Drug Repurposing Explorer

Web Application Development Plan and Technical Specification

Document status	Development plan – Version 0.1
Current implementation scope	INDRA module
Planned extensions	PPI module; INDRA–PPI comparison

October 2026

CONFIDENTIAL – INTERNAL USE ONLY

This document and the associated materials are intended solely for the authorized project team. They must not be published, shared externally, uploaded to public repositories, or provided to external AI/chatbot/agent services without prior authorization.

Table of Contents

1. PURPOSE AND SCOPE 4

1.1 Current Development Objectives 4
1.1.1 Drug Ranking and Exploration 4
1.1.2 Mechanistic Explanation 4
1.1.3 Interactive Mechanistic Graph Visualization 5

2. SYSTEM ARCHITECTURE 5

2.1 High-Level Architecture 5
2.2 Scientific Analysis and Precomputed Results 6
2.3 Backend and Data-Access Layer 6
2.4 Web Application 6
2.5 Application Components and Future Extension 7

3. USER INTERFACE AND INTERACTION CONCEPT 7

3.1 Global Navigation 8
3.2 Disease Selection and Context 8
3.3 Drug Ranking and Selection 8
3.4 Dynamic Disease-Gene Selection 9
3.5 Drug–Gene Subgraph Selection 10
3.6 Selection and Path Summary 10

4. INDRA FUNCTIONAL WORKFLOW 11

4.1 Drug Ranking 12
4.2 Reachable Disease-Gene Selection 12
4.3 Mechanistic Subgraph Generation 12

5. INTERACTIVE INDRA GRAPH SPECIFICATION 13

5.1 Edge evidence details 13
5.2 Graph summary 13

6. DATA ARCHITECTURE AND FILE STRUCTURE 16

6.1 Standard Data Structure 16
6.2 Application Access to Result Files 16
6.3 Separation from the Scientific Analysis Pipeline 17

7. REQUIRED DATA FILES AND DATA CONTRACTS 17

7.1 disease_manifest.csv 18
Required fields 18
7.2 drug_rankings.csv 18
7.3 disease_module_open_targets.csv 19
7.4 mechanistic_paths.pkl.gz 20
Path structure 21
Reachable-gene determination 21
Path retrieval and graph construction 21

8. BACKEND DATA-ACCESS PLAN 22

8.1 Backend Responsibilities 22
8.2 Recommended API Structure 22

8.3 Reachable-Gene Service 23
8.4 Mechanistic Graph Service 23
8.5 Ranking Data Access 24
8.6 Implementation Requirements 25
9. REFERENCE VISUALIZATION CODE 25
9.1 Graph Representation 25
9.2 Relation, Evidence, and Provenance Information 26
9.3 Interactive Graph Behavior 26
10. DEVELOPMENT SCOPE AND RESPONSIBILITIES 26
10.1 Current Web-Development Scope 27
10.2 Outside the Current Web-Development Scope 27
10.3 Implementation and Change Management 27
11. IMPLEMENTATION PHASES 28
11.1 Phase 1 - Data Access and Validation 28
11.2 Phase 2 - Application Structure and Disease Selection 29
11.3 Phase 3 - Drug Ranking and Selection 29
11.4 Phase 4 - Reachable Disease-Gene Selection 29
11.5 Phase 5 - Mechanistic Graph and Evidence Exploration 29
11.6 Phase 6 – Integration and Validation 29
12. ACCEPTANCE CRITERIA 30
12.1 Disease and Data Access 30
12.2 Drug Ranking and Selection 30
12.3 Reachable Disease-Gene Selection 30
12.4 Mechanistic Path Retrieval and Graph Construction 31
12.5 Mechanistic Information and Evidence 31
12.6 Architecture and Scope 31
13. PROVIDED DEVELOPMENT RESOURCES 32
14. FUTURE EXTENSION: PPI AND INDRA–PPI COMPARISON 32
APPENDIX A – INITIAL DEVELOPER CHECKLIST 34

1. Purpose and Scope

This document defines the development plan for the web-based **Drug Repurposing Explorer**.

The current implementation focuses on providing access to precomputed INDRA-based drug-repurposing results and an interactive explorer for the corresponding mechanistic paths.

The **scientific analysis pipeline is separate from the web application**. The web application uses the precomputed disease-specific results generated by the pipeline and provides functionality for exploring drug rankings, disease-associated genes, mechanistic paths, and the underlying INDRA evidence.

The application **architecture should allow additional analysis components** to be integrated in later development phases. In particular, a PPI-based analysis component and an INDRA–PPI comparison functionality are planned as future extensions. The current development phase focuses on the **INDRA component**.

1.1 Current Development Objectives

The current development focuses on three main capabilities: **drug ranking and exploration, mechanistic explanation of drug–disease relationships, and interactive visualization of the underlying INDRA mechanistic paths.**

1.1.1 Drug Ranking and Exploration

The application must provide disease-specific drug-repurposing rankings based on the precomputed analysis results.

- Allow the user to select a disease.
- Display the complete ranked list of candidate drugs for the selected disease.
- Support the existing ranking scores and ranking methods generated by the drug-repurposing pipeline.
- Allow the ranking to be sorted and explored according to the available scores and metrics.
- Allow the user to select up to five drugs for further mechanistic exploration.
- Provide access to the complete ranking results.

1.1.2 Mechanistic Explanation

The application must provide access to the mechanistic information underlying the ranked drug candidates.

- Identify the disease genes that are mechanistically reachable from the selected drug or drugs.
- Dynamically update the available disease genes according to the current drug selection.
- Allow the user to select up to five reachable disease genes for detailed exploration.
- Retrieve the complete stored mechanistic paths connecting the selected drugs to the selected disease genes.

- Provide access to the corresponding INDRA relations and available evidence information supporting these paths.

1.1.3 Interactive Mechanistic Graph Visualization

The application must provide an interactive graphical representation of the selected drug–gene mechanisms.

- Generate a graph from the complete stored mechanistic paths corresponding to the selected drug–gene pairs.
- Clearly distinguish selected drugs, intermediate nodes, and disease genes.
- Preserve the relationship between each selected drug and its corresponding mechanistic paths.
- Display INDRA relations and associated evidence information through the interactive graph.
- Support interactive exploration of nodes, edges, relations, and evidence.

2. System Architecture

The **Drug Repurposing Explorer** is designed as a layered web application that provides structured access to precomputed drug-repurposing results and supports their interactive scientific exploration.

The architecture separates the **scientific analysis pipeline**, the **precomputed result layer**, the **backend/data-access layer**, and the **user-facing web application**.

The current development focuses on the **INDRA component**, providing access to disease-specific drug rankings, mechanistic paths, and the underlying INDRA relations and evidence. The architecture should also allow PPI-based results and INDRA–PPI comparison functionality to be integrated in later development phases.

2.1 High-Level Architecture

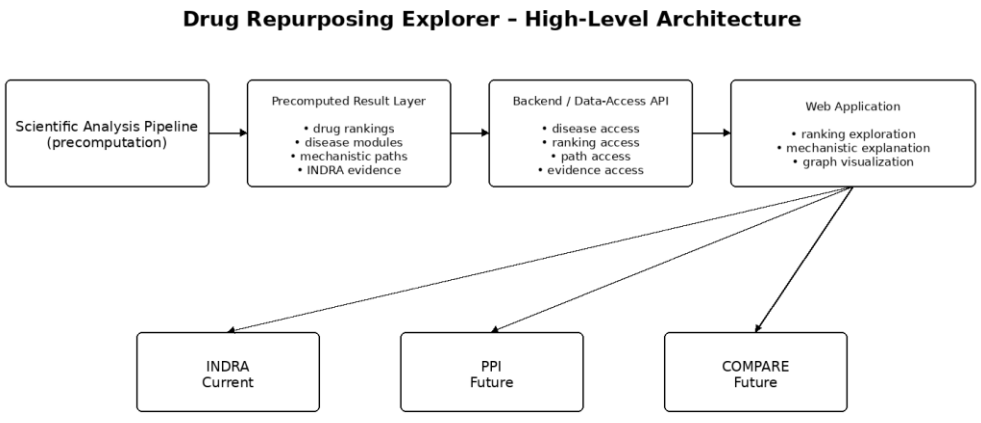


Figure 1. High-level architecture of the Drug Repurposing Explorer.

The current development phase focuses on the INDRA component. PPI-based analysis and INDRA–PPI comparison are planned extensions.

2.2 Scientific Analysis and Precomputed Results

The scientific analysis pipeline performs the computational drug-repurposing analysis and generates disease-specific results. These precomputed results constitute the scientific data source of the Drug Repurposing Explorer.

For the current INDRA component, the result layer provides the information required for:

- disease-specific drug rankings and ranking metrics;
- disease-associated genes;
- complete mechanistic paths connecting candidate drugs to reachable disease genes; and
- INDRA relations and supporting evidence associated with the mechanistic paths.

The web application operates on these precomputed results. Execution or modification of the scientific analysis pipeline is not part of the web application.

2.3 Backend and Data-Access Layer

The backend provides the interface between the precomputed scientific results and the web application.

Its responsibilities include:

- providing access to the available diseases
- retrieving disease-specific drug rankings and ranking metrics
- retrieving disease genes that are mechanistically reachable from selected drugs
- retrieving complete mechanistic paths for selected drug–gene pairs
- retrieving the INDRA relations and evidence associated with those paths
- providing these data to the web application in a structured format

The data-access interface should be clearly defined so that the frontend is independent of the physical organization and storage format of the underlying result files.

2.4 Web Application

The web application provides the interactive user interface of the Drug Repurposing Explorer.

For the current INDRA component, it supports the three principal functions defined in Section 1:

Drug Ranking and Exploration

The user selects a disease and explores the corresponding precomputed drug-repurposing ranking using the available ranking scores and metrics.

Mechanistic Explanation

For selected drug candidates, the application provides access to the disease genes reachable through the stored INDRA mechanistic paths and to the mechanistic information connecting the drugs and disease genes.

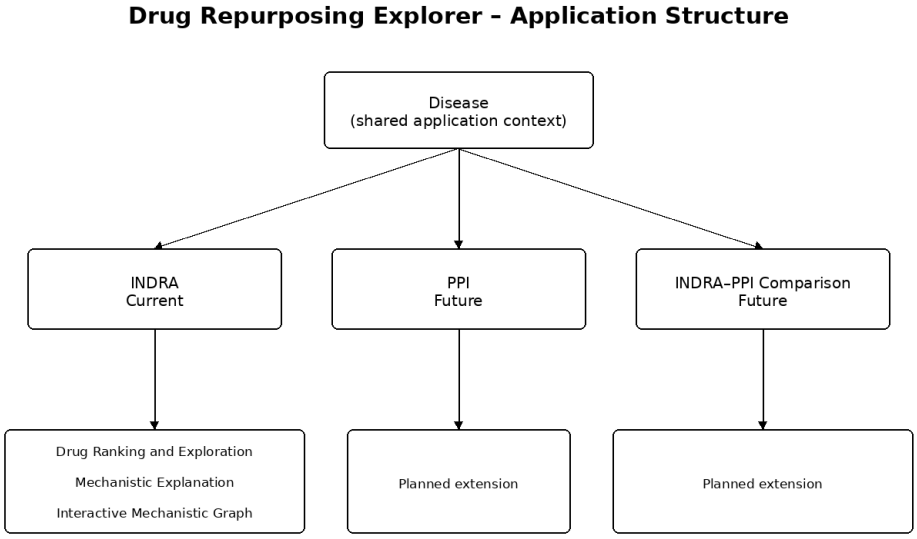
Interactive Mechanistic Graph Visualization

The selected drug–gene relationships can be explored as an interactive graph constructed from the complete stored mechanistic paths, including the corresponding INDRA relations and supporting evidence.

2.5 Application Components and Future Extension

The application uses a common disease context. Once a disease is selected, the corresponding analysis components can provide their disease-specific results.

The intended application structure is:



The **INDRA component** is the focus of the current development phase.

The architecture should allow the **PPI component** and the **INDRA–PPI comparison component** to be integrated later while retaining the same overall application structure and disease-centered workflow.

3. User Interface and Interaction Concept

The user interface of the Drug Repurposing Explorer should follow a **disease-centered workflow**. A single disease is selected first, and all subsequent ranking and mechanistic exploration is performed within the context of that disease.

The INDRA interface consists of two closely connected functions:

1. **Drug Ranking and Selection**
2. **Mechanistic Path Exploration**

The mechanistic exploration workflow should follow the interaction logic defined in this specification: the user first selects drugs, the available disease genes are dynamically restricted to genes reachable from those drugs, and the selected drug–gene pairs are then used to retrieve and visualize their complete stored mechanistic paths.

3.1 Global Navigation

The application should provide a common top-level navigation structure for the main analysis components:

- **INDRA** – current development scope
- **PPI** – future extension
- **COMPARE** – future extension

The current implementation focuses on the INDRA component. PPI and COMPARE may be represented in the application structure as future components but do not require functional implementation in the current development phase.

3.2 Disease Selection and Context

One disease is active at a time.

The list of available diseases must be loaded from the provided **disease_manifest.csv** file.

The disease selector should:

- load the available disease identifiers and names from disease_manifest.csv
- support search by disease name
- display the human-readable disease name
- retain the corresponding disease identifier internally
- use the selected disease identifier to access the corresponding drug-ranking and mechanistic result files
- reset the current drug, gene, and graph selections when the active disease is changed

The selected disease defines the context for all subsequent INDRA ranking and mechanistic exploration functions.

3.3 Drug Ranking and Selection

For the selected disease, the application should display the precomputed drug-repurposing ranking.

The ranking interface should:

- display the ranked candidate drugs
- provide the available precomputed ranking scores and metrics
- allow the user to select the ranking criterion
- support searching and filtering the displayed drugs

- allow the user to select **up to five drugs** for mechanistic exploration
- provide access to the complete ranking results

Drug selection for mechanistic exploration must originate from the ranking results of the currently selected disease.

The selected drugs become the **source nodes** for the mechanistic path exploration.

3.4 Dynamic Disease-Gene Selection

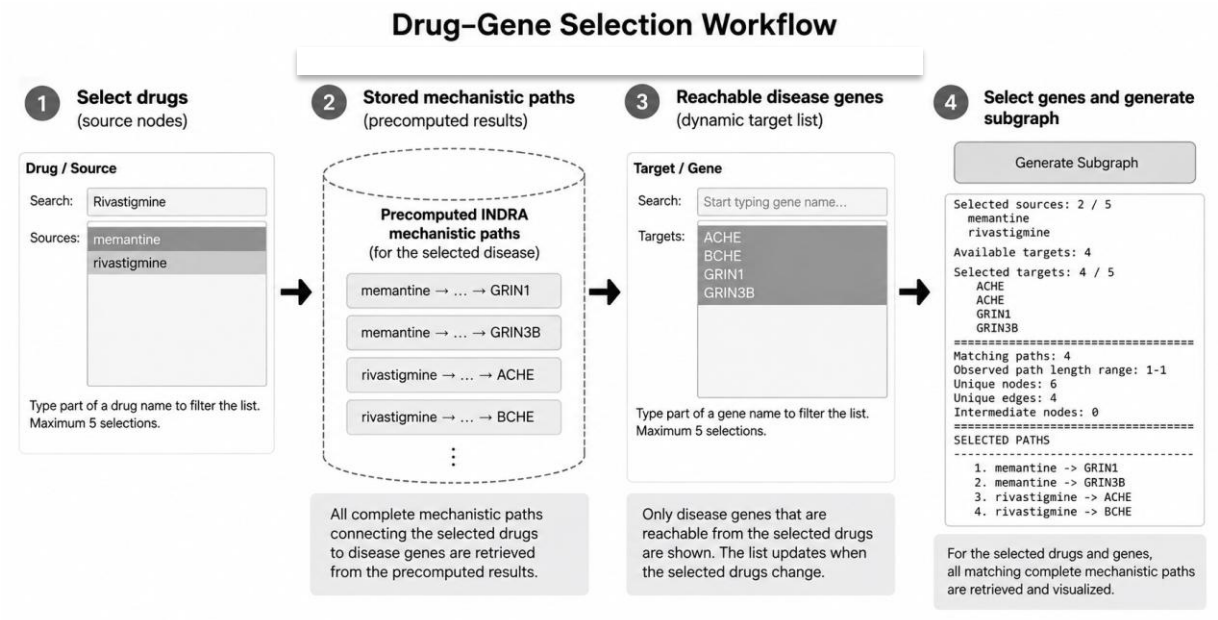
Once one or more drugs have been selected, the application should determine which disease genes are reachable from those drugs through the stored INDRA mechanistic paths.

Only these reachable disease genes should be presented for selection.

The gene-selection interface should:

- update dynamically when the selected drugs change
- show only genes reachable from at least one selected drug
- support gene-name search
- allow selection of **up to five disease genes**
- clearly indicate the number of available and selected genes

This dependency is important: the disease-gene list is **not an independent static list**. It is determined by the currently selected drugs and the available stored mechanistic paths.



3.5 Drug–Gene Subgraph Selection

After selecting the drugs and disease genes, the user should be able to initiate mechanistic exploration through a clear action such as:

Generate Mechanistic Graph

For the selected drug and gene sets, the application should identify all stored mechanistic paths satisfying:

source ∈ selected drugs

AND

target ∈ selected disease genes

All matching **complete paths** should be retained.

The system must not reduce the result to only the shortest path or a single representative path.

For example, for two selected drugs and four selected genes, the application determines which selected drug–gene combinations actually have stored mechanistic paths and retrieves all complete paths corresponding to those combinations.

3.6 Selection and Path Summary

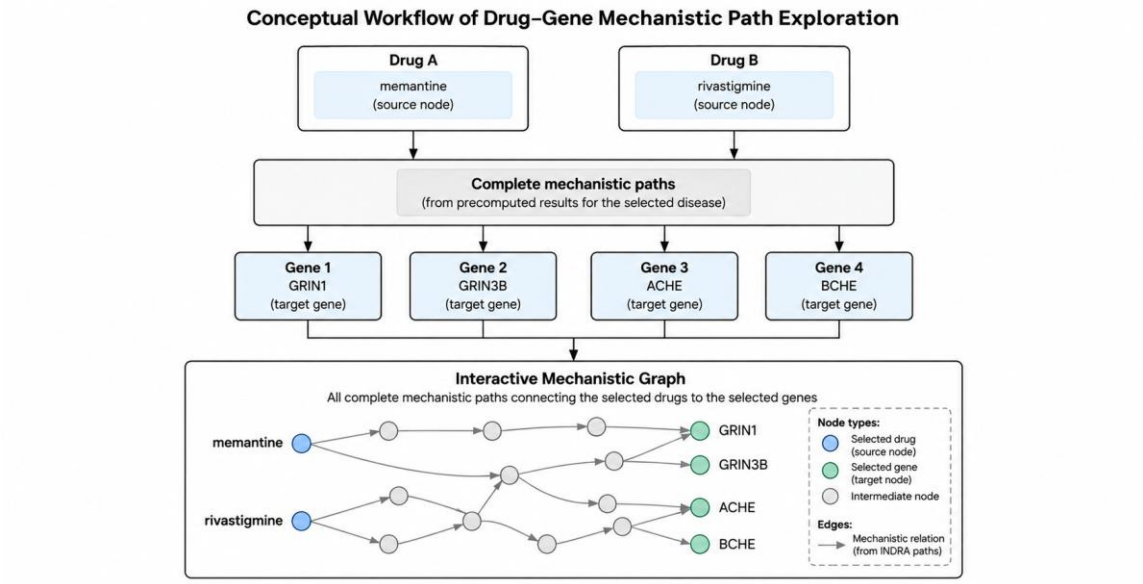
Before or together with the graphical visualization, the application should provide a concise summary of the selected mechanistic subgraph.

The summary should include at least:

- number and names of selected drugs
- number and names of selected disease genes
- number of matching complete paths
- observed path-length range
- number of unique nodes
- number of unique edges
- number of intermediate nodes

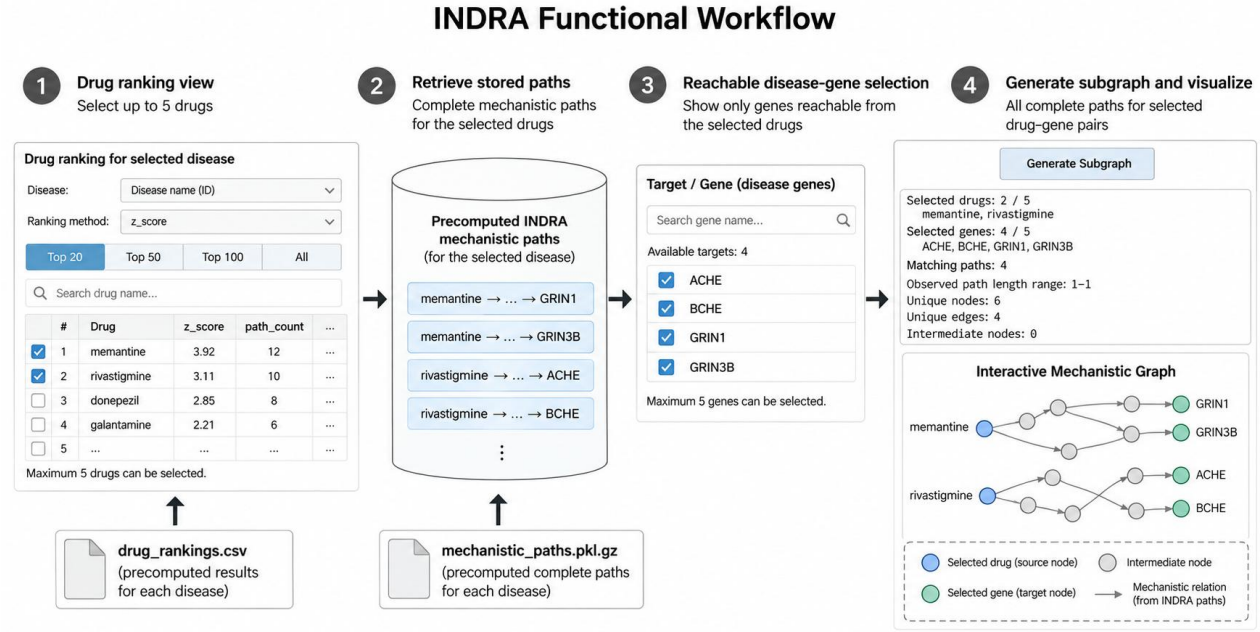
The individual matching drug–gene paths should also be accessible for inspection.

Conceptually:



The detailed graphical representation and evidence-level interaction are specified in the following section.

4. INDRA Functional Workflow



The INDRA component operates entirely on precomputed disease-specific results. The web application does not perform or rerun the scientific analysis. Its role is to load, filter, connect, and visualize the available ranking and mechanistic-path results.

4.1 Drug Ranking

For the selected disease, the drug ranking is loaded from the corresponding **drug_rankings.csv** file.

The application must display the available ranking metrics and support selection of the ranking criterion. The ranking itself is not recalculated by the web application.

Category	Required fields / behavior
Identity	Disease_Id, disease, drug
Core statistics	k, path_count, shortest_path, avg_belief, d_drug, mu_k, sigma_k, z_score
Individual ranks	z_score_rank, k_rank, path_count_rank, avg_belief_rank
Consensus ranks	mean_rank, median_rank, weighted2_rank, weighted3_rank, borda_rank, rrf_rank
Ranking views	Top 20, Top 50, Top 100, All
Search	Search by drug name
Selection	Maximum five drugs
Export	Download the complete ranking

The result file may contain additional structured fields, including target_distances and target_ot_metadata. These should not be displayed as ordinary columns in the main ranking table.

4.2 Reachable Disease-Gene Selection

After one or more drugs have been selected, the application must use the stored mechanistic paths in **mechanistic_paths.pkl.gz** to identify the disease genes reachable from at least one selected drug.

Only these reachable disease genes are presented in the gene selector.

The available gene list must update whenever the selected drug set changes.

A maximum of five disease genes may be selected.

4.3 Mechanistic Subgraph Generation

For the selected drugs and disease genes, the application retrieves **all complete stored mechanistic paths** connecting each selected drug to each selected reachable disease gene.

The interactive mechanistic graph is constructed as the **union of all matching complete paths**.

By default, all complete stored paths matching the selected drug–gene pairs should be included. Path length may be used for filtering or interactive exploration without modifying the underlying stored path data.

5. Interactive INDRA Graph Specification

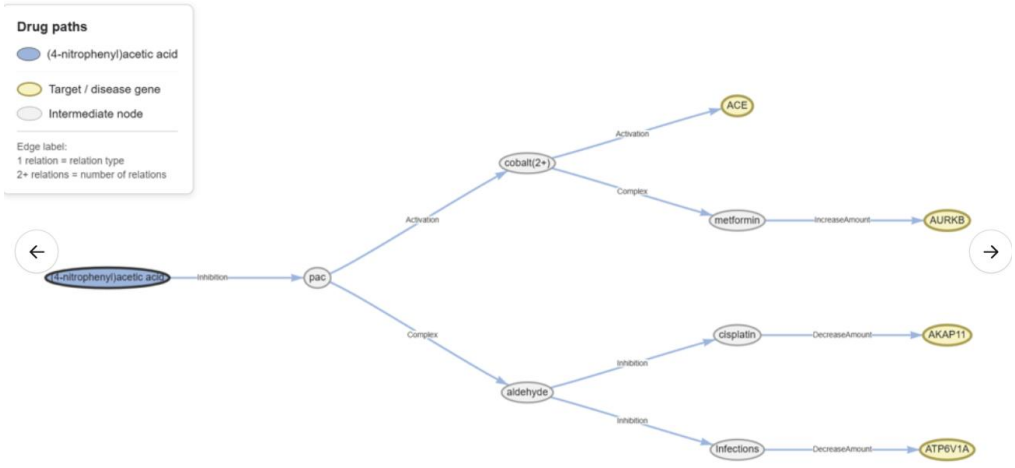
5.1 Edge evidence details

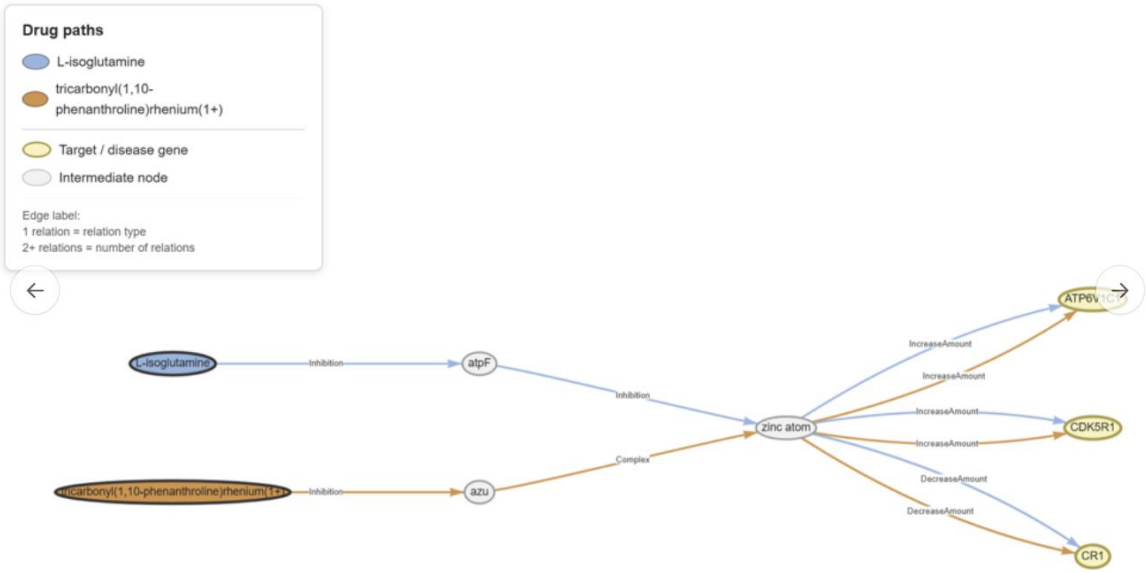
The graph must preserve access to the scientific metadata stored for each path edge. A tooltip, popover, or side panel may be used. The UI should expose the following where available:

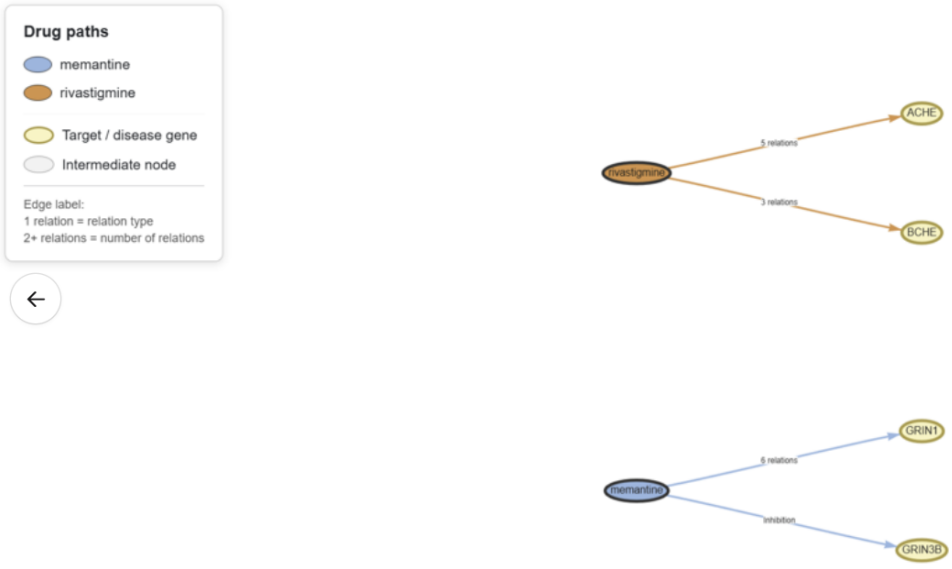
- edge endpoints and selected drug provenance;
- INDRA statement/relation types (stmt_types);
- English INDRA statements (stmt_english);
- belief values (beliefs);
- corrected edge weights (corr_weights);
- evidence source counts (source_counts);
- statement hashes (stmt_hashes);
- statement-level PMIDs and aggregated PMIDs (stmt_pmids / pmids);
- publication years (years), where populated.

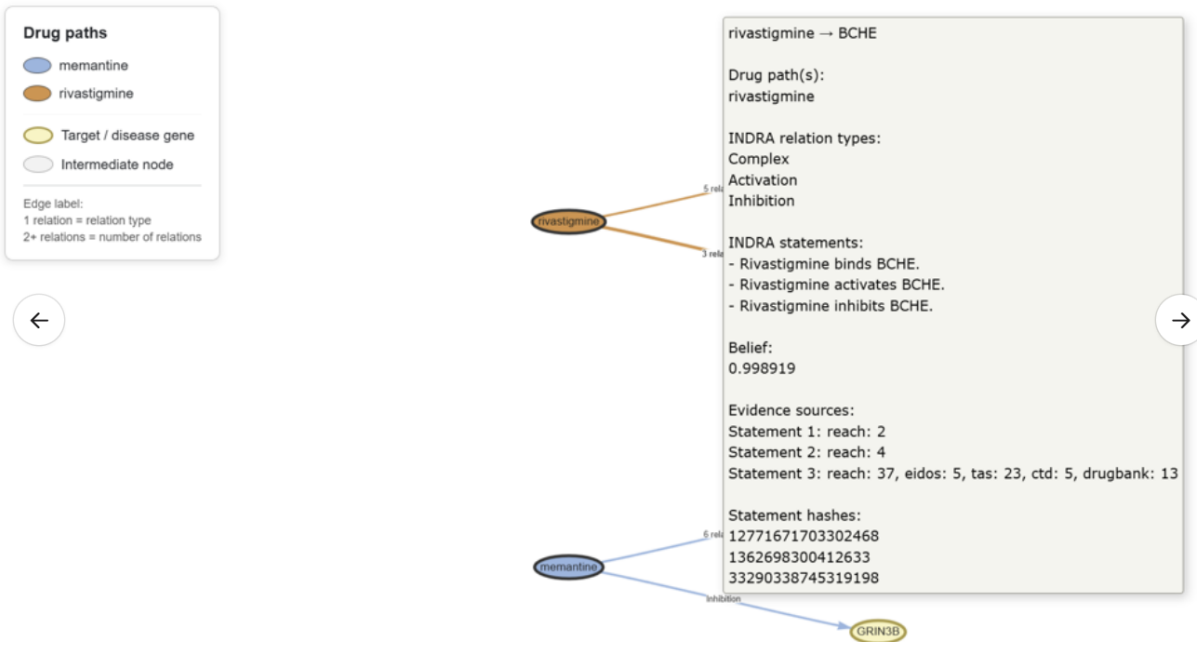
5.2 Graph summary

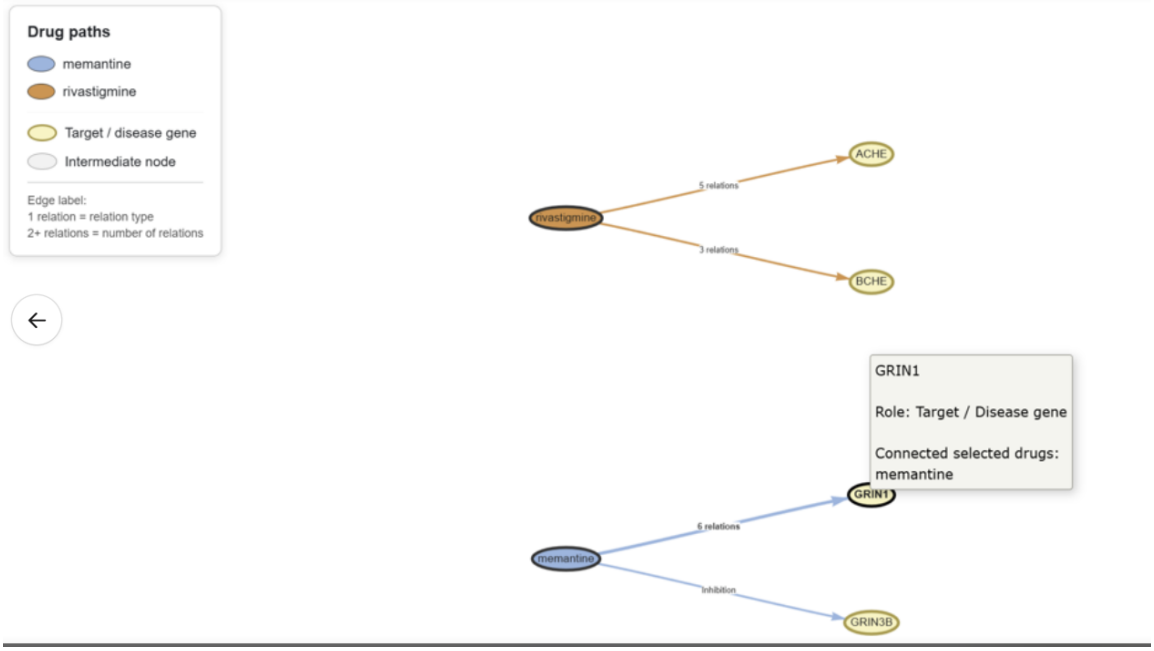
After graph generation, show at least: number of selected drugs, selected genes, complete mechanistic paths, graph nodes, graph edges, intermediate nodes, minimum path length, and maximum path length.




Drug paths
L-isoglutamine
tricarbonyl(1,10-phenanthroline)rhenium(1+)
Target / disease gene
Intermediate node
Edge label:
1 relation = relation type
2+ relations = number of relations
L-isoglutamine
Inhibition
atpF
Inhibition
tricarbonyl(1,10-phenanthroline)rhenium(1+)
Inhibition
azu
Complex
zinc atom
IncreaseAmount
IncreaseAmount
IncreaseAmount
IncreaseAmount
DecreaseAmount
DecreaseAmount
ATP6V1C2
CDK5R1
CR1


Drug paths
memantine
rivastigmine
Target / disease gene
Intermediate node
Edge label:
1 relation = relation type
2+ relations = number of relations
rivastigmine
5 relations
3 relations
ACHE
BCHE
memantine
6 relations
Inhibition
GRIN1
GRIN3B


Drug paths
memantine
rivastigmine
Target / disease gene
Intermediate node
Edge label:
1 relation = relation type
2+ relations = number of relations
rivastigmine → BCHE
Drug path(s):
rivastigmine
INDRA relation types:
Complex
Activation
Inhibition
INDRA statements:
- Rivastigmine binds BCHE.
- Rivastigmine activates BCHE.
- Rivastigmine inhibits BCHE.
Belief:
0.998919
Evidence sources:
Statement 1: reach: 2
Statement 2: reach: 4
Statement 3: reach: 37, eidos: 5, tas: 23, ctd: 5, drugbank: 13
Statement hashes:
12771671703302468
1362698300412633
33290338745319198
rivastigmine
memantine
Inhibition
GRIN3B


Drug paths
memantine
rivastigmine
Target / disease gene
Intermediate node
Edge label:
1 relation = relation type
2+ relations = number of relations
ACHE
5 relations
rivastigmine
3 relations
BCHE
GRIN1
Role: Target / Disease gene
Connected selected drugs:
memantine
GRIN1
6 relations
memantine
Inhibition
GRIN3B

6. Data Architecture and File Structure

The web application operates on a standardized set of precomputed result files provided for each disease. These files constitute the data interface between the scientific analysis and the web application.

The current implementation uses a file-based data layer. A separate database is not required.

The web application must access the data through a configurable data root and must not depend on hard-coded absolute server paths.

6.1 Standard Data Structure

The following logical structure should be used for the INDRA component:

```
data/
├── disease_manifest.csv
└── INDRA/
    └── diseases/
        ├── <disease_id_1>/
        │   ├── drug_rankings.csv
        │   ├── disease_module_open_targets.csv
        │   └── mechanistic_paths.pkl.gz
        ├── <disease_id_2>/
        │   ├── drug_rankings.csv
        │   ├── disease_module_open_targets.csv
        │   └── mechanistic_paths.pkl.gz
        └── ...
```

The **`disease_manifest.csv`** file defines the diseases available in the application and provides the disease identifiers used to locate the corresponding disease-specific result directory.

Each disease directory contains the precomputed data required by the INDRA component:

File	Purpose
`drug_rankings.csv`	Disease-specific drug ranking and associated ranking metrics
`disease_module_open_targets.csv`	Disease-gene information associated with the selected disease
`mechanistic_paths.pkl.gz`	Complete stored mechanistic paths used for reachable-gene selection and graph generation

6.2 Application Access to Result Files

The backend should provide a consistent data-access layer between these files and the frontend.

The web application should use the selected `disease_id` to locate the corresponding disease directory and access the required result files.

The physical location of the data must be configurable, for example through a **`DATA_ROOT`** configuration parameter.

Application code should therefore refer to logical paths such as:

```
<DATA_ROOT>/disease_manifest.csv

<DATA_ROOT>/INDRA/diseases/<disease_id>/drug_rankings.csv

<DATA_ROOT>/INDRA/diseases/<disease_id>/disease_module_open_targets.csv

<DATA_ROOT>/INDRA/diseases/<disease_id>/mechanistic_paths.pkl.gz
```

This allows the same application code to be used with development, testing, and deployed data locations without modifying the application logic.

6.3 Separation from the Scientific Analysis Pipeline

The scientific analysis pipeline and the web application are separate components.

The web application consumes the provided precomputed result files and does not execute or modify the scientific analysis pipeline. Internal computational files that are not required by the web application are outside the web application data interface.

7. Required Data Files and Data Contracts

The INDRA component uses one application-level disease manifest and three disease-specific result files. Together, these files define the data interface between the precomputed scientific results and the web application.

The web application consumes these files as provided. Scientific analysis and generation of the result files are outside the scope of the web application.

File	Required	Purpose in the Web Application
`disease_manifest.csv`	Yes	Defines the diseases available in the application and provides their identifiers and display names.
`drug_rankings.csv`	Yes	Provides the complete disease-specific drug ranking together with the precomputed ranking and statistical fields.
`disease_module_open_targets.csv`	Yes	Provides disease-gene identifiers and associated Open Targets metadata.
`mechanistic_paths.pkl.gz`	Yes	Provides the complete stored INDRA mechanistic paths and associated path and edge metadata used for reachable-gene selection and mechanistic graph generation.

A separate reference implementation, **`indra_mechanistic_graph_reference.py`**, is provided to support implementation of the existing path-selection and graph-visualization behavior. This file is reference code and is not part of the application data interface.

7.1 disease_manifest.csv

The `disease_manifest.csv` file defines the diseases available for selection in the application.

The disease selector described in Section 3.2 must be populated from this file.

Required fields

Column	Meaning
`disease_id`	Stable disease identifier used to locate the corresponding disease-specific result directory.
`disease_name`	Human-readable disease name displayed in the application.

Example:

```
disease_id,disease_name
MONDO_0000376,respiratory system cancer
```

The web application should use `disease_name` for display and `disease_id` internally for locating and accessing the corresponding disease-specific result files.

7.2 drug_rankings.csv

For each disease, `drug_rankings.csv` contains the precomputed drug-ranking results used by the **Drug Ranking and Exploration** interface.

The application does not calculate new rankings. It reads and displays the ranking results and allows the user to select among the available precomputed ranking criteria.

The `drug_rankings.csv` file contains the following fields:

```
diseaseId
disease
primary_order
drug
k
path_count
shortest_path
avg_belief
avg_corr_weight
target_distances
target_ot_metadata
d_drug
mu_k
```

```
sigma_k
z_score
z_score_rank
shortest_path_rank
k_rank
path_count_rank
avg_belief_rank
avg_corr_weight_rank
mean_rank_score
median_rank_score
weighted2_rank_score
weighted3_rank_score
borda_score
rrf_score
mean_rank
median_rank
weighted2_rank
weighted3_rank
borda_rank
rrf_rank
```

Numeric fields must be handled as numeric values by the backend so that ranking, sorting, and filtering behave correctly.

The `target_distances` and `target_ot_metadata` fields contain structured serialized information. They may be used for detailed views where appropriate but should not be displayed as ordinary columns in the main drug-ranking table.

The ranking interface should use this file to support:

- selection of the ranking criterion
- sorting and exploration of the available ranking metrics
- Top 20, Top 50, Top 100, and complete ranking views
- search by drug name
- selection of up to five drugs for mechanistic exploration
- download of the complete ranking

7.3 disease_module_open_targets.csv

For each disease, `disease_module_open_targets.csv` provides the disease-gene information and associated Open Targets metadata.

The file contains the following fields:

```
diseaseId
disease_name
targetId
gene_symbol
ot_rank
ot_score
relative_score
```

```
score_at_rank_1
```

The fields have the following roles:

Field	Purpose
`diseaseId`	Stable identifier of the selected disease.
`disease_name`	Human-readable disease name.
`targetId`	Stable identifier of the disease-associated target/gene.
`gene_symbol`	Human-readable gene symbol used in the interface.
`ot_rank`	Open Targets rank associated with the gene for the disease.
`ot_score`	Open Targets association score.
`relative_score`	Precomputed relative association score.
`score_at_rank_1`	Reference score used in the precomputed disease-module results.

This file provides the authoritative disease-gene metadata for the web application.

It defines the disease-associated gene set and associated metadata. However, the complete disease-gene list must **not** automatically be presented as the set of selectable mechanistic targets. The selectable gene list is determined dynamically from the stored mechanistic paths for the currently selected drugs, as described in Sections 3.4 and 4.2.

7.4 mechanistic_paths.pkl.gz

For each disease, `mechanistic_paths.pkl.gz` contains the precomputed complete INDRA mechanistic paths used by the **Mechanistic Explanation** and **Interactive Mechanistic Graph Visualization** functions.

The file is a gzip-compressed Python pickle containing a list of path records. Each record represents one complete mechanistic path from a source drug/molecule to a disease-gene target.

Each path record contains the following fields:

```
diseaseId
disease_name
molecule
molecule_ns
molecule_id
gene
gene_targetId
gene_ot_rank
gene_ot_score
gene_ot_relative_score
gene_ot_score_at_rank_1
path_nodes
path_length
beliefs
corr_weights
stmt_types
source_counts
```

```
stmt_hashes
stmt_english
stmt_pmids
pmids
years
```

Path structure

For each path record:

- `path_nodes[0]` represents the source drug/molecule;
- `path_nodes[-1]` represents the disease-gene target;
- nodes between the first and last elements of `path_nodes` represent intermediate nodes;
- `path_length` describes the length of the complete stored path; and
- edge-level arrays correspond to the consecutive edges represented by `path_nodes`.

The associated arrays provide the mechanistic and evidence information required for detailed graph exploration, including INDRA statement types, belief information, correlation weights, source information, statement hashes, English statement representations, PMIDs, and publication years.

Reachable-gene determination

After one or more drugs are selected, the application must use `mechanistic_paths.pkl.gz` to determine which disease genes are reachable from at least one selected drug.

Only these reachable disease genes should be offered in the gene selector.

The reachable-gene set must therefore be recalculated whenever the selected drug set changes.

Path retrieval and graph construction

After the user selects the drugs and disease genes, the application must retrieve every stored path record satisfying both conditions:

```
source drug ∈ selected drugs
AND
target gene ∈ selected disease genes
```

All matching **complete stored paths** must be retained.

The Interactive Mechanistic Graph is constructed from the **union of these complete paths**.

The application must not:

- reduce the result to a single representative path
- retain only the shortest path
- truncate intermediate nodes

By default, all complete stored paths matching the selected drug–gene pairs should be included in the mechanistic graph. Path length should be available as descriptive path metadata and may also be used for filtering or interactive exploration if required.

A disease gene may be reachable from **more than one selected drug**. The graph must preserve all such drug–gene relationships and all corresponding complete mechanistic paths.

8. Backend Data-Access Plan

The **backend** provides the data-access layer between the precomputed INDRA result files and the web application.

Its primary responsibility is to load and interpret the standardized result files defined in Sections 6 and 7 and provide the frontend with structured, browser-compatible data.

The frontend should not access server file paths directly and should not be responsible for reading Python pickle files. Loading and processing of `mechanistic_paths.pkl.gz` must therefore be handled by the backend.

8.1 Backend Responsibilities

The backend should provide access to the following application functions:

- disease discovery and selection
- disease-specific drug rankings
- disease-gene metadata
- determination of genes reachable from the currently selected drugs
- retrieval of complete mechanistic paths for selected drug–gene pairs
- construction of the data representation required for interactive mechanistic graph visualization
- download of complete drug-ranking results

The backend operates on the precomputed result files and does not execute the scientific analysis pipeline or recalculate the underlying drug-repurposing results.

8.2 Recommended API Structure

The following API structure defines the required logical data-access functions. Exact endpoint naming may be adapted during implementation while preserving the same functionality and data contract.

Endpoint	Responsibility
`GET /api/diseases`	Read `disease_manifest.csv` and return the diseases available for selection.
`GET /api/diseases/{id}/indra/rankings`	Read `drug_rankings.csv` and return the disease-specific ranking data. Support ranking

	criterion, result limit, sorting, and drug-name search as required by the ranking interface.
GET /api/diseases/{id}/indra/genes	Return disease-gene metadata from `disease_module_open_targets.csv`.
POST /api/diseases/{id}/indra/reachable-genes	Given the currently selected drugs, determine and return the disease genes reachable through at least one stored mechanistic path.
POST /api/diseases/{id}/indra/graph	Given the selected drugs and disease genes, retrieve all matching complete stored paths and return the data required for interactive graph visualization.
GET /api/diseases/{id}/indra/rankings/download	Return the complete disease-specific drug-ranking file for download.

8.3 Reachable-Gene Service

Reachable-gene determination is a dynamic operation and should be handled separately from access to the complete disease-gene module.

The request should contain the currently selected drugs.

For example:

```
{
  "drugs": [
    "memantine",
    "rivastigmine"
  ]
}
```

Using `mechanistic_paths.pkl.gz`, the backend identifies every disease gene for which at least one complete stored path originates from one of the selected drugs.

The response should provide the reachable genes together with the metadata required by the interface, such as the gene symbol and stable target identifier.

This service supports the interaction defined in Section 3.4:

- **Selected drugs → stored mechanistic paths → reachable disease genes → gene selection**

Whenever the selected drug set changes, the frontend should request an updated reachable-gene set.

8.4 Mechanistic Graph Service

The graph service receives the selected drugs and disease genes.

For example:

```
{
```

```
  "drugs": [
    "memantine",
    "rivastigmine"
  ],
  "genes": [
    "GRIN1",
    "GRIN3B",
    "ACHE",
    "BCHE"
  ]
}
```

The backend retrieves all complete stored paths satisfying:

```
source drug ∈ selected drugs
AND
target gene ∈ selected genes
```

By default, all matching complete paths should be retained.

The response should provide a browser-compatible representation containing the information required to:

- construct the union of the matching complete paths
- distinguish source drugs, intermediate nodes, and disease-gene targets
- preserve the association between each path and its originating drug
- represent shared target genes and overlapping mechanistic paths
- display INDRA relations and associated evidence
- provide path-level and graph-level summary information
- support interactive graph exploration

The backend may transform the stored path records into an appropriate JSON representation for the frontend. The frontend should not need to interpret the original Python pickle structure.

8.5 Ranking Data Access

The ranking service reads the disease-specific `drug_rankings.csv` file and provides the data required by the Drug Ranking and Exploration interface.

It should support the available precomputed ranking criteria and the ranking views defined in Section 4.1, including:

- ranking criterion selection
- Top 20, Top 50, Top 100, and complete ranking views
- sorting
- drug-name search
- selection of drugs for subsequent mechanistic exploration

The backend must not recalculate the scientific ranking metrics or ranking scores. It provides access to the precomputed values contained in `drug_rankings.csv`.

The complete ranking CSV should also remain available for download.

8.6 Implementation Requirements

The specific backend framework is not prescribed by this development plan.

The implementation should, however, maintain a clear separation between:

Precomputed result files → Backend data-access layer → Frontend application

The backend should use the configurable data root defined in Section 6 rather than hard-coded absolute server paths.

Data returned to the frontend should use standard browser-compatible representations such as JSON and CSV. Python-specific serialized formats, including the pickle representation used by `mechanistic_paths.pkl.gz`, remain within the backend data-access layer.

9. Reference Visualization Code

A reference implementation, **`indra_mechanistic_graph_reference.py`**, will be provided together with the required data files.

The reference implementation defines the functional behavior for mechanistic path selection, subgraph construction, and interactive INDRA graph visualization.

It includes the following functionality:

- construction of source–target indices from the stored mechanistic paths
- selection of drugs as source nodes
- dynamic identification of disease genes reachable from the selected drugs
- selection of disease genes as target nodes
- retrieval of all complete stored paths corresponding to the selected drug–gene pairs
- construction of the mechanistic subgraph as the union of the selected complete paths
- interactive visualization of the resulting INDRA mechanistic graph

9.1 Graph Representation

The mechanistic graph should be represented as a **directed graph** reflecting the direction of the stored INDRA mechanistic paths.

The visualization should distinguish between:

- selected drugs as source nodes
- disease genes as target nodes
- intermediate nodes occurring along the mechanistic paths

A disease gene may be reachable from more than one selected drug. Shared target genes and overlapping mechanistic paths must therefore be represented without duplicating the underlying biological entity.

The graph must preserve the complete mechanistic paths associated with each selected drug–gene pair.

9.2 Relation, Evidence, and Provenance Information

The graph visualization should provide access to the INDRA information associated with the represented relationships, including, where available:

- INDRA statement/relation type
- belief information
- correlation weight
- evidence/source information
- statement representation
- PMID information
- publication year

This information may be presented through appropriate interactive mechanisms such as node or edge selection, tooltips, detail panels, or other suitable interface elements.

9.3 Interactive Graph Behavior

The graph should support interactive exploration of the selected mechanistic network, including:

- inspection of nodes and edges
- identification of the drug associated with each mechanistic path
- exploration of shared or overlapping paths
- access to relation and evidence information
- zooming and navigation within the graph
- inspection of the complete paths contributing to the displayed subgraph

The specific frontend graph library is not prescribed. An appropriate web-based visualization technology may be selected during implementation.

The implementation must preserve the **path structure, drug–gene relationships, INDRA relations, evidence, and provenance semantics** defined by the provided data and reference code.

10. Development Scope and Responsibilities

The current development scope is defined by this specification and the provided data contracts and reference materials. The implementation should follow these requirements consistently so that the web application remains aligned with the scientific results and can be extended in a controlled manner.

Questions or proposed changes that affect scientific interpretation, ranking logic, mechanistic path handling, data contracts, or the defined application behavior should be discussed before implementation.

10.1 Current Web-Development Scope

The current development includes:

- implementation of the INDRA component of the Drug Repurposing Explorer
- access to the standardized precomputed result files defined in Sections 6 and 7
- implementation of the backend data-access functions defined in Section 8
- disease selection and disease-specific data access
- drug-ranking exploration, search, sorting, and selection
- dynamic determination and selection of reachable disease genes
- retrieval of complete stored mechanistic paths for selected drug–gene pairs
- construction and interactive visualization of the corresponding mechanistic graph
- access to the available INDRA relation, evidence, and provenance information
- ranking-data download functionality
- an application architecture that can accommodate the planned PPI and INDRA–PPI comparison components in later development phases

10.2 Outside the Current Web-Development Scope

The following are outside the current development scope:

- modification of the scientific INDRA analysis pipeline
- recalculation or modification of the precomputed scientific results
- modification of the existing ranking metrics or ranking logic
- creation of new scientific ranking scores or metrics
- modification of the disease-gene data or upstream disease-gene selection logic
- generation, inference, or modification of mechanistic paths not contained in the provided result files
- modification of the defined data contracts without prior discussion
- implementation of the PPI component before its specification and corresponding data interface are provided
- implementation of the INDRA–PPI comparison component before its specification is provided

10.3 Implementation and Change Management

The provided result files and data contracts should be treated as the authoritative input to the web application.

The web application may transform these data into appropriate browser-compatible representations for presentation and interaction, but such transformations must preserve the scientific content and semantics of the source data.

Implementation decisions concerning frontend structure, visual design, component organization, and specific web technologies may be made within the requirements of this specification.

Changes or additions that would affect any of the following should be discussed before implementation:

- scientific interpretation of the results
- ranking calculations or ranking semantics
- mechanistic path selection or interpretation
- disease-gene semantics
- INDRA relation or evidence semantics
- required input or output data structures
- functionality beyond the current development scope

This ensures that implementation decisions remain clearly separated from changes to the underlying scientific methodology or project scope.

11. Implementation Phases

The implementation should proceed in clearly defined phases so that the data-access layer, user-interface components, and mechanistic graph functionality can be developed and validated incrementally.

Each phase should produce a testable result before proceeding to the next stage.

Phase	Main Deliverables
Phase 1 – Data Access and Validation	Implement configurable `DATA_ROOT`; load `disease_manifest.csv`; validate disease-specific result packages; implement loaders for ranking, disease-gene, and mechanistic-path data.
Phase 2 – Application Structure and Disease Selection	Establish the application structure; implement disease search and selection; connect the selected disease to its corresponding INDRA result package.
Phase 3 – Drug Ranking and Selection	Implement ranking-criterion selection; Top 20/50/100/All views; sorting and drug-name search; selection of up to five drugs; complete ranking download.
Phase 4 – Reachable Disease-Gene Selection	Determine reachable disease genes dynamically from the selected drugs and stored mechanistic paths; implement gene search and selection of up to five reachable genes.
Phase 5 – Mechanistic Graph and Evidence Exploration	Retrieve all complete stored paths for the selected drug–gene pairs; construct the union of the matching paths; implement interactive graph visualization and access to relation, evidence, and provenance information.
Phase 6 – Integration and Validation	Validate the complete disease → ranking → drug selection → reachable-gene selection → mechanistic graph workflow across multiple disease result packages; verify consistency with the defined data contracts and reference visualization behavior.

11.1 Phase 1 - Data Access and Validation

Implement the data-access foundation defined in Sections 6–8.

The application should be able to:

- load the disease manifest
- locate disease-specific result packages through the configured data root
- load and validate the required CSV and pickle files
- report missing or invalid required files clearly

11.2 Phase 2 - Application Structure and Disease Selection

Implement the basic application structure and disease-selection workflow.

The selected disease becomes the shared context for the INDRA component and determines which disease-specific result package is used by all subsequent functions.

The application structure should allow the planned PPI and INDRA-PPI comparison components to be added in later development phases without requiring restructuring of the current INDRA component.

11.3 Phase 3 - Drug Ranking and Selection

Implement the Drug Ranking and Exploration functionality defined in Sections 3 and 4.

The user should be able to explore the precomputed ranking using the available ranking criteria, search and sort the results, select up to five drugs, and download the complete ranking.

The selected drugs become the source nodes for subsequent mechanistic exploration.

11.4 Phase 4 - Reachable Disease-Gene Selection

Implement dynamic determination of disease genes reachable from the currently selected drugs.

The available gene set must be derived from the stored mechanistic paths and updated whenever the selected drug set changes.

The user should be able to search the resulting reachable-gene set and select up to five disease genes.

11.5 Phase 5 - Mechanistic Graph and Evidence Exploration

Implement retrieval and visualization of the complete mechanistic paths corresponding to the selected drug–gene pairs.

By default, the graph should represent the union of all matching complete stored paths and preserve their associated relation, evidence, and provenance information.

The implementation should support shared target genes, overlapping paths, and paths originating from multiple selected drugs.

11.6 Phase 6 – Integration and Validation

Validate the complete application workflow across multiple disease result packages.

Validation should confirm that:

- disease selection loads the correct result package

- ranking results correspond to the selected disease
- drug selection produces the correct reachable-gene set
- gene selection retrieves the correct complete mechanistic paths
- graph construction preserves the complete selected paths
- shared targets and overlapping paths are represented correctly
- relation, evidence, and provenance information is preserved
- the implementation conforms to the data contracts and functional requirements defined in this specification

12. Acceptance Criteria

The current INDRA implementation is considered complete when the following functional and technical requirements are satisfied.

12.1 Disease and Data Access

☐ The application loads multiple disease result packages without disease-specific hard-coded application logic.

☐ The disease selector is populated from `disease_manifest.csv`.

☐ A user can search for and select one active disease.

☐ Changing the active disease loads the corresponding disease-specific result package and resets the current drug, gene, and graph selections.

☐ Required result files are accessed through the configurable data root rather than hard-coded absolute server paths.

12.2 Drug Ranking and Selection

☐ The INDRA drug ranking is loaded from the supplied `drug_rankings.csv` without recalculation of the scientific ranking results.

☐ The user can select among the supported precomputed ranking criteria.

☐ The user can view Top 20, Top 50, Top 100, and complete ranking results.

☐ The ranking can be searched by drug name and sorted using the supported ranking information.

☐ A maximum of five drugs can be selected for mechanistic exploration.

☐ The complete disease-specific ranking CSV can be downloaded.

12.3 Reachable Disease-Gene Selection

☐ After drug selection, the available disease genes are dynamically determined from `mechanistic_paths.pkl.gz`.

☐ The gene selector contains only disease genes reachable from at least one currently selected drug.

☐ The reachable-gene list updates when the selected drug set changes.

☐ The reachable-gene list can be searched by gene name.

☐ A maximum of five disease genes can be selected.

12.4 Mechanistic Path Retrieval and Graph Construction

☐ For the selected drugs and disease genes, the application retrieves all matching complete stored mechanistic paths.

☐ By default, the mechanistic graph is constructed from the union of all matching complete stored paths.

☐ Complete paths are not automatically reduced to shortest or representative paths and intermediate nodes are not truncated.

☐ Path length is available as path metadata and may support filtering or interactive exploration without modifying the underlying stored path data.

☐ A disease gene reachable from multiple selected drugs is represented as a shared target rather than as separate biological entities.

☐ Overlapping mechanistic paths and shared graph elements are represented while preserving their association with the originating drugs.

12.5 Mechanistic Information and Evidence

☐ Source drugs, intermediate nodes, and disease-gene targets are visually distinguishable.

☐ The originating drug associated with each mechanistic path can be identified in the graph.

☐ INDRA relation information associated with graph edges is accessible.

☐ Available evidence and provenance metadata associated with the represented relationships are inspectable.

☐ Relevant path-level and graph-level summary information is available to the user.

12.6 Architecture and Scope

☐ The frontend receives browser-compatible data and is not required to read Python pickle files directly.

☐ The application uses the defined backend data-access layer to access result data.

☐ The web application operates on the supplied precomputed results without executing or modifying the scientific analysis pipeline.

☐ The application requires access only to the result data and reference materials required by this specification, not to the scientific pipeline source code.

☐ The architecture allows the planned PPI component and INDRA–PPI comparison functionality to be added in later development phases without requiring changes to the current INDRA data contract or disease-level application model.

13. Provided Development Resources

The following resources will be provided for implementation and testing:

- disease_manifest.csv;
- representative INDRA result packages for multiple diseases;
- for each provided disease:
    - drug_rankings.csv;
    - disease_module_open_targets.csv;
    - mechanistic_paths.pkl.gz;
- indra_mechanistic_graph_reference.py; and
- this development specification.

The provided disease packages should be used to verify that the implementation operates consistently across diseases and does not depend on disease-specific assumptions.

Additional result files or scientific resources will be provided only if they are required by an agreed extension of the current development scope.

14. Future Extension: PPI and INDRA–PPI Comparison

The current development focuses on the INDRA component. A separate specification and data contract will be provided for the planned PPI component.

The application architecture should allow the PPI component to be integrated in a later development phase while preserving the common disease-centered application structure. Where appropriate, the PPI component may use the same disease context and navigation model while maintaining its own component-specific data contract, analysis results, and visualization requirements.

A subsequent **INDRA–PPI Comparison** component is planned to support comparison and integration of results from the two components. Its functionality and data requirements will be defined separately after the PPI component has been specified.

The current INDRA implementation should therefore:

- implement the INDRA functionality defined in this specification completely;
- maintain a shared disease-level application context that can also support future components;
- keep backend routing and data-access conventions extensible;
- avoid assumptions in the application structure that would prevent the addition of PPI-based results; and
- keep INDRA-specific data handling clearly separated so that future PPI integration does not require modification of the current INDRA data contract.

The PPI component and INDRA-PPI Comparison functionality are **not part of the current development scope**.

Appendix A – Initial Developer Checklist

Before and during implementation, verify the following requirements:

☐ Load the available diseases from disease_manifest.csv and use the corresponding disease_id to locate each disease-specific result package.

☐ Validate the presence and readability of the three required disease-specific files: drug_rankings.csv, disease_module_open_targets.csv, and mechanistic_paths.pkl.gz.

☐ Load and deserialize mechanistic_paths.pkl.gz only in the backend. Python pickle data must not be sent to or deserialized by the browser.

☐ Use the precomputed ranking values and scientific scores as provided. Do not recalculate or modify them in the web application.

☐ After drug selection, determine the available disease genes dynamically from the stored mechanistic paths.

☐ By default, retain all complete stored paths matching the selected drug–gene pairs. Do not automatically reduce them to shortest or representative paths or truncate intermediate nodes.

☐ Keep the maximum selection limits at **5 drugs** and **5 disease genes**.

☐ Preserve the association between complete mechanistic paths and their originating drugs, including cases where multiple drugs reach the same disease gene or paths overlap.

☐ Preserve the available INDRA relation, evidence, and provenance information when transforming the stored path data for web visualization.

☐ Use the configurable data root and standardized file structure rather than disease-specific or server-specific hard-coded paths.

☐ Keep the scientific analysis pipeline separate from the web application.

☐ Discuss any proposed change that affects scientific semantics, ranking logic, mechanistic path handling, the defined data contract, or the agreed development scope before implementation.