MageComet

Vincent Xue

August 11, 2011

1 Purpose

MageComet is a web application designed for quick annotation and manipulation of MAGE-TAB files. The webapp features tools that allow curators to easily edit MAGE-TAB documents, without spending excessive time and effort formating files to MAGE-TAB specifications. MageComet's goal is to reduce the amount of time editing MAGE-TAB documents by automating tasks commonly encountered during curation. This automation allows curators to focus more on the biological data presented instead of spending time formating the document.

2 Loading Files

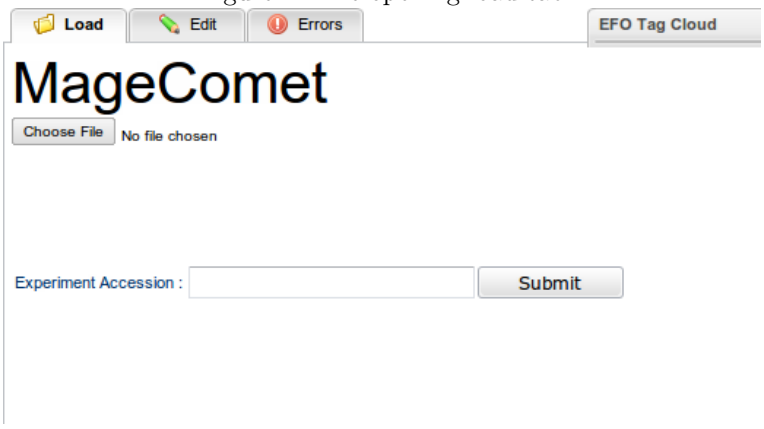
There are two ways to begin editing a set of MAGE-TAB files.

Direct Loading

Direct loading is used when MAGE-TAB documents are locally stored on the client machine. There are two files required to proceed, namely the the SDRF, and IDF files which are usually appended by "sdrf.txt" and "idf.txt" respectively. Though the MAGE-TAB specification does not mandate that SDRF and IDF files have these suffixes, MageComet uses these suffixes during load, and will not proceed without them.

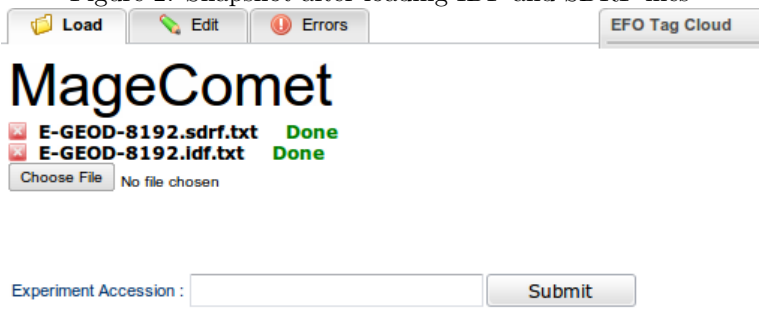
When starting the MageComet webapp, the user will be presented with the "Load Tab". This can be seen in Figure 1.

Figure 1: The opening load tab



To load the MAGE-TAB files to the server, the user must click the "Choose File" button. A popup will appear and the user can select either the SDRF file or the IDF file to upload first. The same "Choose File" button must be clicked after the first file has loaded. When both files have successfully been loaded, the screen should like like Figure 2.

Figure 2: Snapshot after loading IDF and SDRF files



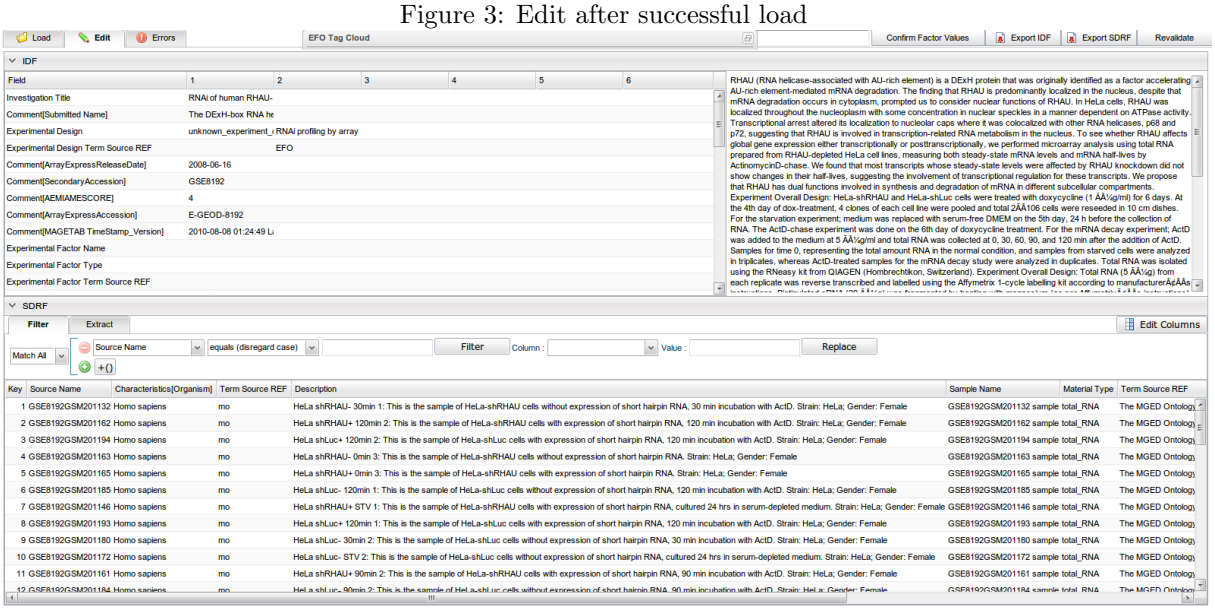
ArrayExpress FTP

To load a file via the ArrayExpress FTP service the user can use the form directly below the "Choose File" button. The user can simply type in the experiment accession and click submit, which will automatically fetch and load the IDF and SDRF files.

3 Filtering Data

Once the files have been loaded the user can proceed to the "Edit" tab. If loading has been successful, the page should resemble Figure 3.

Figure 3: Edit after successful load



The first tool that is visible to the user is the filter and replace tool as shown in Figure 4. This tool is very similar to the search and replace tool in excel, but it is designed to be more specific for editing columns of data.

There are 5 components of this tool that a user can customize.

A Filter Column - This is the column that has a trait of interest. Usually this is the "Description" column.

B Logic - This dropdown box determines the logic a user wants to apply on the filter column It contains a list of items such "equals", "contains", "does not contain" and more.

C Filter Value - This is the value the user will filter on.

Parts A, B, and C contain the logic for filtering.

Example: A user wants to filter for all the rows that contain female in the description column. The values for the A, B, and C would be A:Description, B:Contains, C:Female

D Target Column - The column whose value will be set

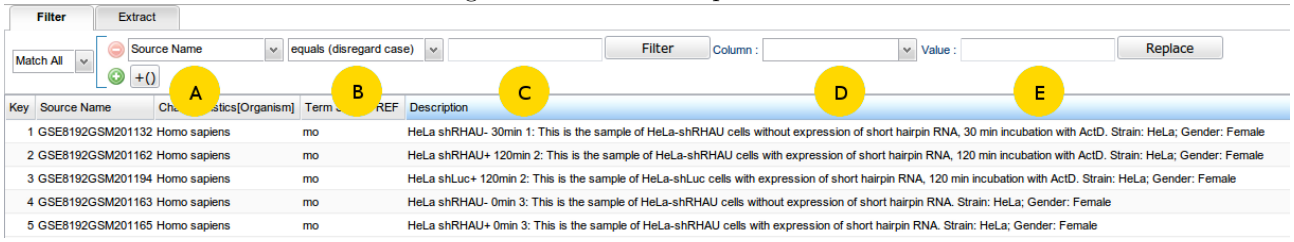
E New Value - The value that the target column will be set to.

Parts D and E contain the logic for replacing.

Example: A user wants to replace all the values in the column Characteristics[sex] with female. The values for D and E would be D:Characteristic[sex], E:female.

Note: The value for D must be a column that already exists

Figure 4: Filter and Replace



4 Column Manipulation

All column manipulation must be performed from the "Column Editor". The button to activate this editor is shown in Figure 5 and the popup editor is shown in Figure 6.

In this column editor, all of the rows are represented as rows. The left side of the editor designated by **A** is the clipboard. This section of the popup acts as a scratch buffer where columns are discarded to. In addition, new columns also appear in this clipboard before they are placed.

The right side of the editor designated by **C** is the representation of the SDRF columns. This section shows what columns are in the SDRF and in what order it appears.

Figure 5: Column editor button

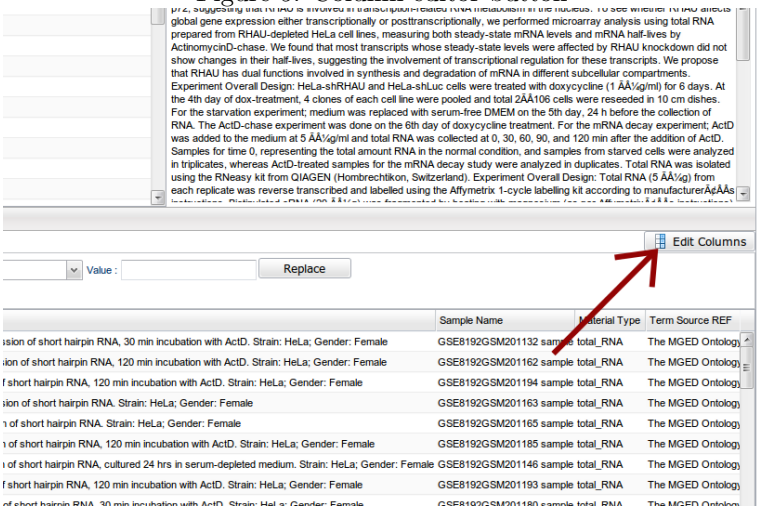
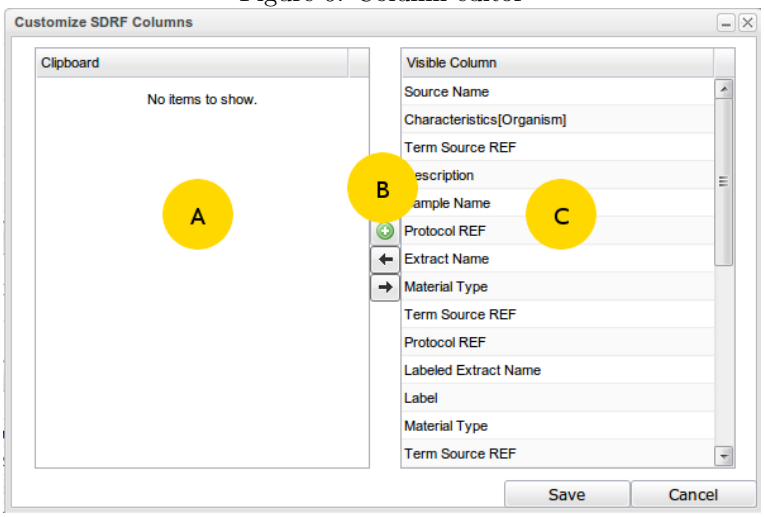


Figure 6: Column editor



Adding Columns

To add a column, click the green plus sign below **B**. This will add a row to the clipboard **A**.

Removing Columns

To remove a column, drag the desired column from section **C** to section **A**. In addition, a user can also select the column from section **C** and click the ← button.

Reordering Columns

To reorder column orders, drag and drop the row to the desired position.

Renaming Columns

To rename a column, double click the row.

5 Extracting Data

The extract tool is a feature that splits row values formated with delimiters. If all of the rows in a column are formated similarly, this tool allows the user to separate a single column into many. An example of splittable column is seen in Figure 7.

In this example the "Description" column should be separated into new columns for organism part, disease stage, sex, age, set, and disease state. Automatic splitting is difficult and avoided because a single column could have many different combinations of delimiters. In this example alone, there are commas, semicolons, colons, and text delimiters in a single row. However, because this format is consistent throughout the column, the extract feature implemented in MageComet works well in these situations.

Figure 7: Example of a splittable column

Description
peripheral blood, n/a. uveitis status: No uveitis; gender: Female; age: 57; set: 1; tissue: Blood; disease state: Sarcoidosis
peripheral blood, n/a. uveitis status: Uveitis; gender: Female; age: 59; set: 1; tissue: Blood; disease state: Sarcoidosis
peripheral blood, n/a. uveitis status: No uveitis; gender: Male; age: 32; set: 1; tissue: Blood; disease state: Control
peripheral blood, n/a. uveitis status: Uveitis; gender: Male; age: 61; set: 1; tissue: Blood; disease state: Axial Spondyloarthropathy
peripheral blood, n/a. uveitis status: Uveitis; gender: Male; age: 28; set: 1; tissue: Blood; disease state: Sarcoidosis
peripheral blood, n/a. uveitis status: Uveitis; gender: Male; age: 51; set: 1; tissue: Blood; disease state: Axial Spondyloarthropathy
peripheral blood, n/a. uveitis status: Uveitis; gender: Male; age: 35; set: 1; tissue: Blood; disease state: Axial Spondyloarthropathy
peripheral blood, n/a. uveitis status: No uveitis; gender: Female; age: 36; set: 1; tissue: Blood; disease state: Control
peripheral blood, n/a. uveitis status: Uveitis; gender: Female; age: 33; set: 2; tissue: Blood; disease state: Axial Spondyloarthropathy
peripheral blood, n/a. uveitis status: Uveitis; gender: Male; age: 68; set: 1; tissue: Blood; disease state: Axial Spondyloarthropathy

The extract tool is located in the same panel as the filter tool. By clicking on the "Extract" tab, the user can activate the panel that should resemble Figure 8. Like the filter tool, the extract tool has 5 components which the user must fill in.

A From Column - This is the column that is splittable. Usually this is the "Description" column.

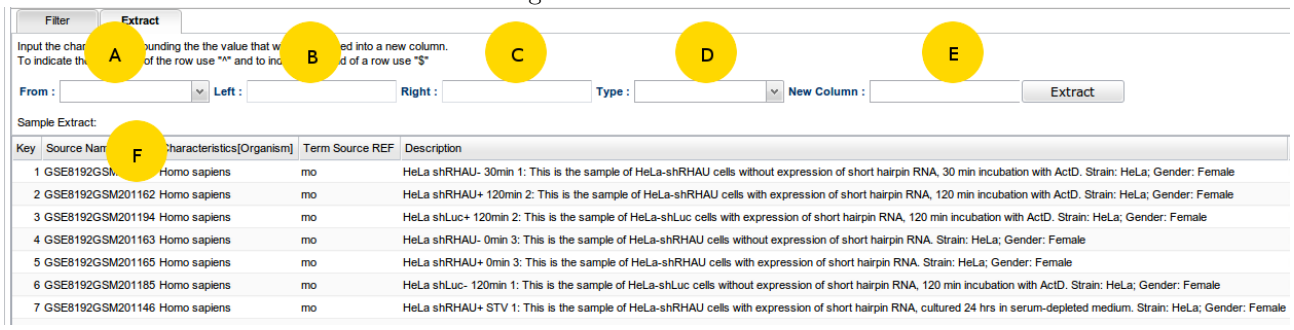
B Left Input - This field should be filled in with the text that is left of the value that should be extracted. The input however should be **unique** as it will only match the first input found.

C Right Input - This field should be filled in with the text that is right of the value that should be extracted.

D Type Column - This field is the target column type. The possible choices are "Clipboard", "Characteristic", Factor Value", or "Both".

E Column Name- This field is the target column name. This is the value that will fit in between the brackets [ ].

Figure 8: Extract



When the user completes values for **B** and **C**, the section designated by **F** will show the sample values that will be extracted.

More Examples

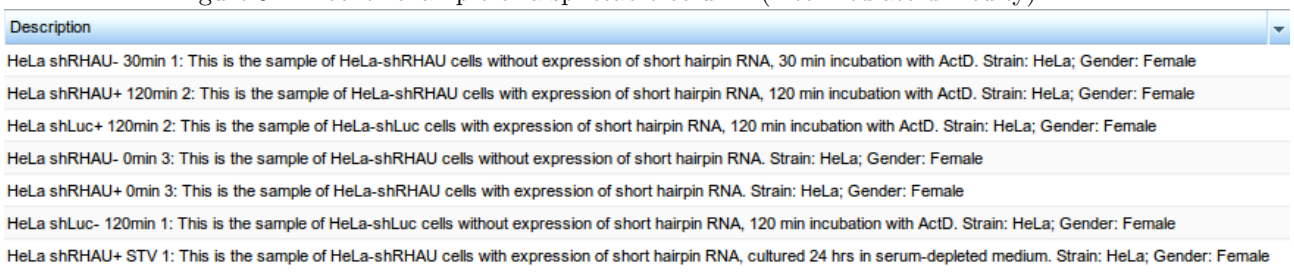
The following lists the input values for **B** and **C** that will extract the targeted value successfully for Figure 7.

Left	Right	Sample Extract Row 1
^	,	peripheral blood
status:	;	No uveitis
gender:	;	Female
set:	;	1
tissue:	;	Blood
state:	$	Sarcoidosis

Table 1: Sample inputs and outputs for Figure 7

The following lists the input values for **B** and **C** that will extract the targeted value successfully for Figure 9.

Figure 9: Another example of a splittable column (intermediate difficulty)



Left	Right	Sample Extract Row 1
^	[0-9]	HeLa shRHAU-
[+-]	min	30
gender:	;	Female
RNA,	min	30
Strain:	;	HeLa
Gender:	$	Female

Table 2: Sample inputs and outputs for Figure 9

6 Tag Cloud

The tag cloud is a feature that helps curators identify important biological information text-mined from the IDF and SDRF text. To open the window, click on the button designated by Figure 10. A window should pop up that resembles Figure 11.

In the tagcloud representation of the text, each item represents an EFO ontology term that has been mined from 2 documents. The size of an item corresponds to where the term came from. The smallest size, indicates that only the IDF mentions the word. The medium sized text indicates that the text was mined from the SDRF, and the largest sized text indicates that both the IDF and SDRF mention the ontology term.

Adding characteristics via tagcloud

A sub feature of the tag cloud is the ability to add a characteristic term to all rows in a document. This feature is useful when some vital information is mentioned in the experiment description but is not mentioned in the SDRF document. By clicking on the term in the tag cloud, a new popup will show up, giving the user the option of adding the selected term to the SDRF. Figure 12.

Figure 10: Default tagcloud position

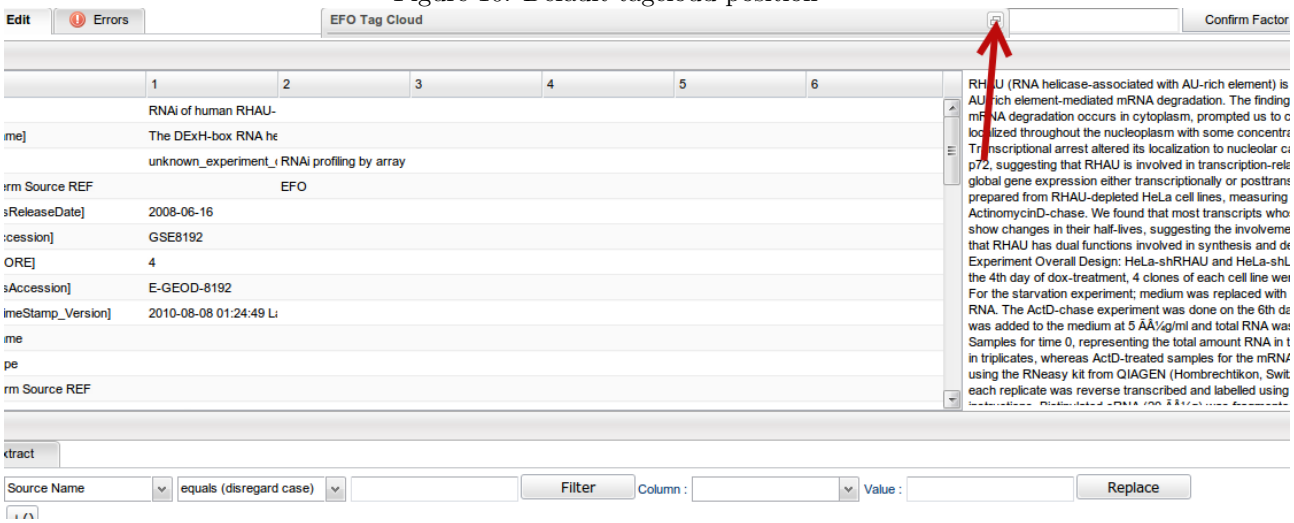
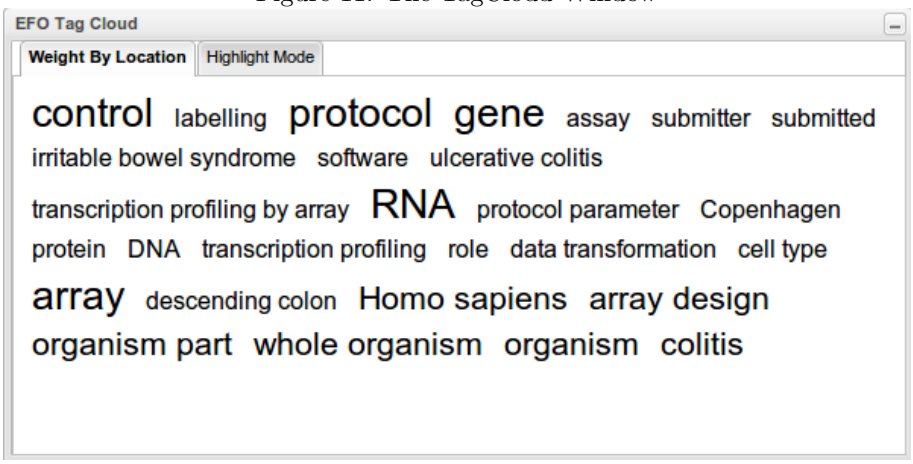


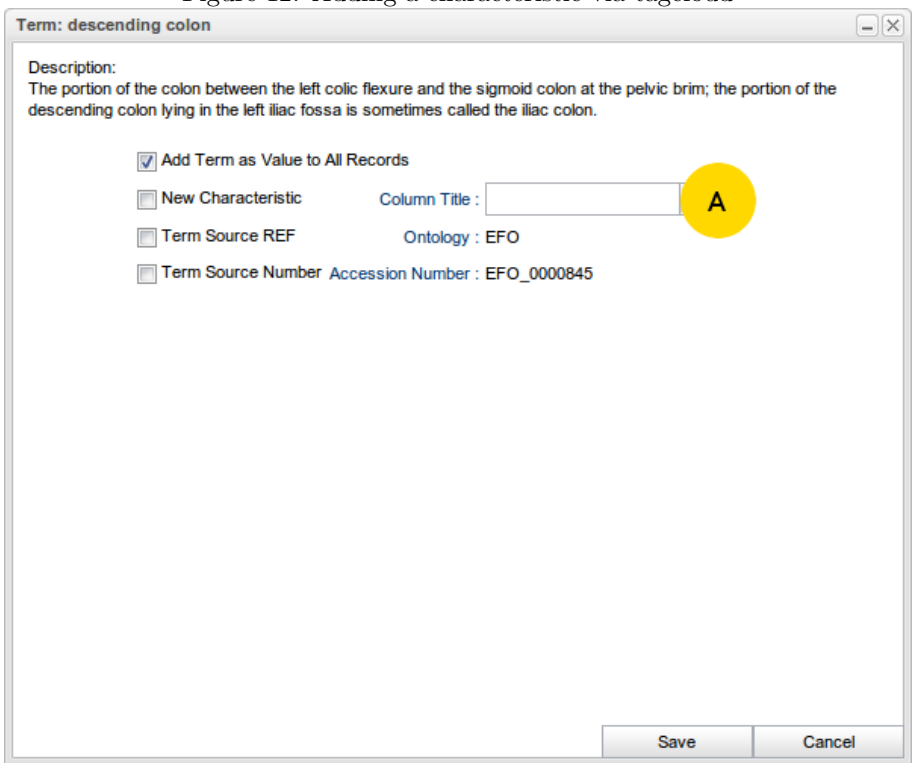
Figure 11: The TagCloud Window



When a tag cloud item is clicked, a popup will appear, providing some useful information about the term. It usually provides a description near the top, which is pulled from the EFO ontology and the term source number.

The user can choose to add a characteristic column, term source ref column, or a term source number column to the SDRF, depending on how granular the curation is. The input field designated by **A** in Figure 12 is the value that will be placed in the brackets [ ].

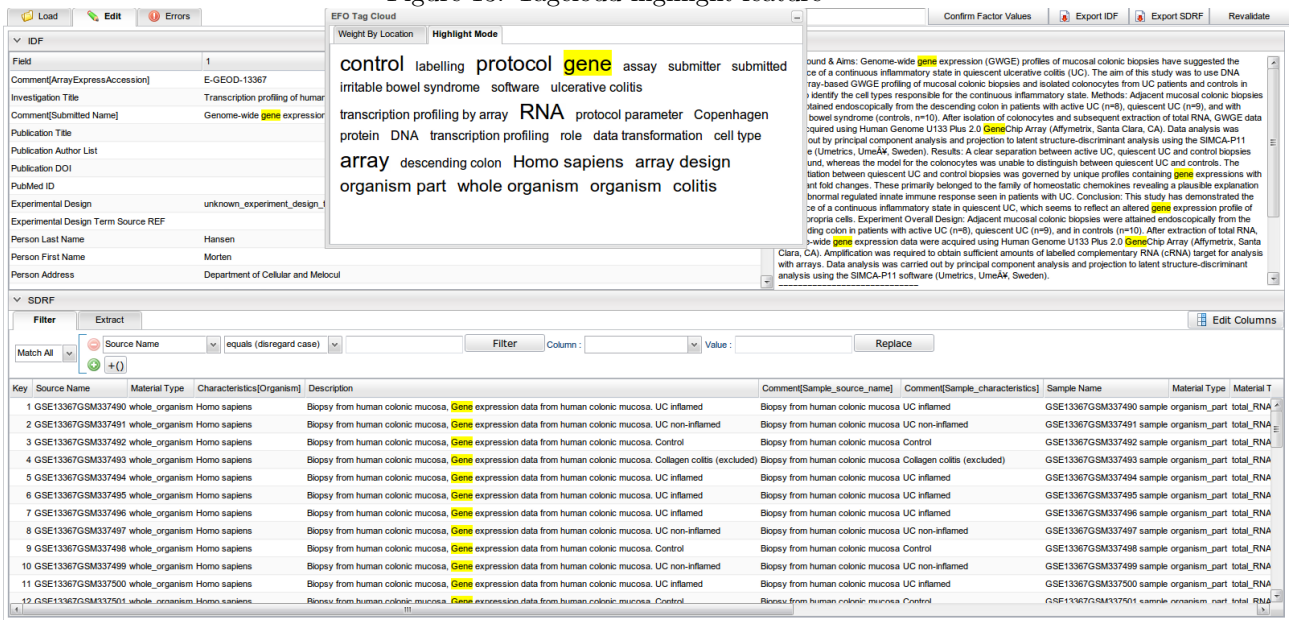
Figure 12: Adding a characteristic via tagcloud



Highlighting text via tagcloud

Another sub feature of the tag cloud is to highlight text. This feature can be accessed by clicking the "Highlight Mode" tab in Figure 11. The same cloud will appear, but clicking on a term will cause it to highlight on the page. This is demonstrated by Figure 13.

Figure 13: Tagcloud highlight feature

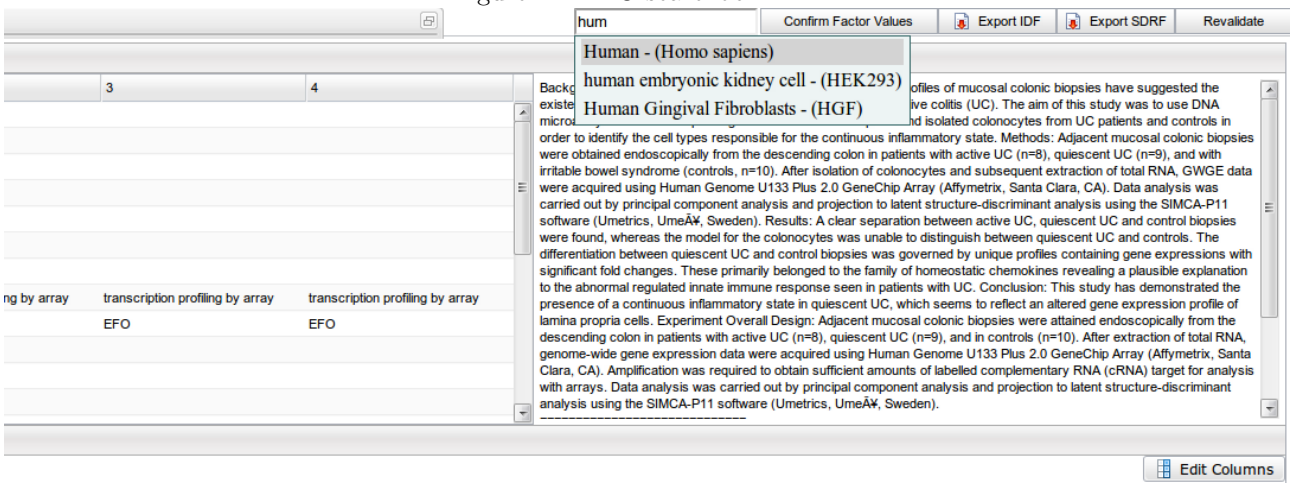


7 EFO Search Box

The EFO search box is a convenience feature implemented into MageComet. If the user wants to confirm that an EFO term exists in the EFO ontology, the EFO search box can be used as shown in Figure 14.

The search field autocompletes based on the query text and displays 3 terms that match the query. If an item is in parentheses as shown with human, the value in parentheses is the standard name for the synonym found. If the user wants to copy the standard ontology name, pressing enter while over the term will fill in the field, making it copyable.

Figure 14: EFO searchbox



8 Adding Factor Values

When the user has finished extracting the factor values in the SDRF document, the IDF document must be updated to reflect the changes. The IDF document can be updated via the "Confirm Factor Value" button as shown in Figure 15.

Once clicked a window will appear that shows all of the Factor Values in the SDRF. Figure 16. After filling in the corresponding Factor Value Types and clicking save, the IDF document will have the correct values automatically inserted.

Figure 15: Confirm factor value button

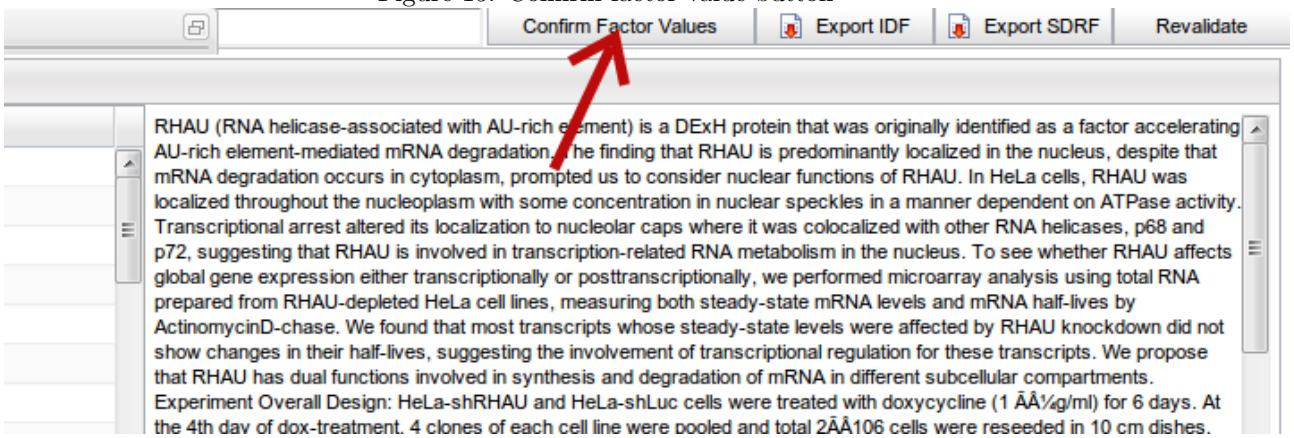
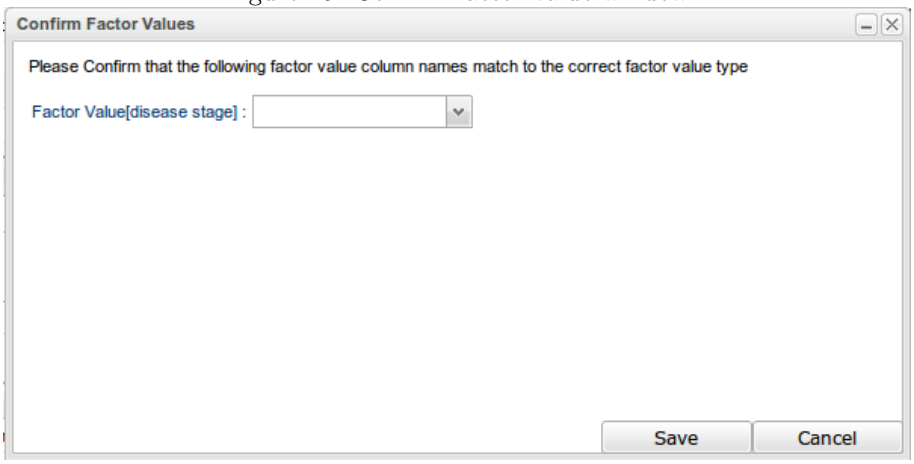


Figure 16: Confirm factor value window



9 Revalidation/Validation

MageComet also has implementation of the Limpopo validators. A user can see all the of the errors by clicking on the Errors tab as shown in Figure 17.

In the errors tab, the user can selectively view errors, warnings, missing information, and revalidate the current MAGE-TAB documents after changes have been made.

Figure 17: Errors button

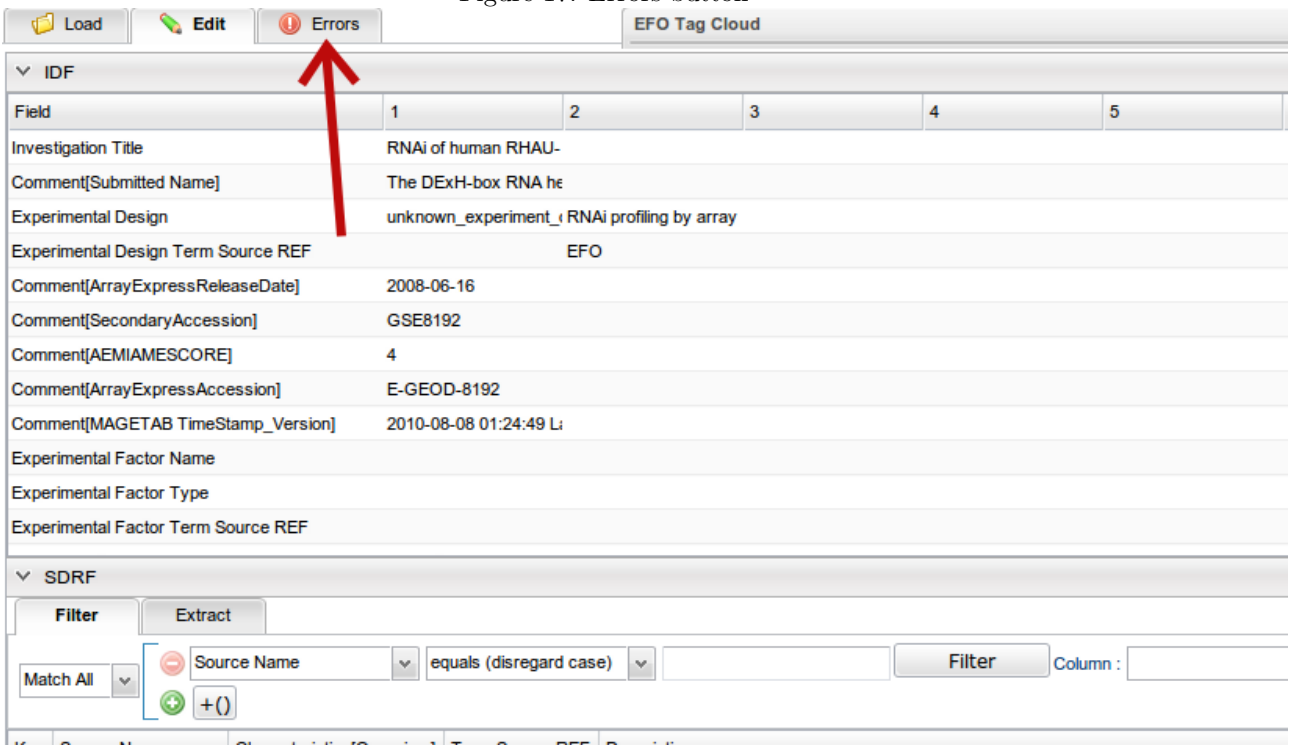


Figure 18: Errors tab

10 Exporting

The final stage in editing is exporting the changes. To save a file locally, the user can click either "Export" button as designated by Figure 19.

Figure 19: Export buttons