From Messy Data to Medical Insights: Creating Knowledge Graphs for Drug Repurposing

A Hands-on Tutorial

Matilde Pato[1,2,3,4], Ana Carolina Pereira[1] & Nuno Datia[1,4]

[1] ISEL, [2] IBEB, [3] LASIGE, [4] NOVALINCS, Lisbon, Portugal

in **womENcourage2025**, Brașov, Romania

September 18, 2025

Tutorial Schedule (90 minutes)

00:00-00:15 (15 min) - Introduction to KGs for Drug Repurposing

- Challenges in **biomedical data** integration
- Advantages of **graph-based representations**
- Overview of public pharmaceutical data sources

Tutorial Schedule (90 minutes)

00:00-00:15 (15 min) - Introduction to KGs for Drug Repurposing

- Challenges in **biomedical data** integration
- Advantages of **graph-based representations**
- Overview of public pharmaceutical data sources

00:15-00:45 (30 min) - Biomedical Data Wrangling Techniques

- **NER** for drugs and diseases
- Relationship extraction from **unstructured text**
- Demonstration of the **MedJsonify** methodology

Tutorial Schedule (90 minutes)

00:45-01:15 (30 min) - Hands-on Session

- Loading prepared datasets into **Neo4j**
- Writing basic **Cypher** queries to explore relationships
- **Visualization techniques** and identifying repurposing candidates

Tutorial Schedule (90 minutes)

00:45-01:15 (30 min) - Hands-on Session

- Loading prepared datasets into **Neo4j**
- Writing basic **Cypher** queries to explore relationships
- **Visualization techniques** and identifying repurposing candidates

01:15-01:30 (15 min) - Q&A and Resources

- Access to code templates and datasets
- Recommended tools and community resources

Welcome & Tutorial Objectives

What You'll Learn Today

- Fundamentals of **knowledge graph** construction for pharmaceutical data
- Practical biomedical **data wrangling** techniques
- **Hands-on** experience with Neo4j visualization and querying
- How KGs can **transform** healthcare education
- **Reusable** templates for your own work

Prerequisites Check

- Python programming experience? ✔
- Neo4j Desktop installed? ✔
- Sample datasets downloaded? ✔

The Drug Repurposing Challenge

Traditional Approach

- Linear drug development
- 10-15 years, $1B+ cost
- High failure rates
- Siloed data sources

The Drug Repurposing Challenge

Traditional Approach

- Linear drug development
- 10-15 years, $1B+ cost
- High failure rates
- Siloed data sources

Drug Repurposing

- Find new uses for existing drugs
- Faster, cheaper development
- Lower risk profile
- **Requires integrated data!**

The Drug Repurposing Challenge

Traditional Approach

- Linear drug development
- 10-15 years, $1B+ cost
- High failure rates
- Siloed data sources

Drug Repurposing

- Find new uses for existing drugs
- Faster, cheaper development
- Lower risk profile
- **Requires integrated data!**

Success Stories

Viagra (angina $\rightarrow$ erectile dysfunction), **Thalidomide** (sedative $\rightarrow$ cancer), **Metformin** (diabetes $\rightarrow$ aging research)

Why Knowledge Graphs?

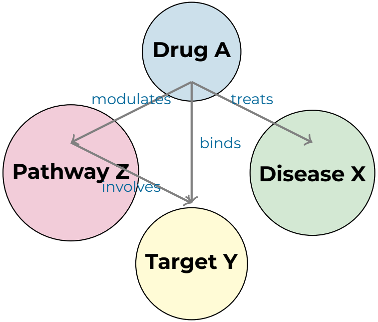
Traditional Data

Drug A | Disease X

Drug B | Disease y

Drug C | Disease Z

Transform to



KG Advantages

- **Relationships**: Explicit representation of **drug-disease-target** connections
- **Flexibility**: Easy to add new data types and relationships
- **Reasoning**: Support for **inference** and **pattern** discovery
- **Visualization**: Intuitive exploration of complex relationships

Biomedical Data Landscape

Public Data Sources

- **DailyMed**: FDA drug labels
- **Orange Book**: Generic equivalents
- **Purple Book**: Biologics
- **DrugBank**: Comprehensive drug data

Biomedical Data Landscape

Public Data Sources

- **DailyMed**: FDA drug labels
- **Orange Book**: Generic equivalents
- **Purple Book**: Biologics
- **DrugBank**: Comprehensive drug data

Ontologies

- **ChEBI**: Chemical entities
- **Disease Ontology (DO)**: Disease classification
- **Orphanet**: Rare diseases

Biomedical Data Landscape

Public Data Sources

- **DailyMed**: FDA drug labels
- **Orange Book**: Generic equivalents
- **Purple Book**: Biologics
- **DrugBank**: Comprehensive drug data

Ontologies

- **ChEBI**: Chemical entities
- **Disease Ontology (DO)**: Disease classification
- **Orphanet**: Rare diseases

Major Challenges

- ⚠ Inconsistent terminology
- 🔀 Heterogeneous formats
- Entity resolution across sources
- ? Missing relationships
- Scale and complexity

The MedJsonify Pipeline

Raw Data → **Clean & Normalize** → **NER + NEL** → **Knowledge Graph**

Text cleaning,
Format standardization

Entity recognition,
Ontology linking

Key Components

- **Text Preprocessing**: Handle Unicode, normalize terminology
- **Named Entity Recognition (NER)**: Identify drugs, diseases, targets
- **Named Entity Linking (NEL)**: Map entities to ontology IDs
- **Relationship Extraction**: Infer connections from text

NER + NEL Example

Sample Text

"Lyrica (pregabalin) is indicated for diabetic peripheral neuropathy and fibromyalgia."

```
# Extracted entities with ontology links
entities = {
    "drugs": [
        {"text": "Lyrica", "chebi_id": "CHEBI:64356", "name": "pregabalin"},
        {"text": "pregabalin", "chebi_id": "CHEBI:64356"}
    ],
    "diseases": [
        {"text": "diabetic peripheral neuropathy",
         "doid_id": "DOID:574", "name": "peripheral neuropathy"},
        {"text": "fibromyalgia",
         "doid_id": "DOID:631", "name": "fibromyalgia"}
    ]
    }
```

NER + NEL Example (cont.)

Pattern Matching Strategies

- Dictionary lookup with fuzzy matching
- Regular expression patterns for medical terms
- Machine learning models (when available)
- Ontology-based expansion using synonyms

Hands-on Session Overview

What We'll Build

A drug-disease knowledge graph using real pharmaceutical data

Tools We'll Use

- **Python**: Data processing
- **Neo4j**: Graph database
- **Cypher**: Query language
- **Prepared datasets**: DailyMed + ontologies

Steps

1. Load preprocessed data
2. Create nodes and relationships
3. Write basic Cypher queries
4. Visualize drug-disease networks
5. Identify repurposing candidates

Check Your Setup

Open Neo4j Desktop and verify Python environment

Step 1: Data Loading

```python
import json
import neo4j
from neo4j import GraphDatabase

# Load processed pharmaceutical data
with open('lyrica_annotated.json', 'r') as f:
    drug_data = json.load(f)

# Neo4j connection
driver = GraphDatabase.driver("bolt://localhost:7687",
                              auth=("neo4j", "password"))

...
```

Step 2: Creating Disease Relationships

```python
def create_indications(tx, drug_name, doid_entities):
    for entity in doid_entities:
        # Create disease node
        create_disease_query = """
        MERGE (dis:Disease {
            doid_id: $doid_id,
            name: $name
        })
        """
        tx.run(create_disease_query,
               doid_id=entity['doid_id'],
               name=entity['text'])

        # Create relationship
        relation_query = """
        MATCH (d:Drug {name: $drug_name})
        MATCH (dis:Disease {doid_id: $doid_id})
        CREATE (d)-[:TREATS]->(dis)
        """
        tx.run(relation_query,
               drug_name=drug_name,
               doid_id=entity['doid_id'])

...
```

Step 3: Basic Cypher Queries

Find all diseases treated by Lyrica

```
MATCH (d:Drug {name: "Lyrica"})-[:TREATS]->(dis:Disease)
RETURN d.name, dis.name, dis.doid_id
```

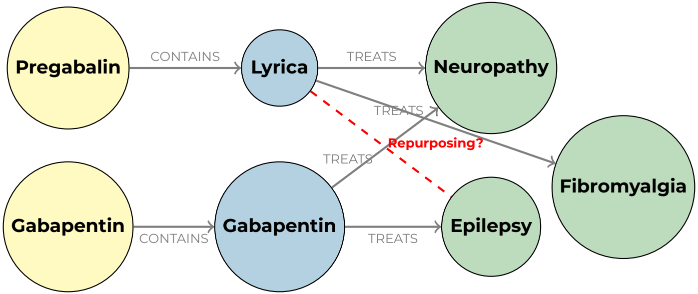
Find drugs that treat neuropathy

```
MATCH (d:Drug)-[:TREATS]->(dis:Disease)
WHERE dis.name CONTAINS "neuropathy"
RETURN d.name, d.organization, dis.name
```

Identify potential repurposing candidates

```
MATCH (d1:Drug)-[:TREATS]->(dis:Disease)<-[:TREATS]-(d2:Drug)
WHERE d1 <> d2
RETURN d1.name, d2.name, dis.name as shared_indication
LIMIT 10
```

Step 4: Graph Visualization



Visualization Features in Neo4j Browser

- **Interactive exploration**: Click to expand relationships
- **Styling**: Custom colors for different node types
- **Filtering**: Focus on specific drug classes or diseases
- **Export**: Save visualizations for presentations

Step 5: Advanced Analysis

Repurposing Signals

- Drugs treating related diseases
- Shared molecular targets
- Similar metabolic pathways
- Patient population overlap

Pattern Discovery

- Drugs with similar indication profiles
- Disease clusters based on treatments
- Mechanism-based groupings
- Adverse event patterns

Validation Steps

- Literature review
- Clinical trial databases
- Regulatory approval history
- Safety profile comparison

Example Insights

Anticonvulsants often treat both epilepsy and neuropathic pain → shared mechanisms

Educational Impact & Applications

Traditional Learning

- Memorization of drug lists
- Linear textbook chapters
- Isolated disease studies
- Limited connection-making

Learning Challenges

- Information overload
- Fragmented knowledge
- ? Difficulty seeing patterns
- Outdated resources

KG-Enhanced Learning

- Visual relationship exploration
- Discovery-based learning
- Real-time data integration
- Collaborative investigation

Success Stories

- Medical schools using KGs for pharmacology
- Pharmacy programs for drug interaction studies
- Clinical training for evidence-based prescribing

Real-world Applications

Clinical Decision Support

- Drug interaction checking
- Alternative therapy suggestions
- Contraindication alerts
- Dosing recommendations

Research Applications

- Hypothesis generation for drug repurposing
- Target identification and validation
- Biomarker discovery
- Clinical trial design

Real-world Applications

Industry Impact

- Pharmaceutical R&D optimization
- Regulatory submission support
- Post-market surveillance
- Competitive intelligence

Access to Code Templates and Datasets

What You Get

- Complete MedJsonify codebase
- Sample processed datasets
- Neo4j database templates
- Cypher query examples
- Visualization scripts

Recommended Tools

- **Neo4j Desktop**: Graph database
- **Python libraries**: pandas, requests, neo4j-driver
- **Jupyter**: Interactive development
- **Gephi**: Advanced graph visualization

Further Reading

- Graph Databases (O'Reilly)
- Neo4j documentation
- BioCypher for biomedical KGs
- OpenTargets platform

Community

- Neo4j Community Forum
- FAIR Data initiatives
- Biomedical informatics conferences
- GitHub repositories

Stay Connected

Contact us for implementation support and collaboration opportunities

Questions & Discussion

Discussion Topics

- Challenges in your own biomedical data integration projects
- Potential applications in your research/teaching
- Technical implementation questions
- Collaboration opportunities

Thank you for participating!

Code: https://matpato.github.io/ReDrug-KG/
Contact: matilde.pato@isel.pt