

Main interface

Similar to [Wolfram Alpha](#):

Query: |

A light gray rounded rectangle representing a user interface. At the top left, the text "Query:" is followed by a horizontal text input field with a thin gray border and rounded ends. The input field contains a single vertical cursor line. Below the input field is a large, empty space, suggesting a display area for search results.

Main interface

Suggested queries:

Eg: the two proof of concept questions

Query: |

Explore some things this Reasoning Tool can answer:

Which genetic conditions protect from disease ____

What is the outcome pathway for drug ____ and condition ____

Which drugs are the least well studied?

Main interface

Auto-complete suggestions:

Eg: data-types that we know about

Query: what genes are |

Transcription regulators

Functionally similar

Co-expressed

Explore some things this Reasoning Tool can answer:

Which genetic conditions protect from disease ____

What is the outcome pathway for drug ____ and condition ____

Which drugs are the least well studied?

Main interface

Error handling:

Eg: don't know how to interpret

Query: what nonsense show high phenotypic similarity to FANCC

Sorry, I don't know how to answer that question: *No available databases contain the data type "nonsense."*

Main interface

Error handling:

Eg: Typos

Query: which genetic conditions protect from ashma

Did you mean:

which genetic conditions protect from asthma

Main interface

Processing query:

Generate a dossier of search strategies (i.e. trees returned by Markov Chain)



Query: which genetic conditions protect from asthma

Result of NLP →

Input interpretation:

Find gene variants that reduce symptoms of bronchial asthma

association

Grounded data type

Data type →

Search strategy 1:

Join databases X, Y, and Z to find genetic variants that foo to bar that downregulate biz which causes asthma

Search strategy 2:

Join databases A and B to find genetic variants that baz to bif that alleviates symptoms of asthma

Search strategy 3:

Join databases C and D to find genetic variants that qux which prevents asthma

association

Main interface

Processing query:

User selects a search strategy

Query: which genetic conditions protect from asthma

Input interpretation:

Find **gene variants** that **reduce symptoms** of **bronchial asthma**

Search strategy 1:

Join databases **X**, **Y**, and **Z** to find **genetic variants** that **foo** to **bar** that **downregulate biz** which causes **asthma**

Search strategy 2:

Join databases **A** and **B** to find **genetic variants** that **baz** to **bif** that **alleviates symptoms** of **asthma**

Search strategy 3:

Join databases **C** and **D** to find **genetic variants** that **qux** which **prevents asthma**

Main interface

Processing query:

Search strategy is executed (modified Dijkstra)

Query: which genetic conditions protect from asthma

Executing search strategy...

Joining databases X, Y, and Z to find genetic variants that foo to bar that downregulate biz which causes asthma

Results:

Genetic variant X_1 protects from asthma (confidence C_1 %) [Show details](#)

Genetic variant X_2 protects from asthma (confidence C_2 %) [Show details](#)

Genetic variant X_2 protects from asthma (confidence C_2 %) [Show details](#)

...

[Export results to CSV/JSON](#)

Shows:

1. Full path through the Markov chain that gave this result.
2. Database queries used
3. Results of database API calls, etc.

Main interface

Showing details

Expose all the guts as to how the answer was obtained

Details for:

Genetic variant X_1 protects from asthma (confidence C_1 %)

Variant X_1 found in database X (API call here)

Database X contains content type foo (supporting info here)

Database Y contains content type foo and bar (supporting info here)
(API call to ground foo and bar)

(API call to get association from $X \rightarrow Y$)

Search database Z to find what grounded bar downregulates what
(Result of search here)

More API calls, steps executed, biological jargon, whatever else....