



St. Jude BioHackathon

Title

Flexible, customizable, browser-editable persistent storage via databases

Category

Data Management

Challenge

Build a flexible data table that could load in arbitrary records with an arbitrary number of fields. For each record, allow some additional fields to be populated by human inputs. Record the updated records (with human inputs) in a persistent database. Package the system so that it could be easily deployed and customized.

Or a more specific use-case would be: 1) Beginning with an annotated VCF file, load annotation information for multiple genetic variants into a web-app data table, where each row represents a variant. 2) In this contrived scenario, three different decisions need to be made by a human. How and on what basis these decisions are made is immaterial. The decisions are: if this variant were a color, what which would it be? if this variant had a favorite ice-cream flavor what would it be? if this variant were an animal, what would it be? Receive the human decisions via either drop-down menus or text inputs on a per-row basis via three different cells (i.e, the cells contain drop-downs). 3) Record those decisions in a connected database in a persistent way. Optionally, create one or more views to query the stored data. 4) Package the aforementioned system so that it could be readily deployed as a basic template from which someone could customize the table to represent any type of entity described by a set a fields, and the decisions could be any decisions based on the data in those fields.

Benefit

This is a task I've frequently encountered and made modest efforts to solve, none of which I've been fully satisfied with due to my limited knowledge of front-end interface development. I recently saw someone post a request for help on a STJUDE forum for precisely this same need. I think the creation of such a core tool would be a springboard for addressing a wide range of human-machine-automation interface problems that people have been reluctant to tackle due to the scarcity of software development bandwidth.

Helpful Tools, Packages, or Software

Vue, Flask, SQL, MongoDB, Django, Shiny, Dash. There are a lot of options here, but picking something that can easily be templated and easily deployed is key.

Test Data

While not a data set, per se, an example would be integrating human judgment calls regarding the technical validity and pathogenicity of genomic variants. See Fabric Genomics and StJude's own PecanPIE for custom solutions to this type of problem. My contention is that a more generalized solution is achievable that could be forked and extended to address a wide range of similar challenges.