neo4jsbml

Release 1.0.1

BRS team

Mar 18, 2024

CONTENTS

CHAPTER

ONE

CONTENTS

1.1 Usage

1.1.1 Installation

With conda:

```
$ conda install -c conda-forge neo4jsbml
```

1.1.2 Principles

Import data from SBML into the Neo4j database is conducted in several steps:

1. Define your schema with Arrows and download the schema at the JSON format
2. Import your data into Neo4j

1.1.3 Define your schema with Arrows

Create the schema First of all, the users create a schema defining which entities will be selected from the SBML model with arrows.

Rules:

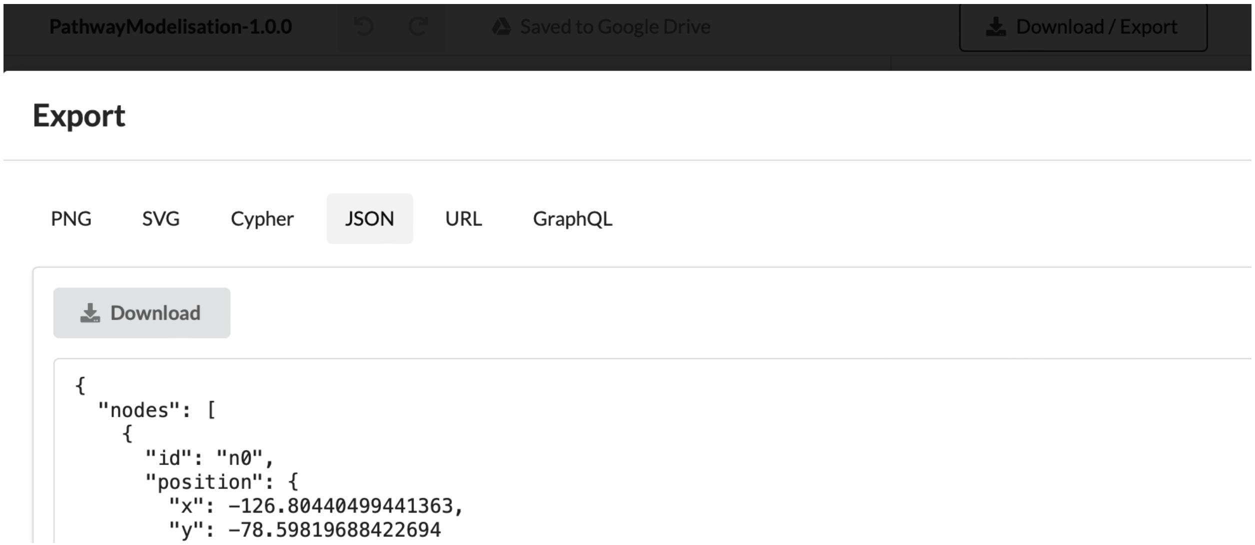
- Nodes are labelled based on SBML object name as defined in the SBML specification
- Properties are labelled based on SBML object properties as defined the SBML specification

Download the schema

1.1.4 Configure access to Neo4j

The connection to the Neo4j database needs to have several parameters defined:

- protocol: `neo4j`, `neo4j+s`, `neo4j+ssc`, `bolt`, `bolt+s`, `bolt+ssc` (default: `neo4j`)
- url (default: `localhost`)
- user name (default: `neo4j`)
- port (default: `None`)
- password (default: `None`)



```
{
  "nodes": [
    {
      "id": "n0",
      "position": {
        "x": -126.80440499441363,
        "y": -78.59819688422694
```

- database name (default: `neo4j`)

The user has two options: passing arguments individually by the command line or by an `ini` file with this structure:

Command line

`--input-protocol-str`
Protocol to communicate with the database

`--input-url-str`
Url of Neo4j

`--input-port-int`
Port number tot communicate with the database

`--input-user-str`
Username to log with the database

`--input-password-txt`
Password to log with the database

`--input-database-str`
Database name

Configuration file Several options, either

`--input-config-ini`
An `ini` file containing all these informations above

```
[connection]
protocol = neo4j
url = localhost
port = 7687

[database]
user = neo4j
password = mypassword
name = neo4j
```

Note: For safety, passing a password through the command line must be given by a file. No extra character must be in the file, otherwise it would be consider as the password.

or either

--input-auradb-txt
A txt file provided by AuraDB

```
# one comment
NEO4J_URI=neo4j+s://test.neo4j.io
NEO4J_USERNAME=neo4j
NEO4J_PASSWORD=thepassword
# second comment
AURA_INSTANCEID=422DEf4
AURA_INSTANCENAME=Instance01
```

1.1.5 Import your data into Neo4j

Command line

To import your data with neo4jsbml into Neo4j, you will need: 1. the database parameters 2. the SBML file, e.g. the model 3. the JSON file downloaded from arrows

```
$ neo4jsbml sbml-to-neo4j
    <database parameters>

    --input-model-sbml <file> \
    --input-arrows-json <file>
```

Note: If you have multiple model in the database, pass a tag to identify the model loaded into the database if you want to avoid collision with the argument --parameter-tag-property-str

Plugins compatibility

Package	Compatibility
Arrays	No
Hierarchical Model Composition	No
Distributions	No
Dynamic Structures	No
Flux Balance Constraints	Yes
Groups	Yes
Layout	Yes
Extended MathML	No
Multistate, Multicomponent and Multicompartment Species	No
Qualitative models	Yes
Rendering	No
Spatial Processes	No

1.1.6 Export your data into Neo4j

Command line

To export your data with `neo4jsbml` into Neo4j, you will need: 1. the database parameters 1. the `JSON` file downloaded from arrows

```
$ neo4jsbml sbml-from-neo4j
    <database parameters>

    --input-arrows-json <file> \
    --output-model-sbml <file>
```

Note: neo4jsbml tries to map your arrows schema to the structure of the SBML document. It's better to keep the SBML structure in Neo4j and in the Arrows schema, notably to keep a Model entity.

Plugins compatibility

Package	Compatibility
Arrays	No
Hierarchical Model Composition	No
Distributions	No
Dynamic Structures	No
Flux Balance Constraints	No
Groups	No
Layout	No
Extended MathML	No
Multistate, Multicomponent and Multicompartment Species	No
Qualitative models	No
Rendering	No
Spatial Processes	No

With python

```python
from neo4jsbml import arrows, connect, sbml

# Either you have a configuration file or overwrite individually
path_config = None
con = connect.Connect.from_config(path=path_config)
# Or
path_password = None
con = connect.Connect(
    protocol="neo4j",
    url="localhost",
    port="7687",
    user="neo4j"
    database="neo4j",
    password_path=path_password,
```

(continues on next page)

(continued from previous page)

```
)

# Load model - Define a tag here if needed
tag = None
path_model = ""
sbm = sbml.Sbml.from_sbml(path=path_model, tag=tag)

# Load modelisation
path_modelisation = ""
arr = arrows.Arrows.from_json(path=path_modelisation)

# Mapping
nod = sbm.format_nodes(nodes=arr.nodes)
rel = sbm.format_relationships(relationships=arr.relationships)

# Import into neo4j
con.create_nodes(nodes=nod)
con.create_relationships(relationships=rel)
```

1.2 Real-life applications

1.2.1 View shared metabolite

View shared metabolite, fumarate, between two models: *iAF1260* and *iML1515* available from BiGG

Arrows modelisation

`(JSON schema)`

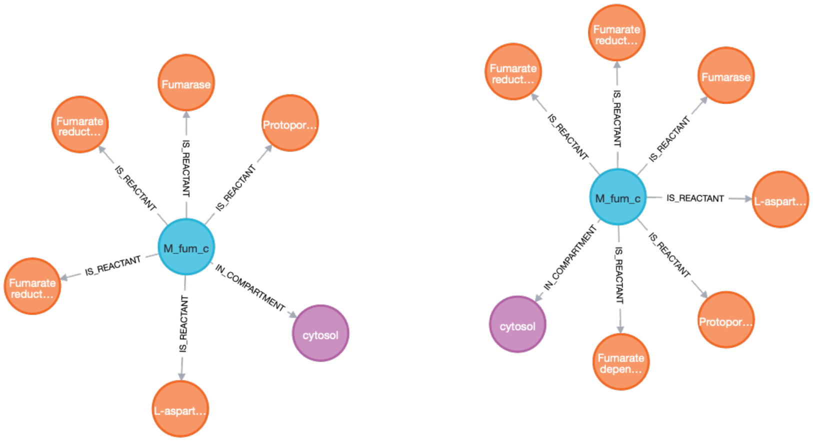
Neo4jSbml

```
$ neo4jsbml sbml-to-neo4j \
    --input-config-ini tests/dataset/database/localhost.ini \
    --input-arrows-json tests/dataset/modelisation/PathwayModelisation-2.0.2.json \
    --input-model-sbml tests/dataset/model/iML1515.xml.gz \
    --parameter-tag-property-str iML1515
$ neo4jsbml sbml-to-neo4j \
    --input-config-ini tests/datasets/database/localhost.ini \
    --input-arrows-json tests/dataset/modelisation/PathwayModelisation-2.0.2.json \
    --input-model-sbml tests/dataset/model/iAF1260.xml.gz \
    --paramter-tag-property-str iAF1260
```

Query Cypher

```
MATCH p=((c:Compartment)<--(n:Species {id: "M_fum_c"})-->(r:Reaction)) RETURN p
```

Neo4j



1.2.2 View precursors

Visualization of the Phosphoenolpyruvate and its precursors at exactly two reactions in the *e_coli_core* available from BiGG

Arrows modelisation

```
(JSON schema)
```

Neo4jSbml

```
$ neo4jsbml sbml-to-neo4j \
    --input-config-ini tests/dataset/database/localhost.ini \
    --input-arrows-json tests/dataset/modelisation/PathwayModelisation-2.0.2.json \
    --input-model-sbml tests/dataset/model/e_coli_core.xml.gz
```

Query Cypher

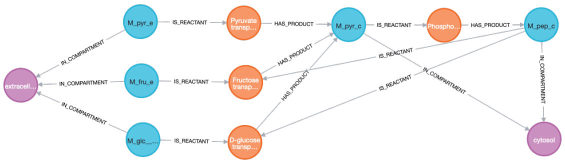
Flag the metabolite nodes implied in more than 10 reactions.

```
MATCH (s:Species)
WITH s, size ([p=(s)-[:IS_SUBSTRATE]->() | p]) as sz order by sz desc
where sz >10
SET s:Hub
```

Select the nodes to highlight them

```
MATCH path=(:Compartment)<-[:IN_COMPARTMENT]-(s1:Species {id: "M_pep_c"})<-[:HAS_
↪PRODUCT|IS_SUBSTRATE*2]-(s2:Species)<-[:HAS_PRODUCT|IS_SUBSTRATE*2]-(s3:Species)-[:IN_
↪COMPARTMENT]->(:Compartment)
WHERE NOT "Hub" IN apoc.coll.flatten([n in nodes(path) | labels(n)])
    AND (s3)-[:IN_COMPARTMENT]->(:Compartment {id: "e"})
    AND NOT (s2)-[:IN_COMPARTMENT]->(:Compartment {id: "e"})
RETURN path
```

Neo4j



1.2.3 Enumerate pathway

Visualization of the pathways where the formate and the Acetaldehyde where involved in the *e_coli_core* available from BiGG

Arrows modelisation

`(JSON schema)`

Neo4jSbml

```
$ neo4jsbml sbml-to-neo4j \
    --input-config-ini tests/dataset/database/localhost.ini \
    --input-arrows-json tests/dataset/modelisation/EnumeratePathway.json \
    --input-model-sbml tests/dataset/model/e_coli_core.xml.gz
```

Graph Refactoring

Add the ID of the Species nodes into the SpeciesReference.

```
MATCH (n:SpeciesReference)-[:HAS_SPECIES]->(s:Species) SET n.id = s.id
```

Remove the Species nodes.

```
MATCH (n:Species) DETACH DELETE n
```

Rename the nodes SpeciesReference to Species

```
MATCH (n:SpeciesReference) REMOVE n:SpeciesReference SET n:Species
```

Remove dense nodes

```
MATCH (s:Species)
WITH s, size ([p=(s)-[:HAS_REACTANT]->() | p]) as sz order by sz desc
WHERE sz > 4
SET s.hub = true

MATCH (s:Species)
WITH s, size ([p=(s)<-[:HAS_PRODUCT]-() | p]) as sz order by sz desc
WHERE sz > 4
SET s.hub = true

MATCH (s:Species) WHERE s.hub DETACH DELETE s
```

Download the plugin Neo4jEfmtool `(JAR file)` and move it to the plugin directory of your instance Neo4j. Restart your database then call the procedure:

```
CALL brsynth.enumeratePathway(["M_for_c", "M_acald_c"], "ep")
```

Neo4j

View the first pathway

```
MATCH (n) WHERE 1 IN n.ep RETURN n
```

View the second pathway

```
MATCH (n) WHERE 2 IN n.ep RETURN n
```

1.3 Example

1.3.1 SBML specification - L3V2

Some examples of Arrow's schemas and Neo4j views coming from the (`SBML specification`) chapter 7.

1 - A simple example application of SBML

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

2 - A simple example using the conversionFactor attribute

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

3 - An alternative formulation of the conversionFactor example

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

4 - Example of a discrete version of a simple dimerization reaction

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

5 - Example involving assignment rules

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

6 - Example involving algebraic rules

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

7 - Example with combinations of boundaryCondition and constant values on Species with RateRule objects

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

8 - Example of translation from a multi-compartmental model to ODEs

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

9 - Example involving function definitions

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

10 - Example involving delay functions

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

11 - Example involving events

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

12 - Example involving two-dimensional compartments

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

13 - Example of a reaction located at a membrane

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

14 - Example using an event with a non-persistent trigger and a delay

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

1.3.2 SBML plugins

Some plugins can be used by Neo4jSbml: * Flux Balance Constraints (`specification`) * Groups (`specification`) * Layout (`specification`) * Qualitative Models (`specification`)

1 - Flux Balance Constraints

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

2 - Groups

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

3 - Layout

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

4 - Qualitative Models

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

1.3.3 SBML specification - L2V5

Some examples of Arrow's schemas and Neo4j views coming from the `(SBML specification)` chapter 7.

1 - A simple example application of SBML

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

2 - Example involving units

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

3 - Example of a discrete version of a simple dimerization reaction

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

4 - Example involving assignment rules

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

5 - Example involving algebraic rules

```
Arrow schema (JSON)
Model (XML.zip)
```

Arrows

Neo4j

6 - Example with combinations of boundaryCondition and constant values on Species with RateRule objects

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

7 - Example of translation from a multi-compartmental model to ODEs

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

8 - Example involving function definitions

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

9 - Example involving delay functions

`Arrow schema (JSON)`
`Model (XML.zip)`

Arrows

Neo4j

10 - Example involving events

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

11 - Example involving two-dimensional compartments

Arrow schema (JSON)
Model (XML.zip)

Arrows

Neo4j

1.4 Code of Conduct

- Be friendly and patient
- Be respectful
- Be welcoming
- Be considerate
- Focus on what's best for the community
- Gracefully accept constructive criticism
- Show courtesy and respect towards other community members
- Be respectful of different viewpoints and experiences
- Use welcoming and inclusive language

1.4.1 Reporting an issue

If you see a violation of the code of conduct, please report it to anyone belonging to the [BioRetroSynth organization](https://github.com/orgs/brsynth/people). If the person who is harassing you belongs to the team, he will recuse himself from handling your incident. We will respond as promptly as we can. To protect volunteers from abuse and burnout, we reserve the right to reject any report we believe to have been made with the deliberate intention of deceiving or misleading the committee.

1.4.2 Enforcement

Upon receiving a report, the Maintainer core team will meet privately to discuss the matter and determine the best course of action.

Possible responses may include:

- taking no further action (if we determine no violation occurred)
- a private reprimand from the core members to the individual(s) involved. In this case, a core member will deliver that reprimand via email, copying the rest of the core.
- a public reprimand. In this case a core member will deliver the reprimand via the same channel as the violation occurred (e.g., via comment on PR or issue)
- an imposed vacation (e.g., asking the individual to refrain from participating for a week). A core member will deliver a request for the individual to take such a "vacation" voluntarily, but if they don't disagree then a temporary ban may be imposed.
- a permanent or temporary ban from the repository

Once a resolution is agreed upon, but before it is enacted, the core will contact the original reporter and discuss the resolution and will note any feedback for the record, but is not obligated to act on such feedback.

The core will never publicly discuss the issue; any public statements will be made by core members.

If the incident involves a core member, the core member must recuse themselves from the response process.

This policy was modified from django project

1.5 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

1.5.1 Ways you can help

Report Bugs

Report bugs at https://github.com/brsynth/neo4jsbml/issues.

If you are reporting a bug, please include:

- Your operating system name and version, versions of Neo4j and SBML specification.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” is open to whoever wants to implement it.

Write Documentation

Same as Implement Features

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/brsynth/neo4jsbml/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- This will hopefully become a community-driven project and contributions are welcome :)

1.5.2 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring.
2. Passing pytest

   ```
   $ python -m pytest
   ```

3. Passing super-linter

This policy was modified from planemo project

1.6 Citation

1.6.1 How to cite ?

```
@article{gricourtNeo4jsbmlImportSystems2024,
  title = {Neo4jsbml: Import Systems Biology Markup Language Data into the Graph␣
↪Database {{Neo4j}}},
  shorttitle = {Neo4jsbml},
  author = {Gricourt, Guillaume and Duigou, Thomas and Dérozier, Sandra and Faulon, Jean-
↪Loup},
  date = {2024-01-16},
  journaltitle = {PeerJ},
  volume = {12},
  pages = {e16726},
  issn = {2167-8359},
  doi = {10.7717/peerj.16726},
  url = {https://peerj.com/articles/16726},
  urldate = {2024-01-22},
  langid = {english},
}
```

1.7 Changelog

1.7.1 1.0.0 (2024-03-04)

Full Changelog

Implemented enhancements:

- sbml from neo4j #20 (guillaume-gricourt)
- Add `statistics` subcmd #19 (guillaume-gricourt)

Merged pull requests:

- Fix github workflow #25 (guillaume-gricourt)
- chore(deps): bump actions/upload-artifact from 3 to 4 #24 (dependabot[bot])
- chore(deps): bump actions/download-artifact from 3 to 4 #23 (dependabot[bot])
- chore(deps): bump conda-incubator/setup-miniconda from 2 to 3 #22 (dependabot[bot])
- chore(deps): bump actions/setup-java from 3 to 4 #21 (dependabot[bot])
- Add `clean` subcmd #18 (guillaume-gricourt)
- Create sub-command sbml-to-neo4j #17 (guillaume-gricourt)

1.7.2 0.12.0 (2023-10-08)

Full Changelog

Merged pull requests:

- Take into account `annotation` field #15 (guillaume-gricourt)

1.7.3 0.11.2 (2023-10-07)

Full Changelog

Merged pull requests:

- AuraDB config #14 (guillaume-gricourt)
- Neo4jefmtool #13 (guillaume-gricourt)
- Manage plugins #12 (guillaume-gricourt)
- Fix Id association #11 (guillaume-gricourt)

1.7.4 0.11.1 (2023-09-22)

Full Changelog

Merged pull requests:

- Add `dry-run` argument #10 (guillaume-gricourt)
- chore(deps): bump actions/checkout from 3 to 4 #9 (dependabot[bot])
- chore(deps): bump github/super-linter from 4 to 5 #8 (dependabot[bot])

1.7.5 0.11.0 (2023-03-28)

Full Changelog

Implemented enhancements:

- feat(neo4jsbml): enable encryption in url #7 (guillaume-gricourt)

1.7.6 0.10.1 (2023-03-17)

Full Changelog

1.7.7 0.10.0 (2023-03-17)

Full Changelog

Merged pull requests:

- chore(deps): bump mathieudutour/github-tag-action from 6.0 to 6.1 #6 (dependabot[bot])
- Read the docs #5 (guillaume-gricourt)

1.7.8 0.9.2 (2023-02-23)

Full Changelog

1.7.9 0.9.1 (2023-02-21)

Full Changelog

1.7.10 0.9.0 (2023-01-27)

Full Changelog

Merged pull requests:

- chore(deps): bump actions/upload-artifact from 2 to 3 #4 (dependabot[bot])

1.7.11 0.8.3 (2023-01-16)

Full Changelog

1.7.12 0.8.0 (2023-01-16)

Full Changelog

Implemented enhancements:

- Provide config file #2 (guillaume-gricourt)

Merged pull requests:

- Feat modelisation pathway #3 (guillaume-gricourt)
- Package #1 (guillaume-gricourt)

* *This Changelog was automatically generated by*github_changelog_generator