

COMBINE annotation library

draft roadmap

Code: <https://github.com/sys-bio/libsemsim>

Docs: <https://sys-bio.github.io/libsemsim-docs/>

Use cases

These are the driving use cases that we need to support, starting simple and adding more complex ones as the simpler ones are implemented. They are based on the collection of “[gold standard](#)” model annotations that have been curated (from a set of models in CellML and SBML format within the domains we are focused on in the short term: electrophysiology, cardiovascular fluids, circadian rhythm, metabolic pathway, signalling pathway).

As we agree on use cases, they should be moved to issues on the Github repo and implementation tracked there. Use cases need to be prioritised in order of importance to make sure the most important ones are implemented first.

Use case 1 - creating templated composite annotations

We know the structure of the semantic composites that we want to use to annotate the models and there is a finite number of possible composites for these domains (I think). The library should provide a convenient way to create the annotations present in the gold standard collection.

e.g., *concentration_of* <chemical entity> *part_of* <compartment>, user just needs to provide the URI of the model entity and the appropriate ontology terms to identify the chemical entity and compartment.

e.g., *chemical_flow_rate_of* <chemical entity> *through* <membrane> *mediated_by* <transporter/enzyme> *with_source* <source compartment> *with_sink* <sink compartment>

Note: can always fall back on SemGen to define more complex composites, the idea is for the library to initially support the range used in the gold standard set.

Implementation note: should we embed the collection of known composites in the library?
Use a webservice to query a repository?

Use case 2 - serialising annotations to a file

Need to be able to serialise the annotations to a file for archiving, sharing, etc.

Use case 3 - parsing annotation file(s) and query for knowledge

Need to be able to parse annotations from files and then query for knowledge. Given the URI of a model entity, query the parsed RDF graph for any annotations that may exist and present them to the “user” in an appropriate manner. Likely to be returning serialised RDF as a first step, but could present the results matching the templates from use case 1 in more convenient ways?

1. Load arbitrary RDF graph (from a file, from the web, from a repository?)
2. Present query interface to the user:
 - a. All annotations for given model entity
 - b. All model entities of a particular type? E.g., all species, all concentrations, all membrane potentials...
 - c. All model entities that have a particular property? E.g., all model entities that are volumes, all fluxes through a given membrane
 - d. How generic to make this versus supporting just the small number of specific queries that we expect a tool to need?

Use case 4 - editing annotations

Users should be able to find existing annotations and then edit them. Initial implementation could simply be to find specific annotations if they exist, delete them from the graph, and then create new annotations. Which could be hidden behind a more convenient API.

If the RDF graph is made available directly, API could also be provided to enable certain types of nodes in the graph to be modified. e.g., changing the ontology term used for a particular chemical entity or compartment location.

Use case 5 - read only COMBINE archives

Need to be able to open a COMBINE archive, check the manifest for all annotation files, and parse them all into a single RDF graph. And then present a query interface for users to find knowledge contained therein. The most general would be to simply provide a SPARQL endpoint that users could use to execute SPARQL queries. Convenience API could also be provided to support the common queries that we would expect users to need. Same as use case 3 if we assume that we are only ever dealing with COMBINE archives?

1. Load OMEX archive
2. Iterate through all [RDF/annotation files](#) in the manifest parsing them into a single RDF graph
3. Present query interface to the user...

Use case 6 - annotate a model

Load a SBML or CellML model, define some new composite annotations and serialise to an OMEX archive.

Use case 7- migrate to COMBINE standard

Load an existing SBML or CellML model containing some annotations, serialise it to an OMEX archive as per the HARMONised rules.

Use case 8 - “copy” annotations from one model to another

Is there a case where the annotations from one model could be applied to another model? For example, if we wanted to annotated an older/newer version of a model and guide a user in terms of migrating the annotations. Or if we have a “gold standard” electrophysiology model and we want to use those annotations as a template for annotating a different electrophysiology model, prompting the user to help connect things together.

Use case 9 - human readable comments

It would be good to make it easy for people to put semantically meaningful comments into the annotations. Not sure sure how to do this, but RDF has types for representing comments (Dublin Core maybe?) and these could be used with appropriate typing definitions to let users provide the semantics for the comment. E.g., “I don’t know what this is but it probably is some kind of calcium concentration”, or “This is a combination of some funky reaction kinetics that are so non-standard that I can’t work out how to annotate them”. See also use case K.

Use case N - ontology lookup

A useful feature in SemGen is that it will look up appropriate ontology terms for you in a specific context of what is allowed in a particular user interface element. Do we want to provide this kind of API? Or just make use of the services provided by the EBI or BioPortal?

Use case M - model composition

Is this something that should be part of this library? It is probably the single coolest feature of SemGen - but maybe not something that can be implemented behind an API in the short term?

Use case L - term/ontology mapping

Closely related to use case N, when a curator or author finds an appropriate term that they would like to use but its from an ontology outside the recommended ontology list - can we map that term to the “closest” term in one of the recommended ontologies. Could potentially simply provide a wrapper around some of the existing ontology mapping tools (e.g., [OxO](#)).

Use case K - term request pipeline with placeholder annotations

Not sure how much something like this should be supported by the library, but perhaps it would be possible to provide a high-level API to make it easy for tool developers to request new terms, or at least make it easy for their users to make statements like “I want to annotate this as X but can’t find the right term” and have that request show up on a tracker somewhere and some kind of reference URI for the user to include in their annotation. Then in future when someone tries to resolve that URI they’d end up with that definition of what the user tried to do (as well as a human readable comment in the annotations)