Fragalysis - Tags

Author:	Duncan Peacock
Version:	Agreed
Date:	17/03/2021

Release History

Version	Contents	
< 0.2	Versions created from initial discussions	
0.3	First complete version with initial estimates and work breakdown	
0.4	Updated following discussion with Frontend development - Main changes: Tag searches moved to Frontend. Tag search apis replaced by APIs returning molecules for a target with tag information and returning session projects for a Target/author with tag information.	
1.0	Agreed design	

Introduction

This document contains a high level design on how to implement/improve/extend the ability to create Tags within Fragalysis. The purpose is to sketch out possible options to create a solution for the database/APIs that can be flexibly used by the Frontend. This functionality will be used to (ultimately) replace the existing sites functionality. Any existing sites must be migrated to the new tags functionality.

It has been agreed to implement tag functionality in stages so that basic functionality can be put live quickly and future improvements prioritised based on need/cost/complexity. Hence this document will attempt to set out a first implementation that covers the functionality that will be required by the initial front end development and provide a basis for future improvements. Potential future improvements are also discussed.

Terminology note: Fragalysis refers *Projects* on the landing page and elsewhere. For historical reasons the Fragalysis backend has two types of project model so in this document I refer to *session-projects* for clarity. From the frontend (user) perspective: Project = *session-project.*

Requirements

Initial Requirements:

No	Description
1	Tags can be attached to Molecules and Session Projects - It will be a many to many relationship.
2	Scope: Tags can be local to one session project (within a Target), or apply across projects.
3	Tags can be “local within a target” or apply across targets
4	Attributes of a tag object include: tag (text), author, create datetime, discourse_url, coordinates, shape, additional_info (free format), Colour, category
5	It should be possible to create tags in both Discourse and Fragalysis.
6	Tags should be classified into categories: Sites, Series, Forum, Other - for display on the Frontend.
7	Current sites (created/modified by the target upload functionality) should be migrated to tags.

Discussion/Limitations

ISPYB authorization

ISPYB limits access to targets to certain users and would affect any tag processing that goes across targets. Initially it will be assumed that tags will apply only within a target. Cross-target searches could still be done on the Front-end through a selection box listing all the available targets and doing separate API calls for each Target. Searches on the backend across targets will therefore be a future addition (See Roadmap).

Note also that design forums discussing tags talk about performance becoming an issue (with a lot of manytomany fields), so limiting searches initially may be prudent for this reason as well until we understand how it will be fully used.

Discourse Integration

Discourse is currently still in the process of being launched. Creation of tags in Discourse is currently only possible at session-project level (i.e. on creation of a topic). Fragalysis can/will push tags to Discourse, not vice-versa. It may be possible other options for molecules (such as bookmarks) can be used. However as it is currently unclear exactly what the ultimate interaction between the two platforms will be and what search facilities will be required, it also makes sense to make this a future addition.

Tag searching

For usability, it will be important to ultimately make this as flexible as possible, but again this is an area where it makes sense to start with a few limitations and expand later on:

- Tags will be at least 3 characters long, so that they are meaningful
- Initial searches will be on whole tags.
- Searches will be a responsibility of the Frontend - at least initially.
- Partial tag searches/autocomplete functionality can be added later (and not just for tags). For performance reasons this would likely be two levels (find the tags and then find molecules connected to the tags) This site has some information on how this might be achieved through the Django Rest Framework: https://drf-haystack.readthedocs.io/en/latest/02_autocomplete.html

Migrating current “Site” functionality

The new functionality should replace the existing functionality for Sites in Fragalysis. Hence the target uploader should be changed to create tags instead of molecule groups.

After looking at this in more detail it seems that this could potentially affect several APIs in the “scoring” functionality (not currently in the fragalysis backend documentation). To avoid breaking too much code in the initial phase I propose that, rather than replacing MolGroup (+ associated functionality) - we simply create the tags alongside the MolGroups when uploading target sets and refer to the MolGroup_id in the tag model. When the tag is retrieved the MolGroup information can be added to it and achieve the same purpose at a lower cost/risk without breaking existing APIs.

Removing the scoring functionality can therefore also be a future ticket - we can fully analyse how/where the Model is used separately (see Roadmap).

Use of existing Tagging functionality

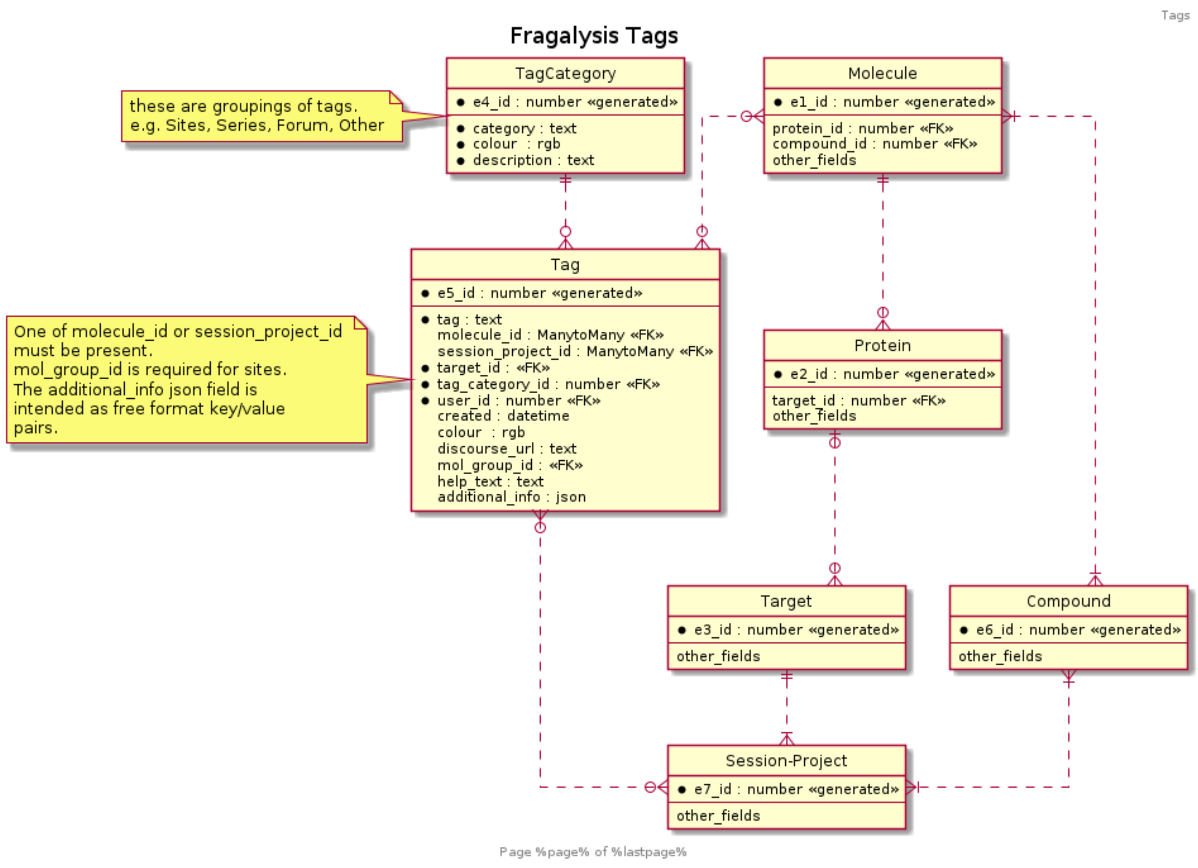
Note that there is existing Django tagging functionality, but this does not fully satisfy the scientific information/Discourse storage requirements. Tags are normally limited to the tag

text and perhaps some additional “Help Text” (I have added this - it could be used as a tool tip/search field):
https://django-taggit.readthedocs.io/en/latest/getting_started.html
https://radiac.net/projects/django-tagulous/documentation/

Functionality (Initial Code Ticket):

No	Description
1	API to return all molecules/tag information for a given Target
2	API to return all session_project/tag information for a given (session project) User
3	Return Tags for a Session Project
4	Return Tags for a Molecule
5	Return Tags for an author
6	Create Tagcategory/Update Tagcategory/Get Tagcategory(s)
7	Create Tag/Update Tag/Delete Tag for Molecule / Session Project.
8	Modify upload target functionality to create “Site” tags in addition to the mol-group functionality.
9	Migrate Sites to tags (will leave current functionality for backwards compatibility)

Data Model


Tags
Fragalysis Tags
TagCategory
● e4_id : number «generated»
● category : text
● colour : rgb
● description : text
these are groupings of tags.
e.g. Sites, Series, Forum, Other
Molecule
● e1_id : number «generated»
protein_id : number «FK»
compound_id : number «FK»
other_fields
Tag
● e5_id : number «generated»
● tag : text
molecule_id : ManytoMany «FK»
session_project_id : ManytoMany «FK»
● target_id : «FK»
● tag_category_id : number «FK»
● user_id : number «FK»
created : datetime
colour : rgb
discourse_url : text
mol_group_id : «FK»
help_text : text
additional_info : json
One of molecule_id or session_project_id
must be present.
mol_group_id is required for sites.
The additional_info json field is
intended as free format key/value
pairs.
Protein
● e2_id : number «generated»
target_id : number «FK»
other_fields
Target
● e3_id : number «generated»
other_fields
Compound
● e6_id : number «generated»
other_fields
Session-Project
● e7_id : number «generated»
other_fields
Page %page% of %lastpage%

1. Tags are grouped within categories. It is proposed to upload the basic categories (Sites, Series, Forum, Other) as part of the migration.
2. A tag is identified by a text field called “tag” of at least 3 characters (to be reviewed). Note that tags should be “meaningful”.
3. A tag must either be attached to a Molecule or a Session Project within a Target. At a technical level I am planning to use inheritance to allow specific behaviour for Molecule/Session Project related tags. This could be extended at a future date.
4. A mol_group_id will (optionally) be present for tags relating to Molecules. The tag in this case will be the site name. These will be created when the target is uploaded. Mol_group will continue to hold information on coordinates etc. This will also be returned from the tags api.
5. Optionally, discourse_url, colour, help_text and addional_info (in the form of key/value pairs) may also be provided.

Processing/APIs

Processing:

The following changes will be required for the initial ticket:

- New Models (Tag, Tagcategory)
- New Views/Serializers/URLs
- New API's for the new Tables and specifically a new api to replace the current functionality provided by the current APIs **GET /api/molgroup and GET /api/molecules** that load the site information on the left hand side of the viewer page.
- Change upload target code to create tags for sites (alongside current molecule group functionality).
- Code to create tags from existing sites (via existing MolGroup model) and Initialise tag-categories.
- Additional tests for the new APIs
- QA/Implementation/Documentation

Implementation should be relatively straightforward as it is primarily new code (with the exception of the changes to the Upload_target functionality). It can be implemented without breaking the existing Frontend.

APIs

Target_molecules (New)

Returns Protein/molecule/tag information for a given target_id. This will be the main API initially used by the Frontend. This API ultimately replaces the current functionality provided by the GET /api/molgroup and GET /api/molecules to the front end to return site information.

The API will work in a different way to the current process. At the moment the sites are returned to the frontend via api/molgroup. Then for each site(mol group), api/molecules is called to get the molecule information for that site. In the new methodology a **single** call will be made to return **all** molecules related to the target and their tag information (if available). Due to the size of the likely reply, the GET limit is likely to be exceeded so the functionality is assumed to be a POST operation. The new API will make use of existing (ISPYB safe) functionality.

The reply will consist of:

1. Serialized target information
2. Followed by a group of serialized molecule/tag id information
3. Followed by a group of tag information for the tags in (2).
4. Followed by a group of tag category information for the tag categories in (3).

POST ../api/target_molecules/

- Returns all protein/molecule/tag information for the given target as JSON string.

Params:
- Target name

Request (Raw example)
{“title” : “Mpro”}

Reply (JSON) (likely to change in small details - but will be agreed with the front end).

```
"target": [
{
    "id": 62,
    "title": "Mpro",
    "project_id": [
        2
    ],
    "protein_set": [
        29281,
        29274,
        29259,
        29305,
        ...,
    ],
    "template_protein": "/media/pdbs/Mpro-x10417_0_apo.pdb",
    "metadata":
"http://fragalysis.diamond.ac.uk/media/metadata/metadata_2FdP5OJ.csv",
    "zip_archive": "http://fragalysis.diamond.ac.uk/media/targets/Mpro.zip"
},
"molecules": [
{ data : {
    "id": 13912,
    "smiles": "CN(C)c1ccc(C(=O)Nc2ccccn2)cc1",
    "cmpd_id": 796,
    "prot_id": 29281,
    "protein_code": "NUDT7A_Crude-x2226_2",
    "mol_type": "PR",
    "molecule_protein": "/media/pdbs/NUDT7A_Crude-x2226_2_apo_x5GxiLq.pdb",
    "lig_id": "LIG",
    "chain_id": "Z",
    "sdf_info": "     RDKit          3D 18 19  0  0  0  0  0  0  0
0999...",
    "mw": 241.12,
    "logp": 2.4,
    "tpsa": 45.23,
    "ha": 18,
    "hacc": 3,
    "hdon": 1,
    "rots": 3,
    "rings": 2,
    "velec": 92
},
 "tags_set": [14652,14654, 14655]
,]
}, ],
tag_info {[
    "id": 14652,
    "tag": "tag text",
    "discourse_url": "www.somediscourse.com/t/33",
    "category_id": 1,
      "Coordinates" : {x_coord, y_coord, z_coord}
      "Mol_group_id" : 234
    Etc (include all relevant fields from tag table).
],},
tag_categories {[
    "id": 1,
    "category": "sites",
    Etc. (include all relevant fields from tag-category table)
],}
```

Session_project (Modified)

Modify existing API *session-projects* to return tag information with the session_project (GET).

Tag-category (New)

Used to create groupings of tags to organise tags on-screen.

POST ../api/tag-category/
- Create tag_category as JSON string.
- Fields {category, colour, description}

GET ../api/tag-category/?category?
- Returns tag_category for the given id as JSON string of (category, colour, description). Default is all tag categories.

PUT ../api/tag-category/{tag_category_id}/
- Replace tag_category as JSON string.

Initially filled with "Sites", "Series", "Forum", "Other".

Note that DELETE tag-category will not be provided as an API for security reasons.

Tags (New)

Create/Update/Delete/Get tags related to individual molecules or session projects.

POST ../api/tags/
- Create a tag for molecule or session_project.
- Fields {*tag, molecule_id/session_project_id, *target_id, *tagcategory_id, *user_id, created, colour, discourse_url, mol_group_id, help_text, additional_info : json}

PUT ../api/tags/{tag_id}/
- Replace tags for tag id as JSON string.

DELETE ../api/tags/{tag_id}/
- Delete tags for tag_id

GET ../api/tags/{tag_id}
- Returns all information for the tag_id.

GET ../api/tags/?tag=<tag>
- Returns all tag_info for the user_id.

GET ../api/tags/?user=<user_id>
- Returns all tag_info for the user_id.

GET ../api/tags/?target=<target_id>
- Returns all tag_info for the target_id.

Note that to save time these last two apis will be based on core Django functionality for a Model. Additional functionality (such as ISPYB authorisation) can be added at a future point if required.

Roadmap/Estimates

This analysis describes a logical first phase of the tagging functionality and several possible future improvement tickets. These are described below:

Ticket	Contents	Estimate
Tags-01	Initial functionality as described in this document.	Around 7 days.
Tags-02	Discourse Integration. Further analysis based on Discourse use/requirements.	small/medium (depending on requirements)
Tags-03	Replacement of Molgroup functionality. Once the Frontend has changed to use tags, it may be possible to simplify the backend to remove the MolGroup model functionality. Use of scoring functionality in Frontend to be investigated.	small/medium
Tags-04	Searches across targets. This adds complexity as it affects authorisation - so is initially limited to the frontend. Details to be investigated.	medium

Tags-05	Autocomplete searches. This is a wider question than just tags as it's a strategic decision for Fragalysis. A suitable library must be chosen and database changes scoped out. Further analysis required.	medium/large

Note that I've updated the estimate following the experience with Discourse to include some expected time for rework following the initial development.

Appendices

Discourse:

See: https://meta.discourse.org/t/a-comprehensive-guide-to-discourse-tags/121041
And: https://blog.discourse.org/2017/10/its-time-we-talked-about-tags/

1. Discourse Tags are attached to a topic rather than a post (so a session project for a target rather than a snapshot). So they are not directly attached to molecules or indeed snapshots.
2. Tags are currently created as a list using the Discourse_api when creating topics.
3. They are adjusted by editing the topic. Currently, there is no mechanism to do this so I would have to add it.
4. It would be a push from Fragalysis to Discourse - Creating/Updating a tag in Fragalysis could update Discourse, but if you, for example, deleted a tag in Discourse it would not be automatically deleted in Fragalysis.
5. In Discourse, there is the ability to restrict tags to certain categories (in our case these are Targets) and users/levels and also to create specific tag relationships like parent/child or only one tag of this category per topic). (https://meta.discourse.org/t/tags-category-restrictions-tag-groups-relationships/48260).
6. There is a nice sidebar plugin for tags.
7. The api provides the ability to create/update/list tag groups.
8. .. but there seems to be no obvious way of getting all posts with a certain tag via the api (although this might be "hidden" functionality - there is quite a lot of that - I just need to experiment).

Comments from Rachael/Frank for future analysis:
- It might be possible to use bookmarks for posts or molecules?
- In place of tags, it might be possible to search in the Discourse text for molecules.