT + search | 0 | 0 | 34 | 0 | 0 | 0 |
| Weighted | MAX-CSP | 0 | 0 | 16 | 3 | 0 | 0 |
| Unweighted | MAX-CSP | 0 | 0 | 8 | 0 | 0 | 0 |
| Weighted | MAX-CSP + search | 0 | 0 | 16 | 3 | 0 | 0 |
| Unweighted | MAX-CSP + search | 0 | 0 | 8 | 0 | 0 | 0 |
| Weighted | MTO | 3.0 | 3.0 | 3.0 | 3.0 | 2.8 | 3.1 |
| Unweighted | MTO | 2.3 | 2.3 | 2.6 | 2.2 | 2.2 | 2.9 |
| Weighted | Unoriented edge selection | 20 | 20 | 20 | 20 | 20 | 20 |
| Unweighted | Unoriented edge selection | 2 | 0 | 20 | 13 | 0 | 0 |
| Weighted | Oriented baseline | 0.4 | 0.2 | 3.2 | 4.6 | 0 | 0 |
| Unweighted | Oriented baseline | 0.1 | 0.2 | 3.1 | 1.6 | 0 | 0 |

## Algorithm runtimes and scalability

The MAX-CSP-based algorithm uses toulbar2, which performs an extensive bounded search of the solution space. For large instances, toulbar2 will run for a very long time but outputs intermediate (possibly suboptimal) solutions as it searches. Thus, in these and all subsequent tests we terminated toulbar2 after 3 hours, as seen in Table 1.

Table S7 presents the exponential growth in the number of paths as the path length increases, which is why we did not measure runtimes for cases where there are 7 or more nodes in the pathway ( $k \geq 6$ ). In particular, the MIN-SAT- and MAX-CSP-based algorithms are not well-suited for instances of this size. Furthermore, even though the random orientation with local search is less complex than these two algorithms, calculating the objective function still requires enumerating and storing all possible paths. Consequently, the random algorithm scales to larger values of *k* much better than the MIN-SAT- and MAX-CSP-based algorithms, but does have an upper bound on the maximum value of *k*, which is dependent on the number of network edges, sources, and targets in the instance

**Table S7.** Number of paths as the maximum path length increases.

| Path length | Number of paths |
|---|---|
| 1 | 5 |
| 2 | 86 |
| 3 | 735 |
| 4 | 6357 |
| 5 | $5.627*10^4$ |
| 6 | $6.103*10^5$ |
| 7 | $7.890*10^6$ |
| 8 | $1.218*10^8$ |
| 9 | $2.112*10^9$ |

For each increase in the maximum path length *k*, the number of paths containing *k* or fewer edges grows by roughly one order of magnitude. The number paths were calculated using protein-protein interactions from BioGRID and the sources and targets in Table S3.

### Receiver operating characteristic curves for comparing algorithms

We used receiver operating characteristic (ROC) curves to compare our best-performing orientation algorithm, random orientation with local search, to MTO and unoriented edge selection. We used the sources and targets in Table S3, enumerated all source-target paths of length 6 vertices, and ranked them by the two best performing metrics from Table 2 (path weight and minimum edge use). Our algorithm was run with 20 starting points, and we display the ROC curve for the starting point that gave the highest objective function value after search. For MTO, we similarly chose the best of 5 runs, where the best run is the one with the highest MTO objective function value (the number of reachable source-target pairs). To break ties, we chose the run that had the most paths from sources to targets after expanding the graph. MTO and unoriented edge selection discover far fewer paths than random orientation with search so we also plot ROC curves for up to the 100 top-ranked paths for each algorithm. Different sets of runs were used to generate the path weight and minimum edge use curves. Figures S5 – S8 summarize these results.

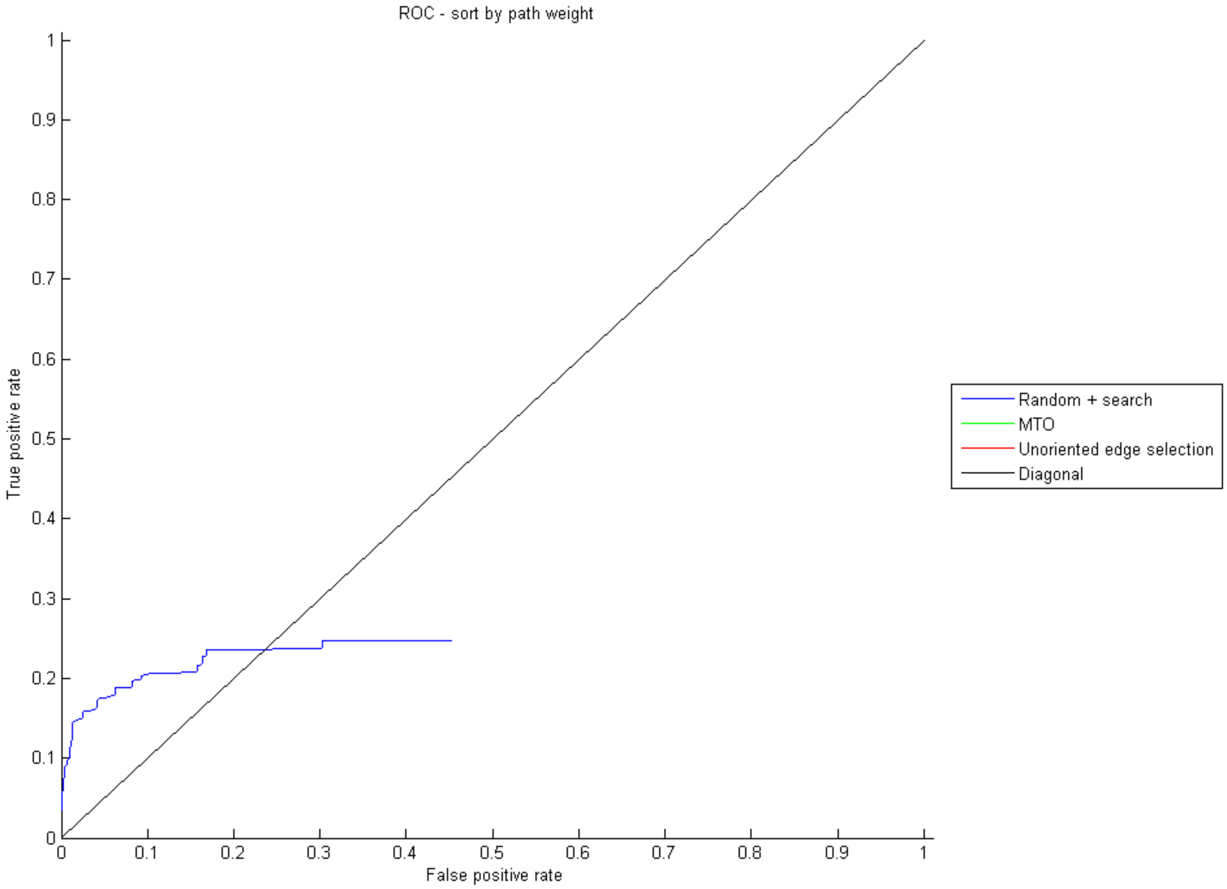


**Figure S5.** ROC curves for random orientation with search, MTO, and unoriented edge selection. Predicted paths were sorted by path weight. The MTO and unoriented edge selection curves are not visible because these algorithms make very few predictions.

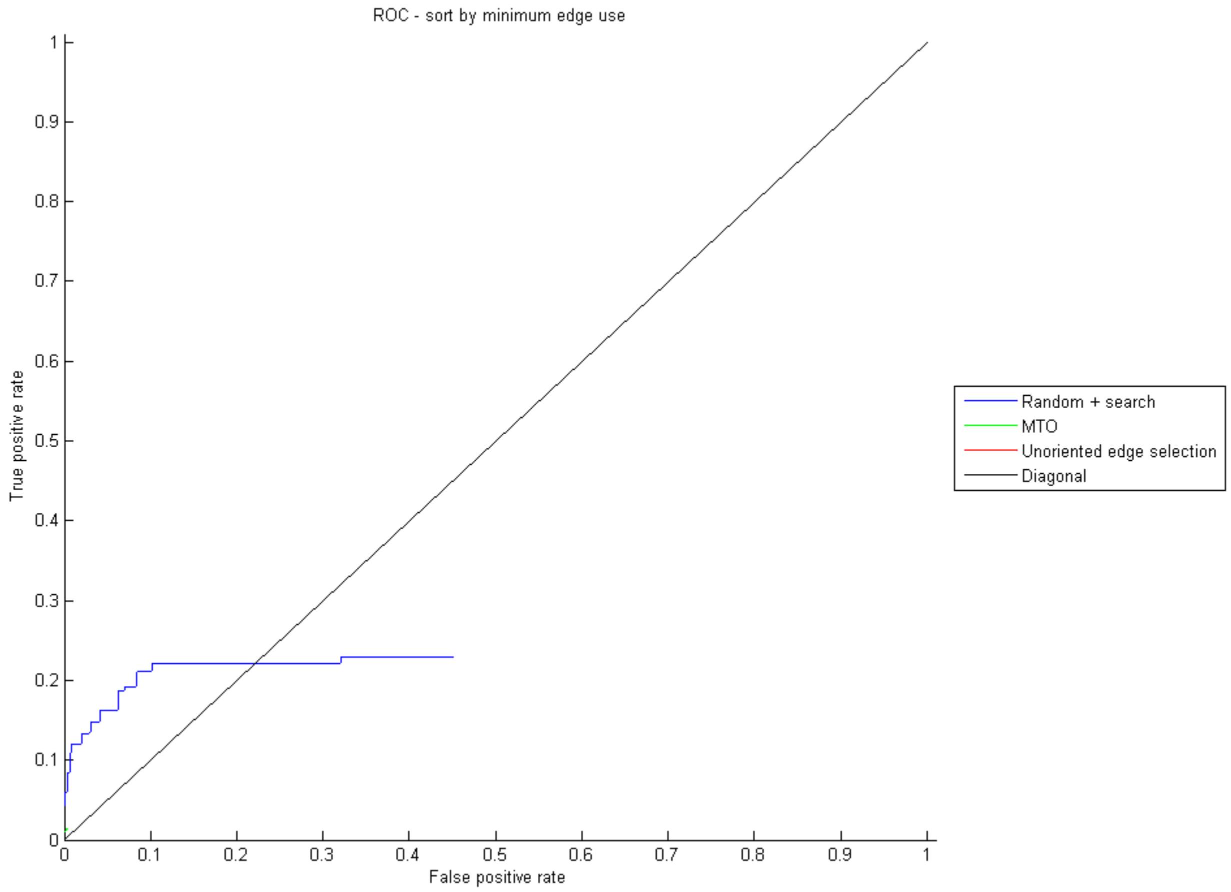


**Figure S6.** ROC curves for random orientation with search, MTO, and unoriented edge selection. Predicted paths were sorted by minimum edge use. The MTO and unoriented edge selection curves are not visible because these algorithms make very few predictions.

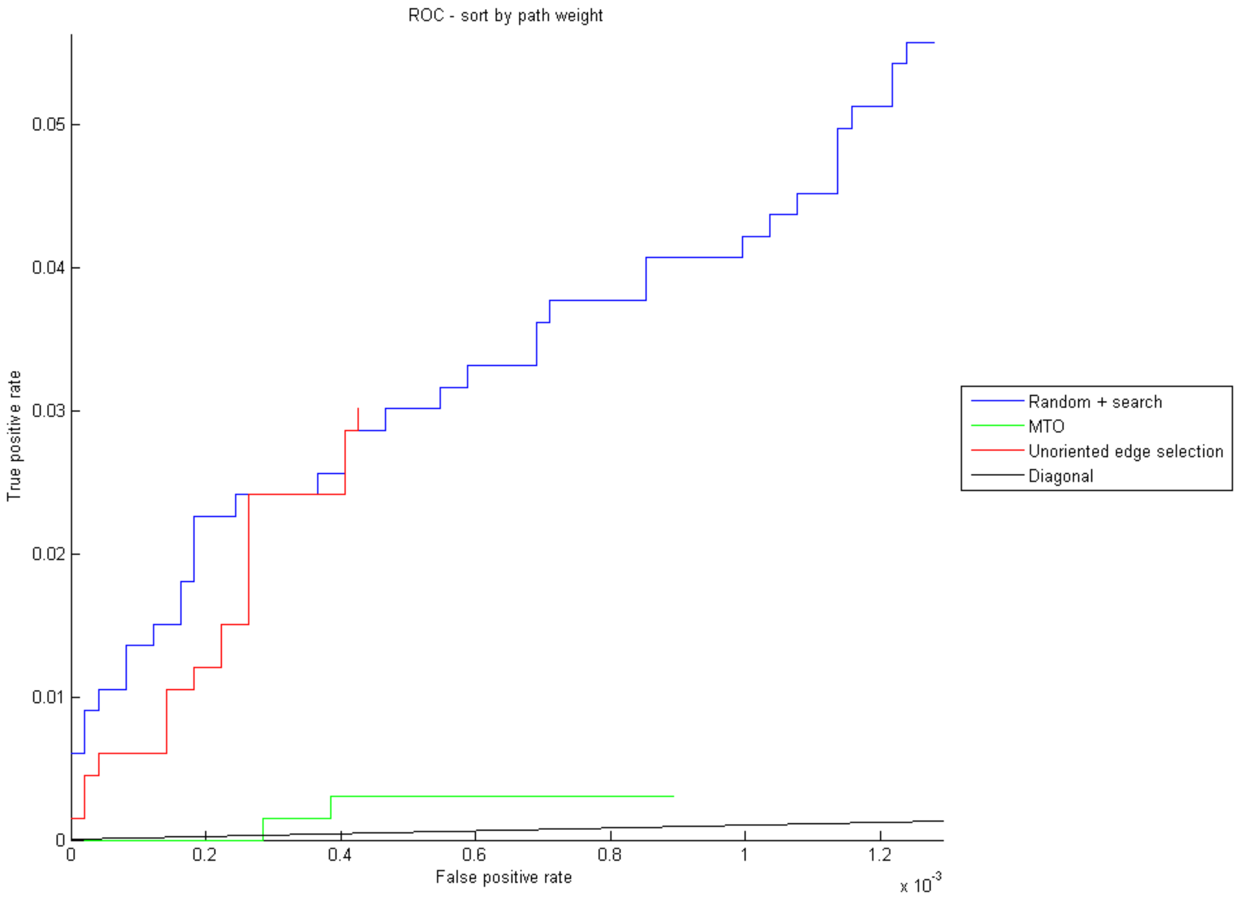


**Figure S7.** ROC curves for random orientation with search, MTO, and unoriented edge selection. Predicted paths were sorted by path weight, and only the top 100 predictions from each algorithm were used. Note that the axes no longer have the same scale.

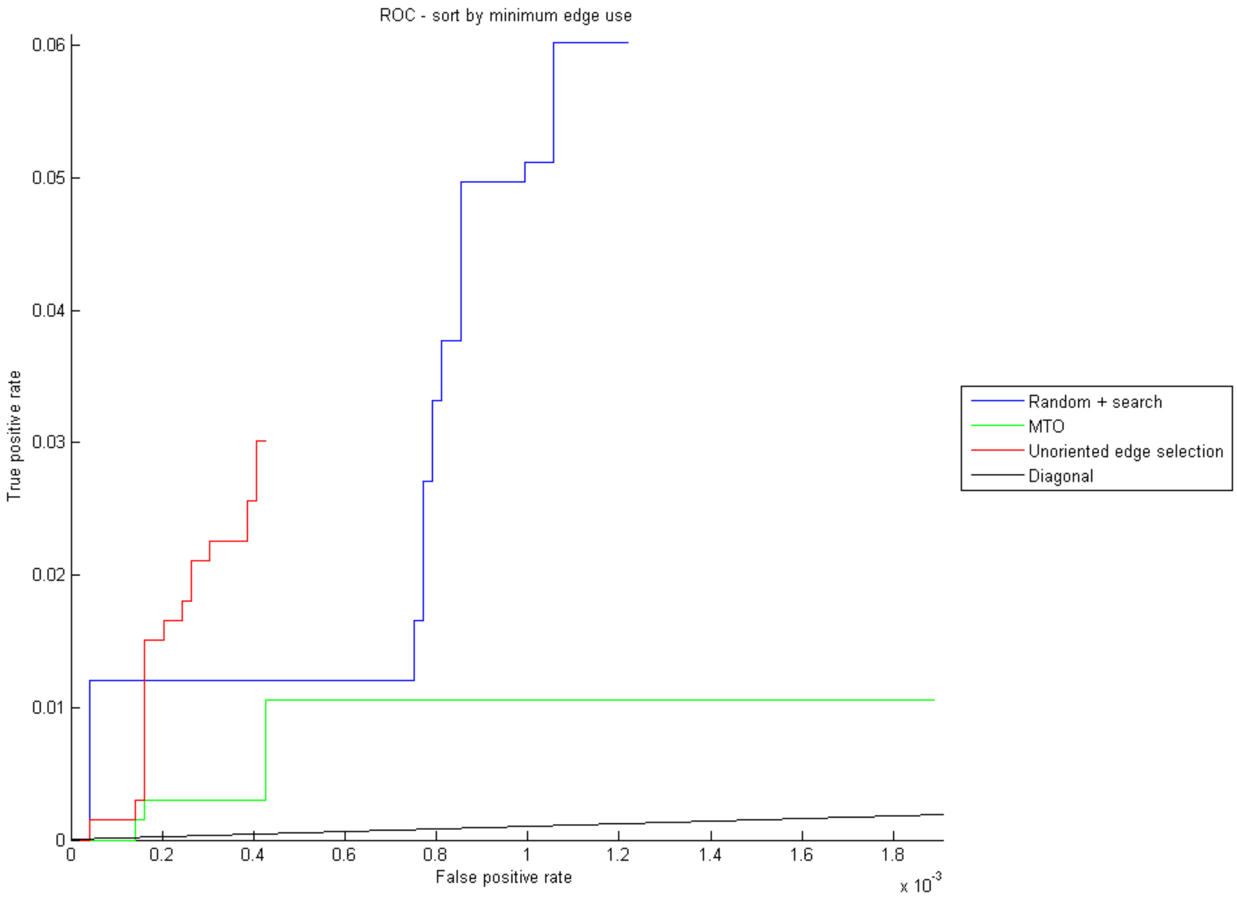


**Figure S8.** ROC curves for random orientation with search, MTO, and unoriented edge selection. Predicted paths were sorted by minimum edge use, and only the top 100 predictions from each algorithm were used. Note that the axes no longer have the same scale.

Figures S5 and S6 support our use of a threshold when evaluating predicted pathways in Table 2 and confirm that the top pathways are more likely to correspond to known signaling pathways than low-ranked pathways. There is clearly a point where the ROC curves level off and very few subsequent predictions are correct according to the gold standard. These figures also highlight a major advantage of our algorithms over MTO and unoriented edge selection. Neither of the competing methods discovers many paths with 6 vertices, whereas our algorithm finds many parallel pathways from the sources to the targets.

Figures S7 and S8 show that although unoriented edge selection does not find many paths, the few paths it does find are generally of high quality. When ranking paths by path weight, its ROC curve is nearly as good as the random orientation with search curves for the few predictions it makes. For the minimum edge use criterion, the MEO local search curve is initially best but the unoriented edge selection curve overtakes it briefly. Ultimately, random orientation with search continues making correct predictions after the undirected method has exhausted all of its paths, thus MEO with local search is better overall. MTO's predictions are worse than those of random orientation with search and undirected edge selection regardless of the ranking metric.

In the figures, none of the curves extend to (1,1), the point where both the true positive rate and false positive rate are 1 because the algorithm has predicted all possible paths. This is because the set of all paths contains paths that use edges in a conflicting manner. Thus, algorithms like random orientation with local search and MTO that orient the network such that all predicted paths use edges in a consistent direction will ignore the subset of paths that use the edges in the opposite direction. Furthermore, even though undirected edge selection allows its identified pathways to use edges in conflicting directions, it discards many edges that are not believed to be relevant to response. Thus, it too does not contain all possible paths in its ranked predictions.

### Evaluation of algorithms using gold standard pathways

Tables S8 and S9 provide the number of top ranked paths that partially match a gold standard pathway when the number of sources and targets is varied.

**Table S8.** Top-ranked predicted pathways for which 4 of the 6 vertices in the path are present consecutively in a gold standard pathway. Pathways were inferred using the first 3 sources and 3 targets listed in Table S3.

| Algorithm | Path weight | Max. edge weight | Avg. edge weight | Min. edge weight | Max. edge use | Avg. edge use | Min. edge use | Max. degree | Avg. degree | Min. degree |
|---|---|---|---|---|---|---|---|---|---|---|
| Random + search | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MIN-SAT | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MIN-SAT + search | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MAX-CSP | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MAX-CSP + search | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| MTO | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Unoriented edge selection | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Oriented baseline | 2.0 | 2.3 | 2.0 | 1.8 | 0.5 | 0 | 0.1 | 0.9 | 0 | 0.9 |

**Table S9.** Top-ranked predicted pathways for which 4 of the 6 vertices in the path are present consecutively in a gold standard pathway. Pathways were inferred using the first 7 sources and 7 targets listed in Table S3.

| Algorithm | Path weight | Max. edge weight | Avg. edge weight | Min. edge weight | Max. edge use | Avg. edge use | Min. edge use | Max. degree | Avg. degree | Min. degree |
|---|---|---|---|---|---|---|---|---|---|---|
| Random + search | 8 | 6 | 9 | 4 | 0 | 0 | 0 | 3 | 0 | 1 |
| MIN-SAT | 6 | 0 | 6 | 4 | 0 | 0 | 8 | 4 | 0 | 0 |
| MIN-SAT + search | 8 | 10 | 9 | 4 | 0 | 0 | 0 | 3 | 0 | 1 |
| MAX-CSP | 7 | 6 | 8 | 4 | 0 | 0 | 0 | 2 | 0 | 0 |
| MAX-CSP + search | 7 | 6 | 8 | 4 | 0 | 0 | 0 | 2 | 0 | 0 |
| MTO | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.9 | 1.0 | 0.9 | 0.9 | 1.0 |
| Unoriented edge selection | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Oriented baseline | 3.2 | 2.6 | 3.3 | 2.7 | 0.7 | 0 | 0.9 | 1.7 | 0 | 0.6 |

While some of the major trends observed in Table 2 are still present when using fewer sources and targets, there are several notable differences. First, all algorithms are able to match a greater number of gold standard pathways when there are more sources and targets. This is as expected because it is less likely that a gold standard pathway originating at a protein that is not in the set of sources given to the algorithms will be recovered than when that protein is designated as a source. In addition, when there are a small number of sources and targets, the algorithms recover essentially optimal solutions with respect to the objective function and do not benefit from local search. Indeed, all three of our algorithms perform identically when there are 3 sources and 3 targets.

The unoriented edge selection algorithm performs especially poorly when there are fewer sources and targets. With 3 sources and 3 targets, is fails to select edges that create even a single source-target path containing 6 or fewer vertices even though there are still 1934 edges in the subnetwork it selects. For 7 sources and 7 targets, it only finds 3 short source-target paths, one of which contains 6 vertices and is included in the evaluation above, although the chosen subnetwork contains 1941 edges. These instances once again reveal the importance of the preference for short, directed paths inherent in our formulation of the signaling pathway prediction problem.

For Tables S8 and S9 the algorithms were run in the same manner that they were for the evaluation summarized in Table 2 of the main text (see Materials and Methods). However, the MIN-SAT and MIN-SAT with local search results here are based on 5 executions of the MIN-SAT solver instead of 20.

### MTO and undirected edge selection local search

As seen in Figure 3 and Table 2, local search is very successful for the MEO problem. However, Table S10 shows local search does not aid MTO and unoriented edge selection in the same manner.

**Table S10.** Local search results for MTO and unoriented edge selection. The random orientation with search, MTO without search, and unoriented edge selection without search rows are copied from Table 2 to facilitate comparison

| Algorithm | Path weight | Max. edge weight | Avg. edge weight | Min. edge weight | Max. edge use | Avg. edge use | Min. edge use | Max. degree | Avg. degree | Min. degree |
|---|---|---|---|---|---|---|---|---|---|---|
| Random + search | 37 | 11 | 36 | 34 | 0 | 0 | 40 | 10 | 0 | 0 |
| MTO | 3.2 | 3.2 | 3.2 | 3.2 | 3.0 | 3.0 | 3.0 | 3.0 | 2.8 | 3.2 |
| MTO + search | 2.4 | 2.4 | 2.4 | 2.3 | 2.1 | 2.0 | 2.3 | 2.0 | 1.9 | 2.2 |
| Unoriented edge selection | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Unoriented edge selection + search | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Because MTO is a randomized algorithm, the results above are averaged over 20 runs. Local search did increase the number of reachable source-target pairs in some of the 20 runs, and after the search the optimal number of reachable pairs in the contracted graph was obtained in all runs. We observed that very few edge flips in the contracted graph were required to achieve this objective function value. On average 0.9 edge flips were performed, and there were never more than 2 flips during the search. However, increasing the number of reachable pairs in the contracted graph does not cause there to be more length-bounded pathways that match known signaling pathways once the graph is expanded. In fact, there are slightly fewer matches on average after local search, which could be an artifact of the variance in the MTO results. Because our goal is to determine which set of assumptions and problem formulation is more likely to identify known pathways, we did not consider running MTO and using the resulting orientations as a starting point for MEO local search.

As indicated in Table S10, local search does not affect the unoriented edge selection results for the instances we tested. The linear program solver produces very good solutions with respect to the unoriented edge selection objective function such that all edges that could be added during local search have weight less than the penalty parameter λ and all edges that could be removed have weight greater than λ. Therefore, any addition or removal would increase the objective function of this minimization problem.

## Detailed discussion of partial match pathways

The top 20 highest ranked partial matches were analyzed in detail to validate if the predicted pair-wise relationships that were not in the gold standard dataset were correct. The predicted paths are within the yeast MAPK overall signaling systems, which contain four analogous signaling pathways that have distinct signaling input: pheromone, hypotonic shock, hypertonic shock, and starvation (also referred to as filamentation pathway). All pathways pass through the conserved module of MAPKKK, MAPKK, and MAPK, but the actual proteins that take these roles can be different. However, there is significant overlap. For example, the MAPKKK Ste11 is used in pheromone, hypertonic, and starvation pathways and the MAPKK Ste7 in pheromone and starvation pathways. Much recent effort is directed at understanding how signaling crosstalk is regulated, addressing questions such as why Ste7 activated by pheromones phosphorylates the MAPK Fus3, while Ste7 activated by starvation phosphorylates the MAPK Kss1. One difference between the pheromone and starvation pathways is the involvement of the Ste5 in pheromone but not starvation signaling. Ste5 has been thought of until recently as a scaffold protein, that brings together the different components required for signaling. Recently, it was shown that the mechanism also involves other roles of Ste5 than being a scaffold alone (24). Binding of a novel domain within Ste5 to Ste7 makes Fus3 a 5000-fold better target for phosphorylation by Ste7, which is a poor substrate in the absence of Ste5. In contrast, Kss1 is a good substrate irrespective of the binding of this novel domain, explaining the specificity in Fus3 versus Kss1 phosphorylation in pheromone and starvation pathways, respectively.

The interaction Ste11→Ste5 was a member in many of the top-ranked pathways but was only present in the gold standard pheromone signaling pathway in the opposite direction, Ste5→Ste11. In the main text we demonstrated the validity of the Ste11→Ste5 orientation, which verifies the many partially redundant paths that contain this edge. For example, the top two predicted paths differ only by the final edge, Fus3→Dig2 versus Fus3→Dig1. Because Dig2 and Dig1 are functionally redundant inhibitors of Ste12 (25), these are both equally valid.

Due to the strict requirement that all directed edges must be present consecutively in a *single* gold standard pathway in order to be considered a complete match, some of our partial match pathways actually agree with the gold standard on all individual edge orientations. For instance, one top ranked path Sho1→Ste11→Ste7→Fus3→Dig1→Ste12 predicts the correct orientation for each edge. However, it is not labeled a complete match because the Sho1→Ste11 edge is a member of the gold standard HOG pathway whereas the other four edges are found consecutively in the gold standard pheromone signaling pathway.

One of the predicted orientations, Ste11→Fus3, is not likely to be correct because it is Ste7-Ste5 that connects to Fus3, while Ste11 is upstream of Ste7. Furthermore, the binary interaction between Ste11 and Fus3 is in the micromolar range and thus relatively weak (26). However, because all of these proteins are part of the Ste5-mediated large complex, many types of experimental evidence support the interaction and the edge was assigned a high confidence score, providing a potential reason as to why the model made this prediction. This result highlights a general difficulty with protein-protein interaction data, which often suffer from being not direct physical interactions but being mediated by indirect interactions in a complex and the lack of quantitative data. In only very few cases have the affinities been measured. The field of quantitative proteomics is directly aimed at providing more quantitative information and in future such information may become available in larger numbers so that it can be included in model building.

### Detailed discussion of no match pathways

As discussed in the main text, although many of the top 20 ranked pathways discovered by random orientation followed by local search did not contain any of the interactions in the gold standard signaling network, 9 of them are known cell cycle paths.  Because evaluating each pathway that is not in the gold standard requires considerable manual effort and literature search, it is impractical to examine all of the pathways that do not match the gold standard.  However, for the cell cycle paths in Figure 4C we found strong evidence for most of the predicted orientations, as summarized in Table S11.

**Table S11.** 10 cell cycle interactions that are present in the pathways in Figure 4C and whose predicted direction is supported by the literature.

| Directed edge | Literature support |
|---|---|
| Cdc28→Far1 | (27) |
| Cdc28→Swi4 | (28) |
| Cdc28→Swi6 | (29) |
| Cks1→Cdc28 | (30) |
| Cks1→Clb2 | (31) |
| Clb2→Cdc28 | (32) |
| Clb3→Cdc28 | (33) |
| Cln2→Cks1 | (34) |
| Cln2→Ste20 | (35) |
| Swi6→Swi4 | (36) |

During the literature search, we also found evidence that "Far1 is a substrate for Fus3" (37) contrary to our Far1→Fus3 orientation. We believe the error is a consequence of choosing Fus3 as a target. Because Fus3 is a target, many source-target paths in the initial undirected graph are likely to use it in the direction we predicted relative to the number of paths that use it in the true direction. If we did not orient the edge from Far1 to Fus3, all of these paths would be violated, decreasing the objective function. This error shows the importance in careful source and target selection and the effect annotating a protein as a source or target can have on the orientations of its edges.

We also inspected the 11 other top-ranked paths in detail to assess to what degree these paths may be real. We found that overall the proteins connected in the paths are functionally related and parts of the predicted paths are likely to be biologically relevant. In particular, there are generally sets of related paths predicted where out of that group of paths, one or more can be selected that seem biologically most feasible. Thus, the predictions generate a set of hypotheses, and variations/alternative hypotheses, that provide an experimentalist with specific experiments to be carried out. For example, among the 11 paths, there are 6 paths that begin with Sin3, a component of the histone deacetylase complex. Subsequent proteins are related to chromatin remodeling, nuclear import, and transcription regulation, all related processes. The predicted edges are feasible and experimentally verifiable. However, some of the predicted directed edges are unlikely to be true, even when there are several feasible edges before or after the edge in question. For 9 out of the 11 paths, there are 1 or 2 interactions per path (13 in total) that are clearly wrong. 6 of these 13 interactions are the same so verifying this single interaction would affect more than half of the predicted paths. Cleaning such paths of these “contaminants” is a future goal that is highly feasible. Moreover, the corresponding pairs are relatively few in number but appear in many paths so the impact would be very high. All top-ranked paths analyzed here can be downloaded from our supporting website http://www.sb.cs.cmu.edu/OrientEdges.

# References

1. Halperin,E. and Zwick,U. (2001) Combinatorial approximation algorithms for the maximum directed cut problem. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Washington, D.C., United States, pp. 1-7.

2. Håstad,J. (2001) Some optimal inapproximability results. *JACM*, **48**, 798-859, 10.1145/502090.502098.

3. Bertsimas,D., Teo,C. and Vohra,R. (1999) On dependent randomized rounding algorithms. *Operations Research Letters*, **24**, 105-114, 10.1016/S0167-6377(99)00010-3.

4. Trevisan,L. (1998) Parallel Approximation Algorithms by Positive Linear Programming. *Algorithmica*, **21**, 72-88, 10.1007/PL00009209.

5. Raghavendra,P. and Steurer,D. (2009) How to Round Any CSP. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. Atlanta, GA.

6. Charikar,M., Makarychev,K. and Makarychev,Y. (2009) Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Trans. Algorithms*, **5**, 1-14, 10.1145/1541885.1541893.

7. Lewin,M., Livnat,D. and Zwick,U. (2002) Improved Rounding Techniques for the MAX 2-SAT and MAX DI-CUT Problems. In *Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science. Vol. 2337, pp. 67-82.

8. Zwick,U. (1998) Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, San Francisco, California, United States, pp. 201-210.

9. Guruswami,V., Lewin,D., Sudan,M. and Trevisan,L. (1998) A Tight Characterization of NP with 3 Query PCPs. In *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, p. 8.

10. Sanchez,M., Bouveret,S., Givry,S.D., Heras,F., J´egou,P., Larrosa,J., Ndiaye,S., Rollon,E., Schiex,T., Terrioux,C. et al. (2008) Max-CSP competition 2008: toulbar2 solver description. In Dongen,M.V., Lecoutre,C., Roussel,O. (eds), *Proceedings of the Third International CSP Solver Competition*.pp. 63-70.

11. Stark,C., Breitkreutz,B., Reguly,T., Boucher,L., Breitkreutz,A. and Tyers,M. (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, D535-539, 10.1093/nar/gkj109.

12. Deng,M., Sun,F. and Chen,T. (2003) Assessment of the reliability of protein-protein interactions and protein function prediction. In *Proceedings of the 8th Pacific Symposium on Biocomputing*. Kauai, Hawaii, pp. 140-151.

13. Bader,J.S., Chaudhuri,A., Rothberg,J.M. and Chant,J. (2004) Gaining confidence in high-throughput protein interaction networks. *Nat. Biotech.*, **22**, 78-85, 10.1038/nbt924.

14. Sharan,R., Suthram,S., Kelley,R.M., Kuhn,T., McCuine,S., Uetz,P., Sittler,T., Karp,R.M. and Ideker,T. (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl. Acad. Sci. U.S.A.*, **102**, 1974-1979, 10.1073/pnas.0409522102.

15. Kanehisa,M. and Goto,S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27-30.

16. Thorner,J., Westfall,P.J. and Ballon,D.R. High Osmolarity Glycerol (HOG) Pathway in Yeast. In *Science Signaling*, Connections Map in the Database of Cell Signaling. Available at: http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14620.

17. Thorner,J., Truckses,D.M. and Garrenton,L.S. Filamentous Growth Pathway in Yeast. In *Science Signaling*, Connections Map in the Database of Cell Signaling. Available at: http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14554.

18. Dohlman,H. and Slessareva,J.E. Pheromone Signaling Pathways in Yeast. In *Science Signaling*, Connections Map in the Database of Cell Signaling. Available at: http://stke.sciencemag.org/cgi/cm/stkecm;CMP_13999.

19. Medvedovsky,A., Bafna,V., Zwick,U. and Sharan,R. (2008) An Algorithm for Orienting Graphs Based on Cause-Effect Pairs and Its Applications to Orienting Protein Networks. In *Proceedings of the 8th international workshop on Algorithms in Bioinformatics*. Karlsruhe, Germany, pp. 222-232.

20. Zhao,X., Wang,R., Chen,L. and Aihara,K. (2008) Uncovering signal transduction networks from high-throughput data by integer linear programming. *Nucleic Acids Res.*, **36**, e48, 10.1093/nar/gkn145.

21. Yeang,C., Ideker,T. and Jaakkola,T. (2004) Physical Network Models. *J. Comput. Biol.*, **11**, 243-262, 10.1089/1066527041410382.

22. Aranda,B., Achuthan,P., Alam-Faruque,Y., Armean,I., Bridge,A., Derow,C., Feuermann,M., Ghanbarian,A.T., Kerrien,S., Khadake,J. et al. (2009) The IntAct molecular interaction database in 2010. *Nucleic Acids Res.*, gkp878, 10.1093/nar/gkp878.

23. Chatr-aryamontri,A., Ceol,A., Palazzi,L.M., Nardelli,G., Schneider,M.V., Castagnoli,L. and Cesareni,G. (2007) MINT: the Molecular INTeraction database. *Nucleic Acids Res.*, **35**, D572-574, 10.1093/nar/gkl950.

24. Good,M., Tang,G., Singleton,J., Reményi,A. and Lim,W.A. (2009) The Ste5 Scaffold Directs Mating Signaling by Catalytically Unlocking the Fus3 MAP Kinase for Activation. *Cell*, **136**, 1085-1097, 10.1016/j.cell.2009.01.049.

25. Tedford,K., Kim,S., Sa,D., Stevens,K. and Tyers,M. (1997) Regulation of the mating pheromone and invasive growth responses in yeast by two MAP kinase substrates. *Curr. Biol.*, **7**, 228-238, 10.1016/S0960-9822(06)00118-7.

26. Maeder,C.I., Hink,M.A., Kinkhabwala,A., Mayr,R., Bastiaens,P.I.H. and Knop,M. (2007) Spatial regulation of Fus3 MAP kinase activity through a reaction-diffusion mechanism in yeast pheromone signalling. *Nat. Cell Biol.*, **9**, 1319-1326, 10.1038/ncb1652.

27. Blondel,M., Galan,J.M., Chi,Y., Lafourcade,C., Longaretti,C., Deshaies,R.J. and Peter,M. (2000) Nuclear-specific degradation of Far1 is controlled by the localization of the F-box protein Cdc4. *EMBO J.*, **19**, 6085-6097, 10.1093/emboj/19.22.6085.

28. Amon,A., Tyers,M., Futcher,B. and Nasmyth,K. (1993) Mechanisms that help the yeast cell cycle clock tick: G2 cyclins transcriptionally activate G2 cyclins and repress G1 cyclins. *Cell*, **74**, 993-1007, 10.1016/0092-8674(93)90722-3.

29. Geymonat,M., Spanos,A., Wells,G.P., Smerdon,S.J. and Sedgwick,S.G. (2004) Clb6/Cdc28 and Cdc14 Regulate Phosphorylation Status and Cellular Localization of Swi6. *Mol. Cell. Biol.*, **24**, 2277-2285, 10.1128/MCB.24.6.2277-2285.2004.

30. Tang,Y. and Reed,S.I. (1993) The Cdk-associated protein Cks1 functions both in G1 and G2 in Saccharomyces cerevisiae. *Genes Dev.*, **7**, 822-832.

31. Kaiser,P., Moncollin,V., Clarke,D.J., Watson,M.H., Bertolaet,B.L., Reed,S.I. and Bailly,E. (1999) Cyclin-dependent kinase and Cks/Suc1 interact with the proteasome in yeast to control proteolysis of M-phase targets. *Genes Dev.*, **13**, 1190-1202.

32. Eluere,R., Offner,N., Varlet,I., Motteux,O., Signon,L., Picard,A., Bailly,E. and Simon,M. (2007) Compartmentalization of the functions and regulation of the mitotic cyclin Clb2 in S. cerevisiae. *J. Cell. Sci.*, **120**, 702-711, 10.1242/jcs.03380.

33. Hu,F., Gan,Y. and Aparicio,O.M. (2008) Identification of Clb2 Residues Required for Swe1 Regulation of Clb2-Cdc28 in Saccharomyces cerevisiae. *Genetics*, **179**, 863-874, 10.1534/genetics.108.086611.

34. Ptacek,J., Devgan,G., Michaud,G., Zhu,H., Zhu,X., Fasolo,J., Guo,H., Jona,G., Breitkreutz,A., Sopko,R. et al. (2005) Global analysis of protein phosphorylation in yeast. *Nature*, **438**, 679-684, 10.1038/nature04187.

35. Wu,C., Leeuw,T., Leberer,E., Thomas,D.Y. and Whiteway,M. (1998) Cell cycle- and Cln2p-

Cdc28p-dependent phosphorylation of the yeast Ste20p protein kinase. *J. Biol. Chem.*, **273**, 28107-28115.

36. Foster,R., Mikesell,G.E. and Breeden,L. (1993) Multiple SWI6-dependent cis-acting elements control SWI4 transcription through the cell cycle. *Mol. Cell. Biol*, **13**, 3792-3801.

37. Peter,M., Gartner,A., Horecka,J., Ammerer,G. and Herskowitz,I. (1993) FAR1 links the signal transduction pathway to the cell cycle machinery in yeast. *Cell*, **73**, 747-760, 10.1016/0092-8674(93)90254-N.