ROBOKOP and the Biomedical Data Translator: Efficiently Leveraging a Distributed Data Ecosystem

Jim Balhoff¹, Chris Bizon¹, Steve Cox¹, Karamarie Fecho¹,², Yaphet Kebede¹, Kenneth Morton³, Alexander Tropsha¹,⁴, Patrick Wang³

¹Renaissance Computing Institute (RENCI), University of North Carolina at Chapel Hill
²Copperline Professional Solutions
³CoVar Applied Technologies
⁴School of Pharmacy, University of North Carolina at Chapel Hill
NIH Mission...*is to seek fundamental knowledge about the nature and behavior of living systems and the application of that knowledge to enhance health, lengthen life, and reduce illness and disability.*
NCATS Mission...is to catalyze the generation of innovative methods and technologies that will enhance the development, testing and implementation of diagnostics and therapeutics across a wide range of human diseases and conditions.
NCATS Biomedical Data Translator Program

- Overarching goal is to promote "serendipitous" discovery
- Primary goal is to build infrastructure to support and facilitate data-driven translational research on a large scale
- Essential aim is to link as many datasets as possible with one another and allow them to be cross-queried and reasoned over by translational researchers...
  ...at speed, with minimal barriers, scalability, accuracy
- Fundamental tenet of the program is open data and open collaborative team science
NCATS Biomedical Data Translator Program

‘Other Transaction Award (OTA)’
NCATS Biomedical Data Translator Program
NCATS Biomedical Data Translator Program
The Biomedical Data Translator Consortium:
11 Teams, 28 institutions, ~200 team members
Phase I: 09/24/2016 - 12/31/2019
The Biomedical Data Translator Consortium: Phase II: kick-off 01/27/2020
"Two hundred years ago, chemists created a comprehensive enumeration of the elements and systematic relationships among them. We envision the Translator doing the same for translational science."

– Christopher P. Austin, MD, director of NCATS, with Christine M. Colvis, PhD, Noel T. Southall, PhD

Image created by Julie McMurry, with input from the Biomedical Data Translator Consortium
Prototype Translator System Architecture
ROBOKOP (Reasoning Over Biomedical Objects linked in Knowledge-Oriented Pathways)

- An open knowledge graph (KG)-based question-answering system
- Can be used to derive mechanistic insights into real-world observations

**what biological pathways might explain the association between exposure to carbon monoxide and multiple sclerosis?**

**how does hair dye influence susceptibility to breast cancer?**

**what genes might explain the effectiveness of isopropyl alcohol in the treatment of cyclic vomiting syndrome?**
Demo

Ask a Question

Question Title

What genetic conditions may provide protection from Ebola?

Machine Question Editor - Question Graph

0: Ebola hemorrhagic fever → 1: Gene → 2: Genetic Condition
Services

- “manager”
  - manages users
  - stores queries and results
  - presents custom web GUI
    - building and running queries
    - exploring results

- “ranker” / “messenger”
  - receives queries from public API, in JSON format
  - transpiles into Cypher query
  - collates, scores, and returns results

- “builder” / “interfaces”
  - consults registry of (third-party) APIs
  - identifies sequences of queries that could combine to produce answers
  - iteratively queries to gather relevant data
  - stores data in Neo4j database “robokopdb”

- “robokopDB”
  - publicly available Neo4j browser
## Builder/Interfaces

**ROBOKOP Builder**

A web application that connects questions with biomedical knowledge services.

**Terms of Service**

**Send email to the developer**

---

### util

**GET** `/builder/api/annotate/{node_id}/{node_type}`

Returns annotation for a node.

**GET** `/builder/api/concepts`

Get known biomedical concepts.

**GET** `/builder/api/connections`

Get a simplified list of all edges in the type graph.

**GET** `/builder/api/node_properties`

Get a JSON object of properties for each node type.

**POST** `/builder/api/node_properties`

Force update of node-type property list from neo4j database.

**POST** `/builder/api/normalize`

Adds synonyms to node and normalize edge db source for a json blob of answer knowledge graph.

**GET** `/builder/api/operations`

Get a JSON list of all edges in the type graph.
Ranker

ROBOKOP Ranker 0.0.1

An API for answering biomedical questions

Terms of service
the developer - Website
Send email to the developer

answer

**POST** /ranker/api/  Get answers to a question

**POST** /ranker/api/query/  Get answers to a question

util

**POST** /ranker/api/count_connections/  Count connections between biomedical entities.

**POST** /ranker/api/count_predicates/  Count predicates between two biomedical entities.

**POST** /ranker/api/entity_lookup/  Look up biomedical entities by name.

**POST** /ranker/api/entity_lookup/{node_type}/  Look up biomedical entities by name.

https://robokop.renci.org/ranker/apidocs/
ROBOKOP KG

- > 6M nodes and 140M edges
- ~30 underlying biomedical data sources and bio-ontologies (e.g., KEGG, Monarch services, DrugCentral, Pharos, Comparative Toxicogenomics Database, Monarch Disease Ontology, Gene Ontology, ChEBI, Human Phenotype Ontology)
- Extensible to non-biomedical domains and knowledge sources
ROBOKOP Tooling

- **Supervisor**: process control system
  - e.g. Flask process, Celery process, …

- **Celery**: distributed task queue
  - master Celery (Python) process
  - multiple Python worker processes
  - jobs distributed to workers via message broker (RabbitMQ)
  - results stored by workers in results database (Redis)

- **NGINX, Gunicorn, Flask**: web server
  - NGINX: reverse proxy
  - Gunicorn: WSGI HTTP server
    - multiple worker processes
  - Flask: web server framework

- **Redis, Neo4j, PostgreSQL**: caching
  - Third-party API calls (Redis)
  - Biomedical data (Neo4j)
  - Entire questions/answers (PostgreSQL)

- **SQLAlchemy, Postgraphile + GraphQL**: interface with PostgreSQL
Containers

- manager
  - Python/Flask
- ranker/messenger
  - Python/Flask
- builder/interfaces
  - Python/Flask
- robokopDB
  - Neo4j
- reverse proxy
  - NGINX
- build cache
  - Redis
- manager store
  - PostgreSQL
- Celery results
  - Redis
- message broker
  - RabbitMQ

robokop.renci.org
ROBOKOP Containers

- manager
- manager store (PostgreSQL)
- builder
- Celery results (Redis)
- message broker (RabbitMQ)
- ranker
- build cache (Redis)
- robokopDB (Neo4j)
ROBOKOP Containers

Diagram showing the relationships between various components:
- builder
- manager
- manager store (PostgreSQL)
- Celery results (Redis)
- message broker (RabbitMQ)
- ranker
- build cache (Redis)
- robokopDB (Neo4j)

robokop.renci.org
ROBOKOP Containers

- **builder**
  - build cache (Redis)
  - query to build
- **message broker (RabbitMQ)**
- **Celery results (Redis)**
- **manager**
  - manager store (PostgreSQL)
- **ranker**
- **robockopDB (Neo4j)**
Shortcomings

- Brittle. If any of the following gets overworked, ROBOKOP becomes unusable:
  - robokopDB (Neo4j) - heap memory, disk space, ???
  - manager (Python) - SQLAlchemy, Celery workers
  - ranker (Python) - Celery workers
  - message broker (RabbitMQ) - this has not happened yet!
  - caches (Redis) - this has not happened yet!
  - manager store (PostgreSQL) - this has not happened yet!

- Opaque. Logging multiple processes, multiple containers is a mess
  - permissions for mapped Docker volumes on Linux are tricky
Future Work: ARAGORN

- Translator is moving from the 3-year “feasibility” phase into the 5-year “development” phase
- Greater focus on distributed system - want to avoid monolithic KG
- Want end-user interface to be responsive: seconds, not hours...
- No need to return all answers quickly, just the best ones...

- Autonomous Relay Agent for Generation of Ranked Networks (ARAGORN)
  - a.k.a Strider
  - solution: asynchronous answering
ARAGORN

• “builder” and “ranker” get combined into a single component that builds and ranks iteratively to generate the best results first
• Plus novel algorithms for augmenting queries and “coalescing” results

• ROBOKOP was made efficient with caching and multiple processes
• ARAGORN will be efficient thanks to prioritization and asynchrony
  ○ We are largely I/O-bound, communicating with “third-party” Translator services
ARAGORN Containers

- Strider
- Temporary storage (Neo4j)
- Temporary storage (Redis)
- Message broker (RabbitMQ)
ARAGORN Tooling

- WSGI → ASGI
- Gunicorn → uvicorn
- Flask → Sanic/Starlette/FastAPI
  - asyncio, aioredis, aiormq, httpx, uvloop

- RabbitMQ + Redis for coordinating “threads”
- Neo4j for storing, querying biomedical data
Deployment

- ROBOKOP is essentially stateful
  - Relies on a single monolithic KG
  - Relies on long-term storage for questions/answers in PostgreSQL
  - Each component scales independently

- ARAGORN is at least *less* stateful
  - Relies on short-term storage using job-specific caches
  - Easier to scale horizontally - can replicate entire ARAGORN service and parallelize behind load balancer
    - e.g. Kubernetes
# Funding

Support was provided by the National Center for Advancing Translational Sciences, National Institutes of Health through the Biomedical Data Translator program (awards OT3TR002020 and OT2TR002514) and the Clinical and Translational Sciences Award program (award UL1TR002489).

## Gamma Team

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Balhoff</td>
</tr>
<tr>
<td>Mohit Bansal†</td>
</tr>
<tr>
<td>Chris Bizon (co-PI)</td>
</tr>
<tr>
<td>Richard Bruskiewich†</td>
</tr>
<tr>
<td>Stephen Capuzzi</td>
</tr>
<tr>
<td>Steven Cox</td>
</tr>
<tr>
<td>Miles Crosskey†</td>
</tr>
<tr>
<td>Karamarie Fecho</td>
</tr>
<tr>
<td>Sudeep Mandal</td>
</tr>
<tr>
<td>Kenneth Morton</td>
</tr>
<tr>
<td>Eugene Muratov</td>
</tr>
<tr>
<td>Kent Shefchek</td>
</tr>
<tr>
<td>Andrew Thieme</td>
</tr>
<tr>
<td>Alexander Tropsha (PI)</td>
</tr>
<tr>
<td>Patrick Wang</td>
</tr>
</tbody>
</table>

## Green Team

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanley Ahalt (PI)</td>
</tr>
<tr>
<td>Sarav Arunachalam</td>
</tr>
<tr>
<td>Stephen Appold</td>
</tr>
<tr>
<td>Jim Balhoff</td>
</tr>
<tr>
<td>Kira Bradford†</td>
</tr>
<tr>
<td>James Champion</td>
</tr>
<tr>
<td>Steve Cox</td>
</tr>
<tr>
<td>Karamarie Fecho</td>
</tr>
<tr>
<td>Karl Gustafson</td>
</tr>
<tr>
<td>Adel Hanna†</td>
</tr>
<tr>
<td>Ray Idaszak†</td>
</tr>
<tr>
<td>Ann Moss Joyner†</td>
</tr>
<tr>
<td>Ashok Krishnamurthy</td>
</tr>
<tr>
<td>Benjamin Marsh†</td>
</tr>
<tr>
<td>Asia Mieczkowska†</td>
</tr>
<tr>
<td>Allan Parnell†</td>
</tr>
<tr>
<td>Dave Peden</td>
</tr>
<tr>
<td>Emily Pfaff</td>
</tr>
<tr>
<td>Kimberly Robasky†</td>
</tr>
<tr>
<td>Charles Schmitt</td>
</tr>
<tr>
<td>Michael Stealey†</td>
</tr>
<tr>
<td>Lisa Stillwell</td>
</tr>
<tr>
<td>Alexander Tropsha (co-PI)</td>
</tr>
<tr>
<td>Alejandro (Alex) Valencia</td>
</tr>
<tr>
<td>Hao Xu</td>
</tr>
<tr>
<td>Dongmei Yang†</td>
</tr>
<tr>
<td>Hong Yi†</td>
</tr>
</tbody>
</table>

†Former team member
References and Resources

Green/Gamma Translator Documentation Website
- ROBOKOP page
- ROBOKOP APIs


Contact information:
- kfecho@copperlineprofessionalsolutions.com
- patrick@covar.com