
INDRA Network Service

K. Karis

Oct 05, 2021

CONTENTS:

1	Network Search Web UI	3
1.1	Source and Target	3
1.1.1	Autocompleting source/target inputs	3
1.2	Detailed Search Options - General Options	4
1.2.1	Path Length	4
1.2.2	Node Blacklist	5
1.2.3	Max Paths	5
1.2.4	Signed Search	5
1.2.5	Highest Degree Node Culling Frequency	6
1.2.6	Belief Cutoff	6
1.2.7	Allowed Statement Types	6
1.2.8	Allowed Node Namespaces	6
1.2.9	Checkboxes	6
1.3	Detailed Search Options - Context and Weighted Search Options	7
1.3.1	Unweighted	7
1.3.2	Belief Weighted	7
1.3.3	DepMap z-score weighted	8
1.3.4	Mesh Context	8
1.4	Detailed Search Options - Open Search Options	8
1.4.1	Terminal Namespaces	8
1.4.2	Max Children Per Node (Unweighted Search)	9
1.4.3	Depth Limit (Unweighted Search)	9
1.5	Timeout	9
1.6	Result Categories	10
1.6.1	Common Parents	10
1.6.2	Shared Targets	10
1.6.3	Shared Regulators	10
1.6.4	Path Results	10
1.7	Detailed Results	11
1.8	The Graphs Used	11
2	INDRA Network Search Modules Reference	13
2.1	Autocomplete (indra_network_search.autocomplete.autocomplete)	13
2.2	Data Models (indra_network_search.data_models)	13
2.3	Rest Models (indra_network_search.data_models.rest_models)	13
2.4	Pathfinding (indra_network_search.pathfinding.pathfinding)	13
2.5	Utility Functions (indra_network_search.util)	13
2.5.1	Util (indra_network_search.util.curation_cache)	13
2.6	Query Classes (indra_network_search.query)	13
2.7	Query Handler (indra_network_search.query_handler)	13

2.8	Rest API (<code>indra_network_search.rest_api</code>)	13
2.9	Rest API Utilities (<code>indra_network_search.rest_util</code>)	14
2.10	Result Handlers (<code>indra_network_search.result_handler</code>)	14
2.11	Search API (<code>indra_network_search.search_api</code>)	14
3	Indices and tables	15

This documentation covers a tutorial and the modules of the INDRA Network Search and is part of a broader set of INDRA derived applications. To read more about INDRA, see the [homepage](#), the [documentation](#) and the project on [github](#).

NETWORK SEARCH WEB UI

This document introduces the web interface of the INDRA Network Search Service

The INDRA Network Search
Search across 96617 nodes, 5303703 edges.
Last updated: 2021-08-09. Current status: Available
Read the [API Docs](#) and read the [General Docs](#)
[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'tp1xcmek'

Target node, e.g. 'ACE2' or 'hgnc:13557'

Detailed Search Options

General Options▼

Context and Weighted Search Options▼

Open Search Options▼

Submit

Timeout
30

Share

No results

About

[INDRALAB](#)
[Harvard Program in Therapeutic Science \(HTS\)](#)
Automated Scientific Discovery Framework (ASDF)
The DARPA ASDF project develops algorithms and software for reasoning about complex mechanisms operating in the natural world, explaining large-scale data, assisting humans in generating actionable, model-based hypotheses and testing these hypotheses empirically.
ASDF is funded by the Defense Advanced Research Projects Agency under award W911NF018-1-0124.[Back to top](#)

Fig. 1: *The network search interface with no input or results.*

1.1 Source and Target

The source and target are the nodes between which to find a path and at least one of source and target is needed to do a search. If only one of source or target is provided, an open ended breadth first search is done instead of a path search. Note that the source and target are not affected by the choice of *allowed namespaces* (see below at [Allowed Node Namespaces](#)).

1.1.1 Autocompleting source/target inputs

Autocompletion of source/target based on prefix and entity identifier are made automatically as an input is typed or pasted into the text boxes. The suggestions are picked from the nodes in the graph and the text box will mark the entered text as correct if it matches an existing node in the graph.

The INDRA Network Search

Search across 96617 nodes, 5303703 edges.
Last updated: 2021-08-09. Current status: available

Read the [API Docs](#) and read the [General Docs](#)

[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'tpixmek'

Target node, e.g. 'ACE2' or 'hgnc:13557'

ace2

Detailed Search Options

General Options	ACE2 HGNC:13557	▼
Context and Weighted Search Options	Ace2 LIP:QBR010	▼
Open Search Options		▼

Fig. 2: Autocompleting an entity. As an entity name is typed into the source or target text boxes, suggestions from the graph nodes are provided.

The INDRA Network Search

Search across 96617 nodes, 5303703 edges.
Last updated: 2021-08-09. Current status: available

Read the [API Docs](#) and read the [General Docs](#)

[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'tpixmek'

Target node, e.g. 'ACE2' or 'hgnc:13557'

ACE2 ✓

Detailed Search Options

General Options		▼
Context and Weighted Search Options		▼
Open Search Options		▼

Fig. 3: Verified entry. Entries are verified as they are being typed or pasted into the text box. When the entered entity is verified to exist in the graph, the text box border will switch to green and a checkmark will appear.

1.2 Detailed Search Options - General Options

The general detailed search options contain filters that apply to most searches, regardless of weighting or openness.

1.2.1 Path Length

Only paths of this many edges will be returned. Must be a positive integer.

The INDRA Network Search

Search across 96617 nodes, 5303703 edges.
 Last updated: 2021-08-09. Current status: available

Read the [API Docs](#) and read the [General Docs](#)

[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'Tp1cmeK'

Target node, e.g. 'ACE2' or 'hgnc:13557'

Detailed Search Options

General Options ^

<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Path length</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Max Paths 50</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Highest Degree Node Culling Frequency</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Allowed Statement Types v</div> <div style="display: flex; justify-content: space-between;"> <input type="checkbox"/> Only Database Supported Sources <input type="checkbox"/> Allow ontological edges <input type="checkbox"/> Include Reverse Search <input type="checkbox"/> Include Search for shared regulators of source/target </div>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Node Blacklist</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Signed Search No sign v</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Belief Cutoff 0</div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">Allowed Node Namespaces v</div>
--	--

Context and Weighted Search Options v

Open Search Options v

Fig. 4: The search interface with the general options section expanded.

1.2.2 Node Blacklist

Node names entered here are skipped in the path search. This is a good way to avoid nodes of extremely high degree that overwhelms the results and effectively blocks out results that include nodes of lower degree. *See also [Highest Degree Node Culling Frequency](#) below.*

1.2.3 Max Paths

The maximum number of results to return per category in the results. The default and the maximum allowed is 50 results. For unweighted searches this number rarely makes a perceivable difference in response time but for weighted searches keep this number low for a faster response time.

1.2.4 Signed Search

To perform a signed search, click on the drop down menu that says “No sign” and chose a sign. “+” means that the returned paths are upregulations, and “-” means that the returned paths are downregulations. For the purpose of signed search, only statements that imply a clear up- or downregulation are considered. Currently this mean *IncreaseAmount* and *Activation* for upregulation, and *DecreaseAmount* and *Inhibition* for downregulation.

1.2.5 Highest Degree Node Culling Frequency

Entering a positive integer here allows the path search to include the highest degree node for the first N returned paths, after which that node is added to the *Node Blacklist*. This is repeated for the second highest degree node for the following N paths, then for the third highest degree node and so forth. *Limitations:* This option is only applied to unweighted open search and source-target searches.

1.2.6 Belief Cutoff

Any statement with a belief score below this number will be excluded from the edge support. If all statements are excluded from the edge, all paths containing that edge become invalid and are skipped. It is set to zero by default to include all edges. Read more about belief scores in the [belief module](#) of INDRA.

1.2.7 Allowed Statement Types

This is a multiselect dropdown which contains multiple statement types to allow in the results. If an edge of a path does not contain any of the selected statement types, the whole path will be skipped from the result. Read more about statement types in the [statements module](#) of INDRA.

1.2.8 Allowed Node Namespaces

The namespaces included here are the ones that are allowed on any node visited in the path search. The namespace of the source and target are excluded from this restriction. A namespace in INDRA is the prefix or name of the *type* of identifier used to uniquely identify an entity from a specific knowledge source. For example, a chemical can be identified using a *CHEBI* identifier and would then be identified in the *CHEBI* namespace.

1.2.9 Checkboxes

The following options are available as checkboxes:

- **Only Database Supported Sources:** Check this box to enforce that all edges must be supported by at least one statement sourced from curated databases like PathwayCommons and Signor
- **Allow Ontological Edges:** Check this box to allow directed edges that go from an entity to its ontological parent, e.g. from the NFKB1 sub-unit to the NFkappaB complex.
- **Include Reverse Search:** Check this box to also search for paths with source and target swapped. With this option, the reverse search *from* target *to* source is done as well as the original search from source to target. If the *timeout* is reached (see below) before the reverse search can start, the reverse search will not return any paths. If the *timeout* is reached during the reverse search, fewer paths than for the original search will be returned.
- **Include Search for Shared Regulators of Source/Target:** Check this box to include a search for common upstream nodes one edge away from both source and target. This option is only available when both source and target specified.

1.3 Detailed Search Options - Context and Weighted Search Options

This section of the search options allows control over how to prioritize or *weight* edges in paths differently. During weighted search, the cost along every path encountered is calculated as the sum of the edge weights along the path. The paths are returned in ascending order of cost.

The different ways of weighting the search are available in the dropdown menu “Weighted Search”. *Note:* A weighted search is costly and usually takes longer than an unweighted search. It is common that a very heavy weighted search times out, especially for a *signed weighted* search, even with the highest allowed *timeout* of 120 seconds.

The weighted search uses a slightly modified version of the Dijkstra weighted search employed in Networkx.

The code implemented for the weighted search is available on [github](#) in the functions *shortest_simple_paths()* and *open_dijkstra_search()* for closed and open paths, respectively.

The INDRA Network Search

Search across 96617 nodes, 5303703 edges.
Last updated: 2021-08-09. Current status: available

Read the [API Docs](#) and read the [General Docs](#)

[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'fplx:mek'

Target node, e.g. 'ACE2' or 'hgnc:13557'

Detailed Search Options

General Options ▼

Context and Weighted Search Options ▲

Weighted Search
Unweighted ▼

Belief weighted
DepMap z-score weighted
Mesh Context
Unweighted

Mesh IDs (comma separated)

☐ Strict Mesh ID filtering without weights

Constant C
1

Constant Tk
10

Open Search Options ▼

Fig. 5: The search interface with the Context and Weighted search options section expanded.

1.3.1 Unweighted

This is the default option and imposes no weight on the edges and is equivalent to all edges having a unit weight.

1.3.2 Belief Weighted

The belief weight of an edge is calculated as the negative log of the aggregated belief scores of all the statements supporting edge e :

$$w_e = -\log \left(1 - \prod_i (1 - b_i) \right)$$

where b_i is the belief score of supporting statement i of edge e . Since the belief score is limited to the interval $[0, 1]$, it can interpreted as a probability and the above weight can therefore be seen as the log of the *complement* to the probability that every supporting statement is *incorrect*.

1.3.3 DepMap z-score weighted

The z-score edge weight is focused around prioritizing edges between human genes that have been targeted in knockout screens performed at the Broad Institute's Dependency Map project. The z-score is obtained from first calculating the pearson correlation between all pairs of genes in the gene knockout screen. Then the log of the p-values of the correlations are calculated using the CDF of the beta distribution. Finally the strength of the z-scores are obtained from the p-values and the signs are recuperated from the original correlation matrix.

The edge weight, assuming both nodes are human genes, is calculated by normalizing the difference between the z-score associated with a self-correlation and the strength of the z-score between the two nodes of the edge. In the case that one or both of the nodes of the edge are non-gene entities, the z-score weight is set to 1:

$$w_e = \begin{cases} 1 & \text{if } z_e = z_0 \\ \frac{z_0 - |z_e|}{z_0} & \text{if } z_e \neq z_0 \end{cases}$$

where z_0 is the z-score associated with self correlation and z_e is the z-score of the edge.

1.3.4 Mesh Context

The context based search allows a search to prioritize or only include connections that are relevant to the provided context. The context is given as MeSH terms.

- **MeSH IDs:** Enter the MeSH IDs, separated by comma, that should be used in the search.
- **Strict Filtering on MeSH IDs:** Check this box to *only* allow edges with associated with the provided MeSH IDs. If left unchecked, the search is weighted.
- **Constants C and T_k :** These two constant allow for changing the importance of the context in a weighted context based search. For any edge e , the weight w_e of the edge in the context based search is calculated in the following way:

$$w_e = -C \cdot \log \left(\frac{\text{refcount}}{\text{total} + T_k} \right)$$

Here, *refcount* is the number of references with the associated MeSH ID(s) that are supporting edge e and *total* is the total number of references supporting edge e .

1.4 Detailed Search Options - Open Search Options

Options under the Open Search Options are only applied during open searches, i.e. when either of source or target is provided.

1.4.1 Terminal Namespaces

Namespaces selected here restrict the search to only return paths that *end* (open search from source) or *start* (open search from target) on the given namespaces and then not consider these nodes further. For example: if namespace A is selected, then a downstream path might look like this: X->Y->A, but not like this: X->Y->A->Z, where X, Y, Z are all namespaces other than A.

The INDRA Network Search

Search across 96617 nodes, 5303703 edges.
 Last updated: 2021-08-09. Current status: available

Read the [API Docs](#) and read the [General Docs](#)

[Search](#) | [About](#)

Basic Search Options

Source node, e.g. 'MEK' or 'tpbcmek'

Target node, e.g. 'ACE2' or 'hgnc:13557'

Detailed Search Options

General Options ▼

Context and Weighted Search Options ▼

Open Search Options ▲

Terminal Namespaces ▼

Max children per node
5

Depth limit in unweighted search
2

Submit

Timeout
30

Share

Fig. 6: The search interface with the Open search options section expanded.

1.4.2 Max Children Per Node (Unweighted Search)

The integer provided here gives a maximum for the number of children to continue to open search from. For example: if N is set here, the first N nodes selected from the starting node are then considered for the next layer in the breadth first search. This option is only available for *unweighted* searches.

1.4.3 Depth Limit (Unweighted Search)

This option limits how deep, i.e. how many edges, the returned paths are allowed to be/have. This option is only available for *unweighted* searches.

1.5 Timeout

Setting a timeout allows to set a larger (or smaller) timeout than the default 30 seconds timeout. The time since the path search was started is checked before each iteration of data assembly for a returned path during the search. If the time passed is larger than the allowed timeout, the search is stopped. The timeout provided has to be a decimal number smaller than or equal to 120 seconds.

1.6 Result Categories

Note: If there are no results for a specific category, that section will be hidden.

1.6.1 Common Parents

This section shows the result of a search for common ontological parents of source and target. For example with *GP1BA* and *GP1BB* as source and target, respectively, the Platelet membrane glycoprotein complex shows up as a shared ontological parent.

The screenshot shows the 'Basic Search Options' section with two input fields: 'Source node, e.g. "MEK" or "tp53mek"' containing 'GP1BA' and 'Target node, e.g. "ACE2" or "hgnc:13557"' containing 'GP1BB'. Below these are 'Detailed Search Options' including 'General Options', 'Context and Weighted Search Options', and 'Open Search Options'. A 'Submit' button is present, along with a 'Timeout' field set to '30' and a 'Share' button. The 'Results' section shows a table of 'Common Parents' with one entry: 'GP1b_IX_V' in the 'Name' column, 'FPLX' in the 'Namespace' column, 'GP1b_IX_V' in the 'Identifier' column, and a 'Lookup' icon in the 'Lookup' column.

Name	Namespace	Identifier	Lookup
GP1b_IX_V	FPLX	GP1b_IX_V	

Fig. 7: The result of a search with *GP1BA* and *GP1BB* as source and target, respectively, for Common Parents showing the Platelet membrane glycoprotein complex as their shared protein complex.

1.6.2 Shared Targets

This section shows the direct downstream targets that are shared between *source* and *target*.

1.6.3 Shared Regulators

Shared regulators are only searched for if the corresponding checkbox is checked (see [checkboxes](#) above). The results shown are the direct upstream regulators that are shared between *source* and *target*.

1.6.4 Path Results

This section show path results per path length, i.e. all paths with the same number of edges share a specific subsection. The division of paths per subsection is done regardless if the path search is weighted or not.

Source node, e.g. "MEK" or "tpcx.mek" ✓

Target node, e.g. "ACE2" or "hgnc:13557" ✓

SIK3JUN

Detailed Search Options

General Options

Context and Weighted Search Options

Open Search Options

Submit

Timeout 30

Share

Results

Click on ⊕/⊖ to expand/collapse results

Path Results

2-edge paths; SIK3 → X1 → JUN 36

3-edge paths; SIK3 → X1 → X2 → JUN 14

Shared Targets

Shared Regulators

Shared regulator	Source edge:	Target edge:	Support
PKA	⊖	⊖	PKA → SIK3 PKA → JUN
AKT	⊖	⊖	AKT → SIK3 AKT → JUN

Fig. 8: Search results with SIK3 as source and JUN as target with the Shared Regulator section expanded.

1.7 Detailed Results

For each result section, excluding Common Parents, there are two levels of detail. Results for Common Parents only have one level of results: name, namespace, identifier and entity lookup. The first level shows path (for Path Results), target (for Shared Targets) or regulator (for Shared Regulators) together with weight (if the search is weighted) the edge, source counts and a link to the INDRA DB for that specific edge.

The second level of results is collapsed by default. To expand it, the circled “+” (⊕) need to be clicked. Once expanded, source counts and a link to more specific information in the INDRA DB per statement type are shown.

As the network search results can be filtered in more detail than what is possible using the INDRA DB, the statements shown in the DB can sometimes be a superset of the statements shown in the second level of the results.

1.8 The Graphs Used

The two graphs used for the network search are assembled from a full snapshot of the [INDRA DataBase](#) that is updated regularly. Any statement that includes two or three agents are assembled into the support for the edges for the graphs, with one edge containing one or more statements. The two types of graphs used are:

1. Unsigned directed graph
2. Signed node directed graph

The edges in the signed graph only contain statements that have clear up- or downregulations associated with them, which currently are *IncreaseAmount* and *Activation* for upregulation, and *DecreaseAmount* and *Inhibition* for downregulation.

The code assembling the graphs can be found in `net_functions.py` in the function `sif_dump_df_to_digraph()` in the `depmap_analysis` repository.

Path Results					
2-edge paths; SIK3→ X1→JUN 36					
Path	Support				
SIK3→MTOR→JUN	Weight	Edge	Sources	DB Link	
	⊕ N/A	SIK3→MTOR	1 2 6	ⓘ	
	⊖ N/A	MTOR→JUN	1 2 2	ⓘ	
SIK3→calcium(2+)→JUN	Weight	Edge	Sources	DB Link	
	⊕ N/A	SIK3→calcium(2+)	1 1	ⓘ	
	⊖ N/A	calcium(2+)→JUN	1 3 24	ⓘ	
SIK3→PI3K→JUN	Weight	Edge	Sources	DB Link	
	⊕ N/A	SIK3→PI3K	1 1	ⓘ	
	⊖ N/A	PI3K→JUN	1 1 3 11 29	ⓘ	
SIK3→TLR4→JUN	Weight	Edge	Sources	DB Link	
	⊕ N/A	SIK3→TLR4	1 1	ⓘ	
	⊖ N/A	TLR4→JUN	1 1 2 27	ⓘ	
	Statement Type		Sources	DB Link	
	DecreaseAmount		1 1	ⓘ	
	Complex		1 4	ⓘ	
	Dephosphorylation		1 1	ⓘ	
	Activation		1 1 3 21	ⓘ	
	Phosphorylation		1 2 3	ⓘ	
	IncreaseAmount		1 1	ⓘ	
SIK3→KRAS→JUN	Weight	Edge	Sources	DB Link	
	⊕ N/A	SIK3→KRAS	1 2	ⓘ	
	⊖ N/A	KRAS→JUN	1 2	ⓘ	

Fig. 9: Path search results with one of the edges of a path expanded to more detail.

INDRA NETWORK SEARCH MODULES REFERENCE

2.1 Autocomplete (`indra_network_search.autocomplete.autocomplete`)

2.2 Data Models (`indra_network_search.data_models`)

This module contains all data models used in the repository. They are all built around the `Pydantic BaseModel`.

2.3 Rest Models (`indra_network_search.data_models.rest_models`)

2.4 Pathfinding (`indra_network_search.pathfinding.pathfinding`)

2.5 Utility Functions (`indra_network_search.util`)

2.5.1 Util (`indra_network_search.util.curation_cache`)

2.6 Query Classes (`indra_network_search.query`)

2.7 Query Handler (`indra_network_search.query_handler`)

2.8 Rest API (`indra_network_search.rest_api`)

This documentation is here for completeness and for code documentation purposes. For a better description of the Rest API, read the [API docs](#).

2.9 Rest API Utilities (`indra_network_search.rest_util`)

2.10 Result Handlers (`indra_network_search.result_handler`)

2.11 Search API (`indra_network_search.search_api`)

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`