MageComet User Guide

Author: Vincent Xue

Last Updated: October 16, 2011

Contents

1 Introduction

1.1 What is MageComet?

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.

1.2 Where can I use MageComet?

MageComet is freely available for use at http://wwwdev.ebi.ac.uk/fgpt/magecomet. There is no need for installation as everything in this tutorial can be performed using any modern browser. The source code can be found at http://code.google.com/p/magecomet/

1.3 Getting Started

Users can get started by following the step by step tutorial below or by jumping to a specific component of interest.

2 Tutorial

2.1 Screencast

This tutorial is also available as a screencast at Here

2.2 Steps

3 Components

3.1 Loading Files

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

3.1.1 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.

To load the MAGE-TAB files to the server, the user must click the "Browse" button[1]. A popup will appear where the user can select either the SDRF file or the IDF file to upload first. The same "Browse" button must be clicked after the first file has loaded. When both files have successfully been loaded, the screen should look like Figure 2.

---

[1]The label may be "Choose File", depending on your browser
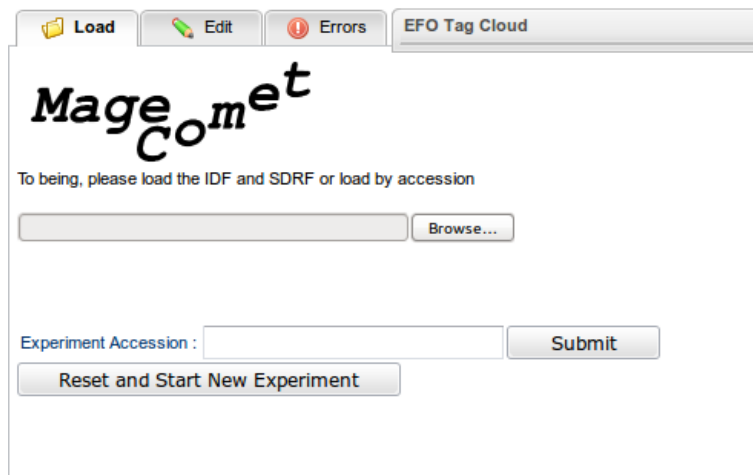
Figure 1: The opening load tab

Figure 2: Snapshot after loading IDF and SDRF files

3.1.2 ArrayExpress FTP

To load a file via the ArrayExpress FTP service the user can use the input box designated as "Experiment Accession". The user can simply type in the experiment accession and click submit, which will automatically fetch and load the IDF and SDRF files.

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.

3.2 Filtering and Replacing Data

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.

3.2.1 Filter

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

3

Figure 3: Edit after successful load

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

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

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

3.2.2 Replace

D Target Column - The column whose value will be set

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

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

3.3 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 columns in the SDRF 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. In addition, new columns also appear in this clipboard before they are placed into the SDRF.

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.
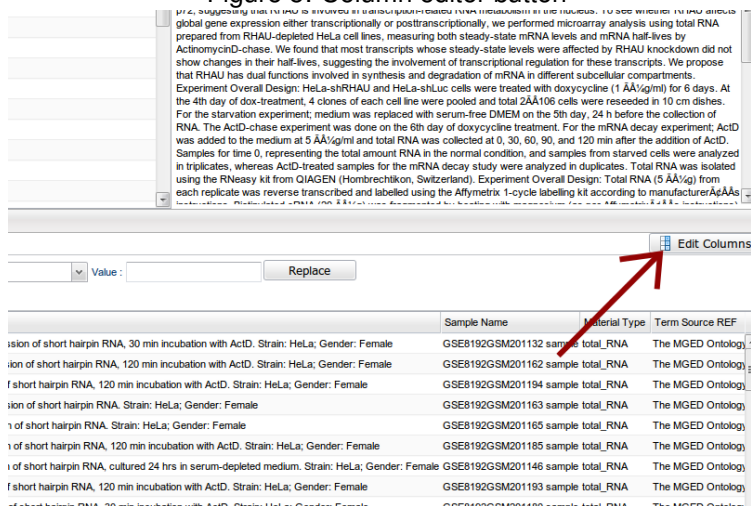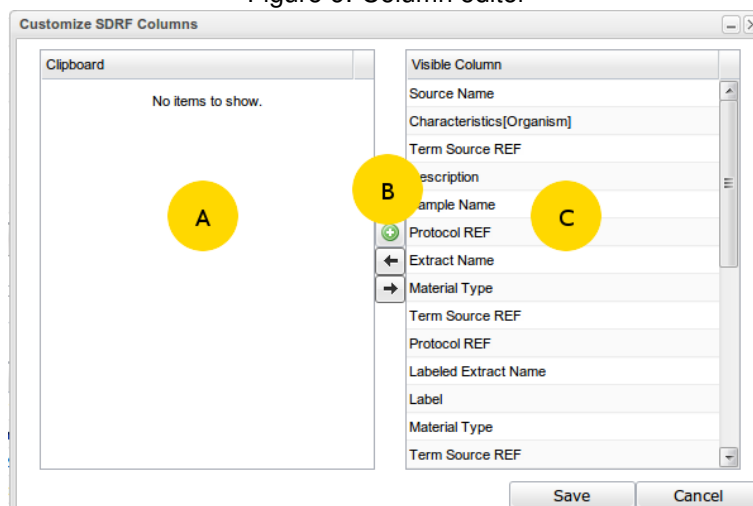
Figure 5: Column editor button

Figure 6: Column editor

3.3.1 Adding Columns

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

3.3.2 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.

3.3.3 Reordering Columns

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

3.3.4 Renaming Columns

To rename a column, double click the row.

3.4 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.

3.4.1 What type of columns can I extract from?

An example of splittable column is seen in Figure 7. This data comes from accession `E-GEOD-18781` before curation and can be found accompanying the documentation.

Figure 7: Example of a splittable column

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 between the rows, the extract feature implemented in Mage-Comet works well in these situations.

3.4.2 How do I use the extract tool?

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.

Figure 8: Extract

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 [ ].

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

3.4.3 Sample Extraction 1

Table 1 lists the input values for **A**, **B** and **C** that will extract the targeted value successfully for Figure 7.

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

Table 1: Sample inputs and outputs for Figure 7

3.4.4 Sample Extraction 2

Table 2 lists the input values for **A, B** and **C** that will extract the targeted value successfully for Figure 9. This data comes from accession `E-GEOD-8192`.

This dataset introduces the use of square brackets **[ ]**. If rows are delimited by a set of known values, these values can combined and treated as a single delimiter as shown. In this example, numbers are used as a set of delimeters, and the symbols **+** and **-** are used as another set of delimiters.

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

Description
HeLa shRHAU- 30min 1: This is the sample of HeLa-shRHAU cells without expression of short hairpin RNA, 30 min incubation with ActD. Strain: HeLa; Gender: Female
HeLa shRHAU+ 120min 2: This is the sample of HeLa-shRHAU cells with expression of short hairpin RNA, 120 min incubation with ActD. Strain: HeLa; Gender: Female
HeLa shLuc+ 120min 2: This is the sample of HeLa-shLuc cells with expression of short hairpin RNA, 120 min incubation with ActD. Strain: HeLa; Gender: Female
HeLa shRHAU- 0min 3: This is the sample of HeLa-shRHAU cells without expression of short hairpin RNA. Strain: HeLa; Gender: Female
HeLa shRHAU+ 0min 3: This is the sample of HeLa-shRHAU cells with expression of short hairpin RNA. Strain: HeLa; Gender: Female
HeLa shLuc- 120min 1: This is the sample of HeLa-shLuc cells without expression of short hairpin RNA, 120 min incubation with ActD. Strain: HeLa; Gender: Female
HeLa shRHAU+ STV 1: This is the sample of HeLa-shRHAU cells with expression of short hairpin RNA, cultured 24 hrs in serum-depleted medium. Strain: HeLa; Gender: Female

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

Table 2: Sample inputs and outputs for Figure 9
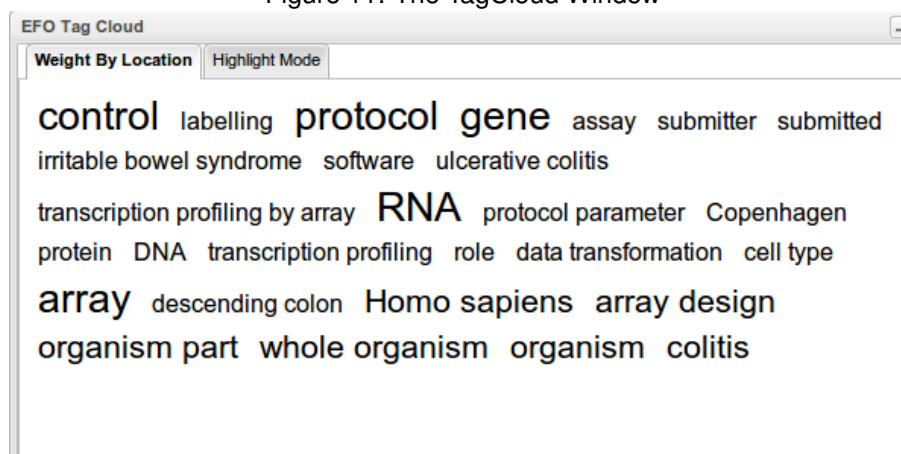
3.5 Tag Cloud

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.

Figure 10: Default tagcloud position

In the tagcloud representation, each tag represents an EFO ontology term that has been mined from the IDF and SDRF. The size of a tag corresponds to its origin. The small tag, indicates that the term is only used in the IDF; The medium sized tag indicates that the term is only used in the SDRF; and the largest tag indicates that term can be found in both IDF and SDRF.

Terms which originate from the IDF are weighted less than those from the SDRF because IDF documents are generally broader in description.

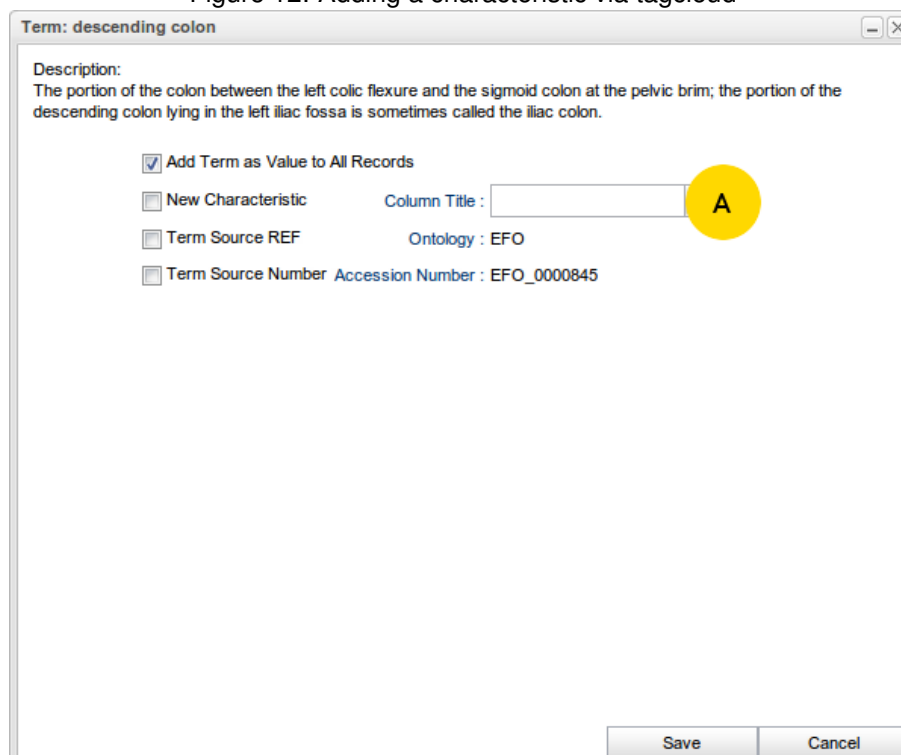Figure 11: The TagCloud Window

3.5.1 Adding characteristics to the SDRF 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 an important term is mentioned in the IDF document but is not referenced in the SDRF. By clicking on the term in the tag cloud, a new popup will give users the option of adding the selected term to all rows in the SDRF as a new column. Figure 12.

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

3.5.2 Highlighting text via tagcloud

Another sub feature of the tag cloud is the ability 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 now cause instances of the term to highlight on the page. This is demonstrated by Figure 13.

Figure 13: Tagcloud highlight feature

3.6　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
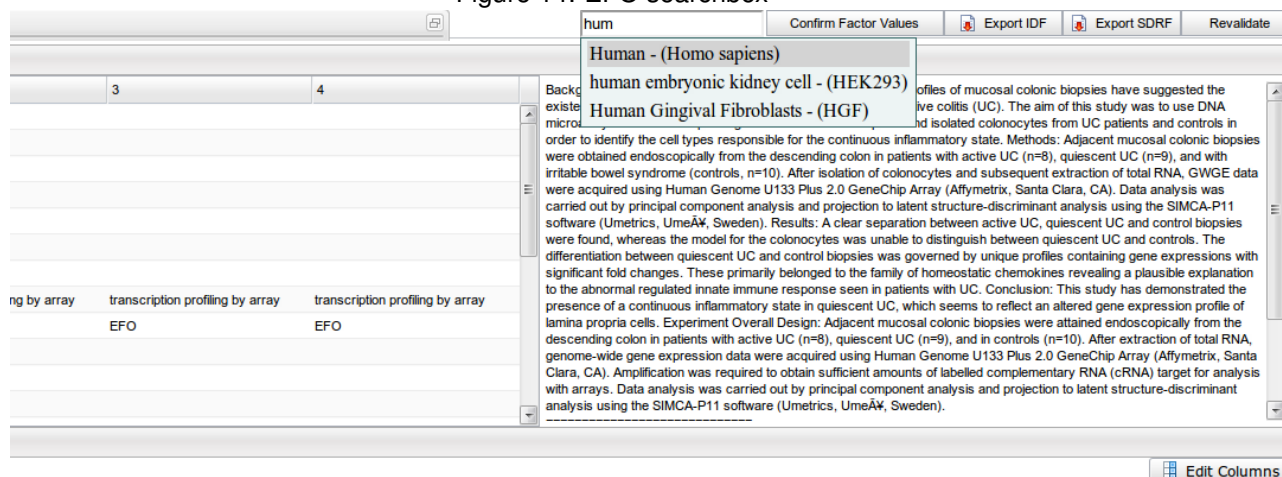
3.7　Adding Factor Values

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
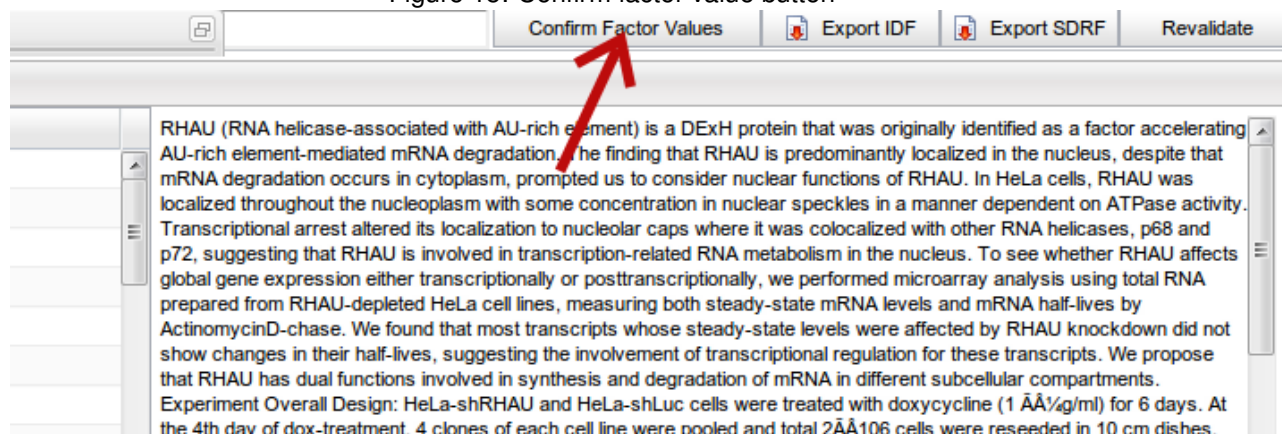
Figure 16: Confirm factor value window
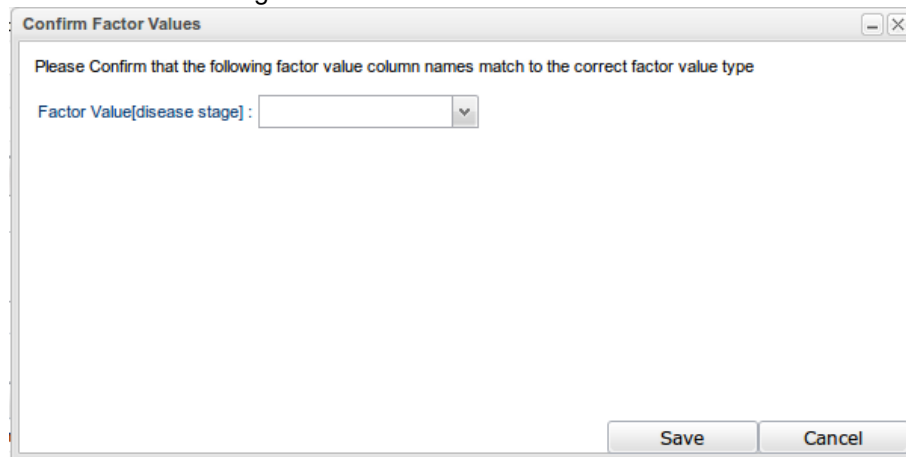
3.8 Revalidation/Validation

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 18)

Figure 17: Errors button
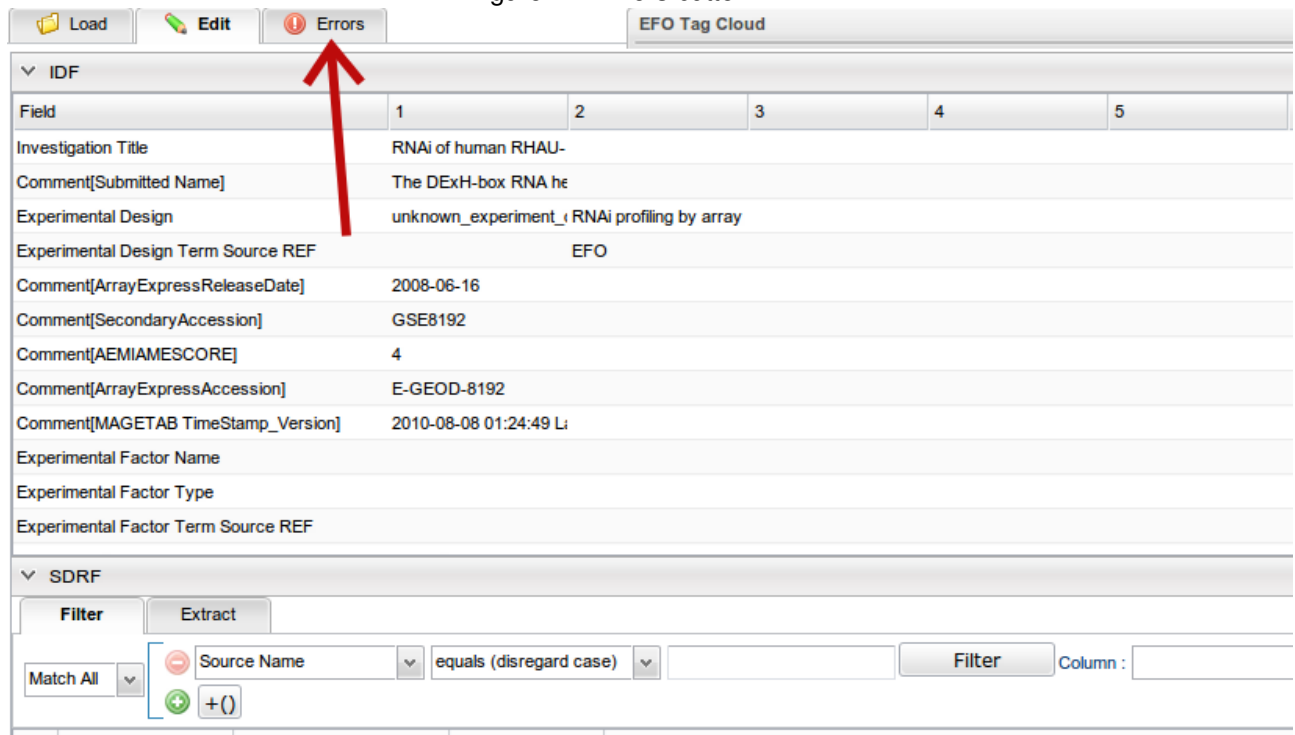
3.9 Exporting

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 18: Errors tab

Figure 18: Errors tab

Code	Type	Message	Comment	Line	Column
1015	validation warning	Missing info in IDF: 'the following '	IDF date tag Date Of Experiment i	29	-1
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF: Ter	49	2
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF: Ter	49	3
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF: Ter	49	4
21	validation warning	Duplicated column is present in I[	Duplicate information in IDF for Te	47	5
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF: Ter	49	5
21	validation warning	Duplicated column is present in I[	Duplicate information in IDF for Te	47	6
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF: Ter	49	6
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF for I	17	2
1015	validation warning	Missing info in IDF: 'the following '	Incomplete information in IDF for I	15	2

Figure 19: Export buttons

Figure 19: Export buttons

RHAU (RNA helicase-associated with AU-rich element) is a DExH protein that was originally identified as a factor accelerating AU-rich element-mediated mRNA degradation. The finding that RHAU is predominantly localized in the nucleus, despite that mRNA degradation occurs in cytoplasm, prompted us to consider nuclear functions of RHAU. In HeLa cells, RHAU was localized throughout the nucleoplasm with some concentration in nuclear speckles in a manner dependent on ATPase activity. Transcriptional arrest altered its localization to nucleolar caps where it was colocalized with other RNA helicases, p68 and p72, suggesting that RHAU is involved in transcription-related RNA metabolism in the nucleus. To see whether RHAU affects global gene expression either transcriptionally or posttranscriptionally, we performed microarray analysis using total RNA prepared from RHAU-depleted HeLa cell lines, measuring both steady-state mRNA levels and mRNA half-lives by ActinomycinD-chase. We found that most transcripts whose steady-state levels were affected by RHAU knockdown did not show changes in their half-lives, suggesting the involvement of transcriptional regulation for these transcripts. We propose that RHAU has dual functions involved in synthesis and degradation of mRNA in different subcellular compartments. Experiment Overall Design: HeLa-shRHAU and HeLa-shLuc cells were treated with doxycycline (1 Ã¼g/ml) for 6 days. At the 4th day of dox-treatment, 4 clones of each cell line were pooled and total 2ÃÂ106 cells were reseeded in 10 cm dishes. For the starvation experiment; medium was replaced with serum-free DMEM on the 5th day, 24 h before the collection of RNA. The ActD-chase experiment was done on the 6th day of doxycycline treatment. For the mRNA decay experiment; ActD was added to the medium at 5 Ã¼g/ml and total RNA was collected at 0, 30, 60, 90, and 120 min after the addition of ActD. Samples for time 0, representing the total amount RNA in the normal condition, and samples from starved cells were analyzed in triplicates, whereas ActD-treated samples for the mRNA decay study were analyzed in duplicates. Total RNA was isolated using the RNeasy kit from QIAGEN (Hombrechtikon, Switzerland). Experiment Overall Design: Total RNA (5 Ã¼g) from each replicate was reverse transcribed and labelled using the Affymetrix 1-cycle labelling kit according to manufacturerÃ¢ÂÂs