

BBF RFC 86: Creating native registry functions to accommodate mutant libraries

Kenneth O. Xu, Eric M. Walters

11/14/11

1. Purpose

The current registry system accommodates normal parts fairly well but has difficulty when adding mutant libraries. If all mutant offspring were added as new parts, the registry would be comprehensive but the parts registry would become filled with parts and eventually become unmanageable to navigate. If the mutants were added directly to the source's page, it could become unwieldy and unintuitive to find. We propose a system of adding a few features to the registry that SHALL allow a user-friendly method of visualizing mutant offspring and siblings while condensing the registry as much as possible when mutants are introduced to allow for easier navigation.

2. Relation to other BBF RFCs

None as of yet.

3. Copyright Notice

Copyright (C) The BioBricks Foundation (2011). All Rights Reserved.

4. Mutagenesis Library Additions:

Specifically, we propose two additions. The current top bar above a parts page looks like this:

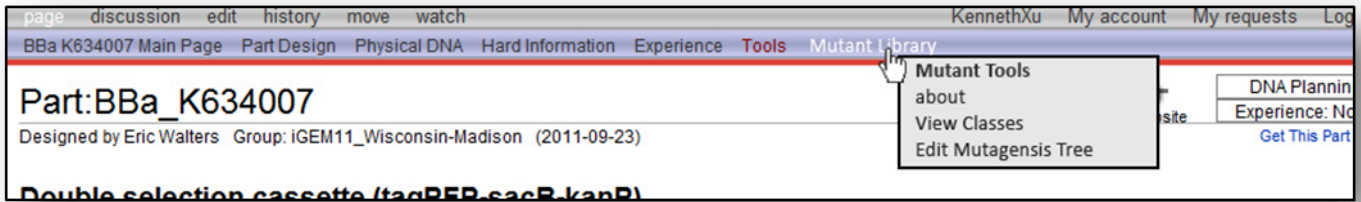


1) We propose adding another dropdown menu to accommodate the new features of the mutagenesis library. This drop down menu can be added before or after tools as is pictured in the mock-up above.

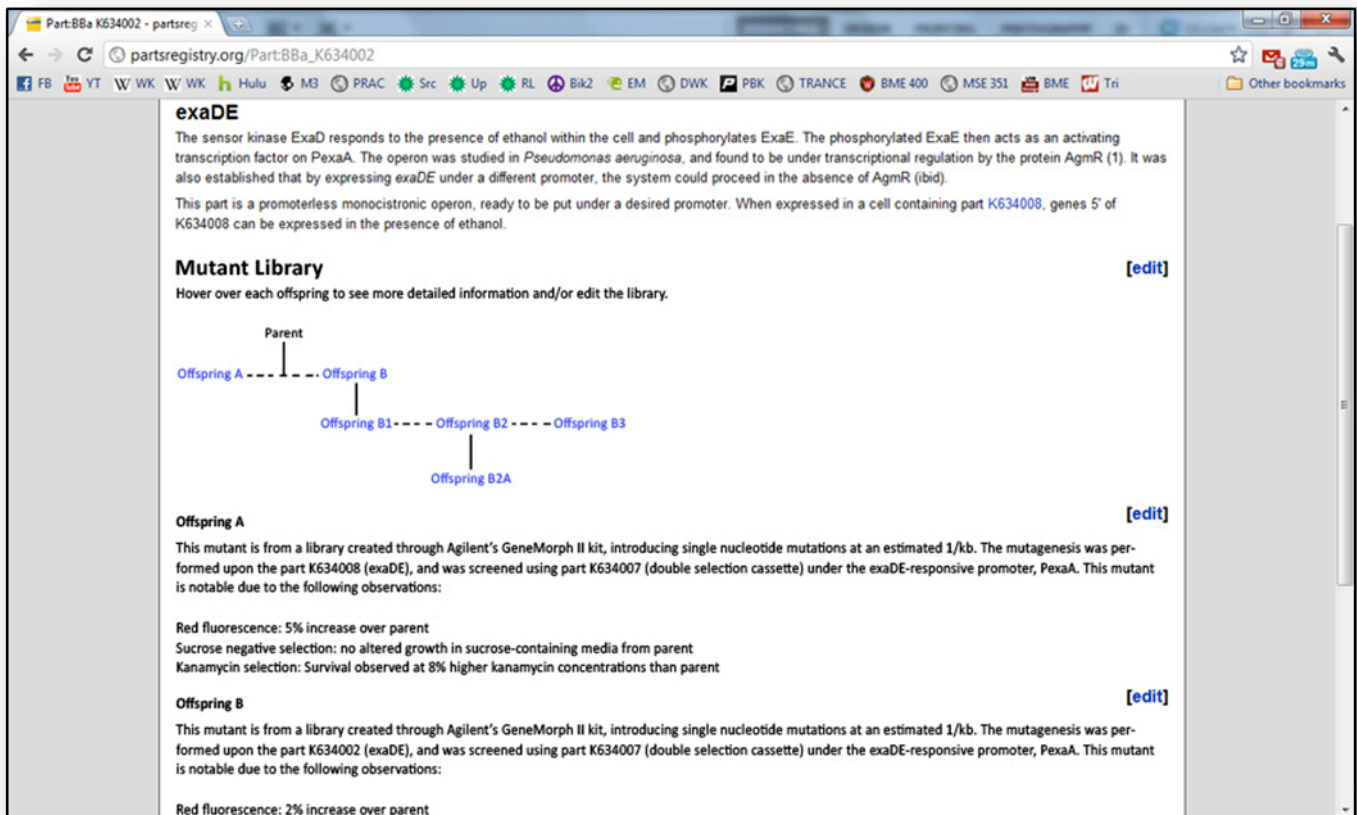
Upon mouseover of our proposed system, a drop down menu with the following entries are shown: Mutant tools, about, view classes and edit Mutagenesis Tree. This is of course, subject to change but covers the core basic entries.

- **Mutant Tools:** provides tools to edit mutants, this can be appended later on

- **about:** gives a basic run down of what the mutagenesis library is and how it is used
- **View Classes:** Displays the available classes of mutants available along with the protocol used to get the classes
- **Edit Mutagenesis:** allows adding, editing or remove mutants from current trees.



2) The real meat and potatoes of our proposal is to include a fully editable Mutant Library section at the bottom of the page just below the footer. This page would firstly include an autogenerated “Mutant Library Tree” of all available mutants of the part in question and the offspring of those mutants. In addition, the page would include the protocols of how the initial offspring were made and the characterization of those offspring.



Ideally, we want a simple visual representation of the entire family that is easy to navigate and does not clutter your screen or the registry. This was our primary inspiration for creating the mutant library tree. Features of the tree include:

- **Clean, holistic visualization of the library:**

- A solid/dashed line indicates direct offspring/siblings. A 2-D space was the primary limitation and inspiration for this so that direct offspring do not have to stem directly from the same parent. This reduces clutter while keeping it clear the siblings are exactly that.
- **Intuitive naming system:**
 - Instead of displaying the 7 digit names of all the parts and the mutant identifier, the first offspring is just displayed as “A”, the second as “B” and so on. If more than 26 are used, the 27th would be named AA then AB and so on.
 - The 2nd generation would be given a number designation to avoid confusion with the first generation. For example, 2nd generation offspring from the 1st generation offspring “B” would be called B1,B2 and B3.
 - Finally, these two methodologies would switch every generation so that users can get a sense of what kind of mutants they are using from just the name.
 - In the future, if mutant families grow too large and too many (10+) generations, we can create a nickname system for those lengthy generations. For example if a 10th generation offspring was named “A3B15A3A6G67”, one could simply give it the nickname of “pfleger” and thus the mutants of the 11th generation would be named “pfleger1, pfleger2, etc”. Starting with a number after the nickname is important to avoid confusion.
- **Quick, easy navigation between mutants**
 - Upon mouseover of each of the mutants, a popup would occur that details its exact relations to its other mutants. This information would include the offspring name of its parents, siblings,and offspring allowing for editing of this library. (Note that each offspring is called a “Library”)



We believe this addition can be extremely powerful in allowing users to discover the relation between parts but it can also be abused and neglected. A strict requirement system for submitting mutants must be implemented so that this system can be self sustainable. It is really up to the users to create a quality set of descriptions and sequence the offspring to ensure high fidelity in the system.

A few basic but key requirements:

- The offspring must be sequence verified prior to submission
- A protocol description and some characterization must be included
- Relation to a parent, sibling or offspring must be clearly defined

We believe that if these rules are followed, this system will provide a new, powerful

avenue for organization in the parts registry and allow the users an opportunity to actually modify another team's part and have that rich history of team collaboration be recorded and displayed in a clean and elegant manner easily accessible to all others willing to participate.

5. Author's Contact Information

Kenneth O. Xu
kkxu@wisc.edu
952-905-4298

Eric M. Walters
ewalters@wisc.edu
608-358-6340

6. References

None as of yet.