SOFTWARE TOOL ARTICLE

Improve your Galaxy text life: The Query Tabular Tool [version 1; peer review: 1 approved, 2 approved with reservations]

James E. Johnson[1], Praveen Kumar[2,3], Caleb Easterly[2], Mark Esler[4], Subina Mehta[2], Arthur C. Eschenlauer[2,4], Adrian D. Hegeman[4], Pratik D. Jagtap[2], Timothy J. Griffin[2]

[1]Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN, 55455, USA
[2]Department of Biochemistry, Molecular Biology and Biophysics, University of Minnesota, Minneapolis, Minnesota, 55455, USA
[3]Bioinformatics and Computational Biology Program, University of Minnesota-Rochester, Rochester, MN, 55904, USA
[4]Department of Horticulture, University of Minnesota, St. Paul, MN, 55108, USA

v1 **First published:** 05 Oct 2018, **7**:1604
https://doi.org/10.12688/f1000research.16450.1
Latest published: 09 Jan 2019, **7**:1604
https://doi.org/10.12688/f1000research.16450.2

Abstract

Galaxy provides an accessible platform where multi-step data analysis workflows integrating disparate software can be run, even by researchers with limited programming expertise. Applications of such sophisticated workflows are many, including those which integrate software from different 'omic domains (e.g. genomics, proteomics, metabolomics). In these complex workflows, intermediate outputs are often generated as tabular text files, which must be transformed into customized formats which are compatible with the next software tools in the pipeline. Consequently, many text manipulation steps are added to an already complex workflow, overly complicating the process and decreasing usability, especially for non-expert bench researchers focused on obtaining results. In some cases, limitations to existing text manipulation are such that desired analyses can only be carried out using highly sophisticated processing steps beyond the reach of most users. As a solution, we have developed the Query Tabular Galaxy tool, which leverages a SQLite database generated from tabular input data. This database can be queried and manipulated to produce transformed and customized tabular outputs compatible with downstream processing steps. Regular expressions can also be utilized for even more sophisticated manipulations, such as find and replace and other filtering actions. Using several Galaxy-based multi-omic workflows as an example, we demonstrate how the Query Tabular tool dramatically streamlines and simplifies the creation of multi-step analyses, efficiently enabling complicated textual manipulations and processing. This tool should find broad utility for users of the Galaxy platform seeking to develop and use sophisticated workflows involving text manipulation on tabular outputs.

Open Peer Review

Reviewer Status ✓ ✓ ✓

	Invited Reviewers		
	1	**2**	**3**
version 2 (revision) 09 Jan 2019	✓ report		✓ report
version 1 05 Oct 2018	? report	✓ report	? report

1. **Daniel Blankenberg**, Cleveland Clinic, Cleveland, USA
2. **Maria A. Doyle**, Peter MacCallum Cancer Centre, Melbourne, Australia
3. **Margaret E. Staton**, University of Tennessee , Knoxville, USA

Any reports and responses or comments on the article can be found at the end of the article.

Keywords
Galaxy, Workflows, SQLite, Multi-omics, Genomics, Proteomics, Metaproteomics, Proteogenomics, Metabolomics

This article is included in the Galaxy gateway.

Corresponding authors: James E. Johnson (johns198@umn.edu), Timothy J. Griffin (tgriffin@umn.edu)

Author roles: Johnson JE: Conceptualization, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Kumar P**: Methodology, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Easterly C**: Methodology, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Esler M**: Methodology, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Mehta S**: Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Eschenlauer AC**: Methodology, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Hegeman AD**: Funding Acquisition, Supervision, Writing – Review & Editing; **Jagtap PD**: Methodology, Project Administration, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Griffin TJ**: Funding Acquisition, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported in part by NSF award 1458524 and NIH award U24CA199347 to T.J. Griffin and the Galaxy for proteomics (Galaxy-P) research team.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Johnson JE, Kumar P, Easterly C *et al.* **Improve your Galaxy text life: The Query Tabular Tool [version 1; peer review: 1 approved, 2 approved with reservations]** F1000Research 2018, **7**:1604 https://doi.org/10.12688/f1000research.16450.1

First published: 05 Oct 2018, **7**:1604 https://doi.org/10.12688/f1000research.16450.1

Introduction

The Galaxy platform[1] offers a highly flexible bioinformatics workbench in which disparate software tools can be deployed and integrated into sophisticated workflows. Frequently, these workflows contain many steps and different software tools, with many different types of outputs. Each output can then act as the input for a subsequent software tool. Often, the results outputted from a software tool are in the form of a tabular file, which serve as input to a subsequent tool in the workflow. To make these workflows functional, usually the tabular output(s) must be manipulated, extracting and re-formatting the original file and creating a new tabular file with a data structure which can be read by a downstream software tool. In some cases, the final tabular results file from the workflow must be further processed and manipulated to obtain desired information for interpretation by the user.

There are many examples of multi-step workflows requiring manipulations of tabular text files employed across the diverse analysis applications facilitated by Galaxy. One example is emerging "multi-omic" analyses, which integrate software from different 'omic domains and are well suited to the strengths of Galaxy[2]. For example, proteogenomics integrates tools for RNA-Seq assembly and analysis, software for matching tandem mass spectrometry (MS/MS) data to peptide and protein sequences, and other customized tools to characterize novel, variant protein sequences expressed within a sample[3,4]. To enable compatibility between the software tools composing a proteogenomics workflow, tabular files often must be manipulated into appropriate formats recognized by specific tools. Another example is Galaxy workflows for metaproteomics[5,6], a multi-omics analysis which requires text manipulations in workflows integrating metagenomic, MS-based proteomics and other functional and taxonomic software tools. Finally, Galaxy-based metabolomics data analysis solutions are also emerging[7–9], which utilize tabular inputs and outputs within multiple step workflows.

Under the category of "Text Manipulation", the Galaxy Tool Shed has long offered many tools for extracting and transforming information within tabular files produced in workflows. However, sophisticated workflows (e.g. multi-omics, metabolomics), can require numerous manipulations to tabular files in order to build fully integrated and automated pipelines. Consequently, workflows can grow to hundreds of steps, dominated by sequential text manipulation steps. This situation makes the building and optimizing of such workflows highly time-consuming and prone to errors, requiring much effort even by experienced Galaxy users. It also hampers efforts to further customize or modify workflows by other users, if these change formats of the tabular files, necessitating another round of optimization of many text manipulations.

To improve the available options for text manipulation in Galaxy, we have developed a new suite of tools, which we generally refer to as Query Tabular. Query Tabular leverages the power of SQLite, automatically creating a database directly from desired tabular outputs within a workflow using the Query Tabular tool. The SQLite database can be saved to the Galaxy history, and acted upon by the companion SQLite_to_Tabular tool, generating additional tabular outputs containing desired information and formatting. As such, Query Tabular streamlines complicated text manipulations, greatly simplifying the creation and customization of Galaxy workflows, and in some cases enabling new analyses. Here, we show the use of Query Tabular in several example Galaxy-based workflows, demonstrating its value. Query Tabular is available through the Galaxy Tool Shed and should prove highly useful to a broad community of Galaxy users.

Methods

Implementation

The Query Tabular tools use Python applications to read and filter tabular files, and the Python package sqlite to create and query a SQLite database. There are 3 main functions performed:

1. Line filtering. For a tabular file, a sequence of line filters can be used to transform each line as it is read. A line filter takes one TAB-separated line and produces 0 or more TAB-separated lines. For example, a line filter that filters out comment lines only produces an output line when an input line does not begin with a comment character. The normalization line filter splits a line that has a comma-separated value in one (or more) specified fields into one output row per list item.

2. Loading a SQLite table. The filtered tabular file is inspected for number of TAB-separated fields and the SQLite type of the values in each field - Real, Integer, or Text - followed by generation of a database table for that file. Each line from the filtered tabular file is then loaded as a row in that table.

3. Querying the database. A SQL query is executed on the database. The results are written out as a new tabular-formatted text file.

The query_tabular.py application can perform all three of the steps above. However, the query can be omitted when the SQLite database is the only desired output. The sqlite_to_tabular.py application only performs the query function given an existing SQLite database as input. This can be useful when one needs to perform several queries on the same database. The filter_tabular.py application performs the line filtering function to directly produce a tabular file. This can be sufficient for simple selection of rows and columns from a single file.

Galaxy tools have been developed for each of the actions described above, and are called "Query Tabular", "SQLite to Tabular", and "Filter Tabular", respectively. These Galaxy tools provide a web form for a user to specify input files and settings for line filters, table and column names, and a SQL query. The Galaxy framework makes it easy to link these tools with other software and processing steps, creating multi-step workflows.

Operation

Figure 1 shows a screenshot of the Galaxy-based Query Tabular tool. The Query Tabular Galaxy tool loads any number of tabular datasets into a new or existing SQLite database

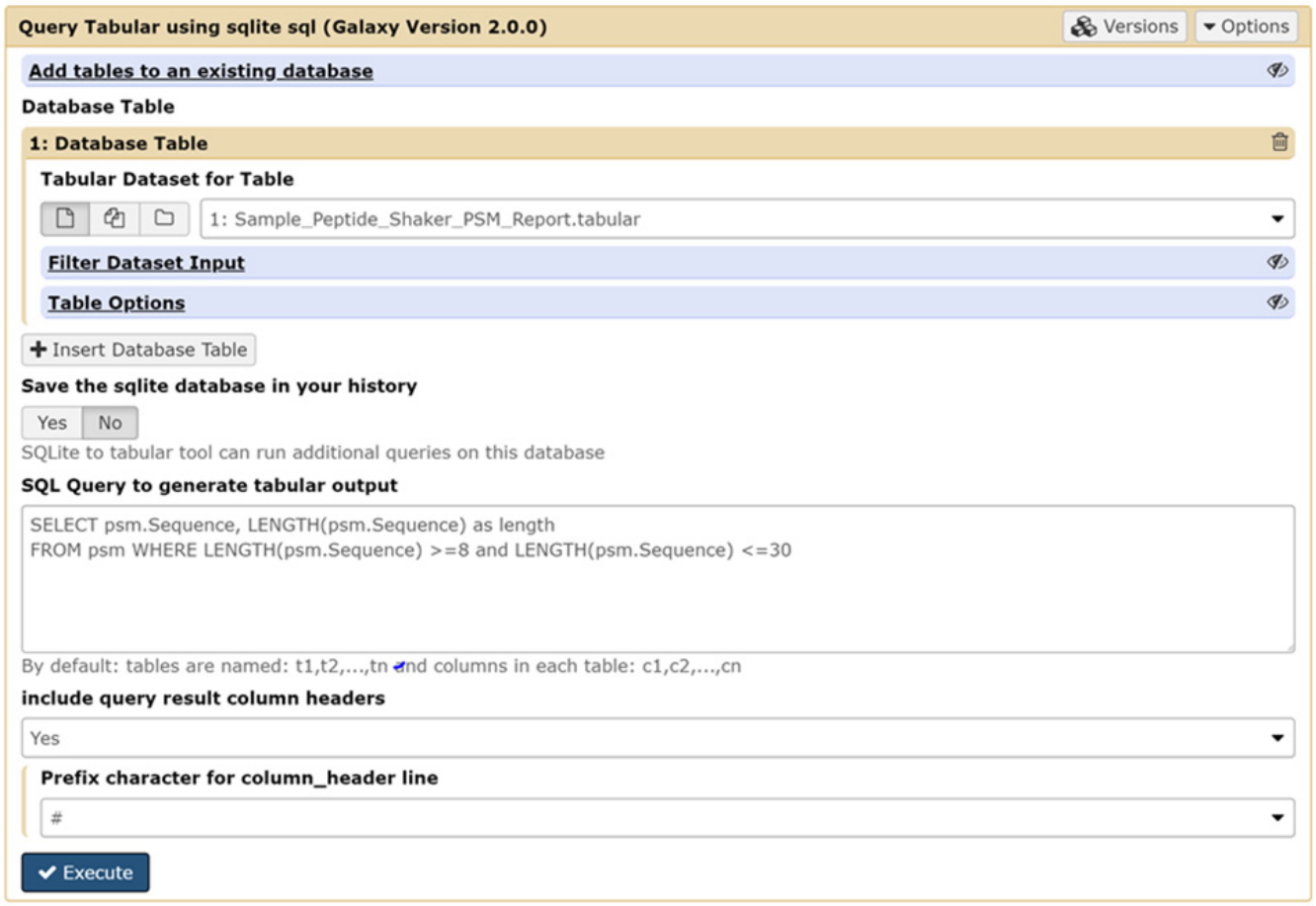

Figure 1. The Galaxy user interface for the Query Tabular suite of tools. The user can select the tabular data which acts as input and is converted to a SQLite database. The input data tables can be filtered if desired. The interface also provides a field to define the queries for the SQLite database that will be carried out, along with options for displaying results from the query in the tabular output.

allowing the full power of a SQL query to produce a new tabular output. Long, complicated workflows of Galaxy text manipulation tools can be replaced by Query Tabular in a single step.

The Query Tabular tool provides default names for tables - t1, t2, etc. - and columns - c1, c2, etc. - but a user can specify more specific and meaningful names for tables and columns. When column names are specified in the first row of the tabular file, the user has the option to use those names when selecting the columns to be loaded into the SQLite database.

Regex functions, which apply regular expressions, are added to sqlite connections so that re.search, re.match, and re.replace functions are available for use in the SQL query. Line filters can apply regular expressions while reading tabular input files to include, exclude, or modify lines before entering the values as rows in the database table. A column replace line filter can use a regex function to change, for example, a date value to the SQLite recognized format. A normalize filter can convert list fields in the input to first normal form with an individual list item per row; when several fields are specified in a normalization filter, an input line having lists of length n in the specified columns results in n output row, each with one respective pair of values from the specified fields.

Use cases

Below we provide examples of use cases for Query Tabular, focusing on Galaxy-based workflows for proteogenomics, metaproteomics and metabolomics.

Proteogenomics

A common task in a proteogenomics data analysis is to match MS/MS fragmentation spectra to variant peptide sequences, which derive from genomic mutations, expression from genomic regions thought to be non-coding or silenced, or unexpected RNA splicing events[10]. The veracity of putative variant sequences matched to MS/MS spectra must be confirmed, which can be accomplished by querying the variant peptide sequences against NCBI's non-redundant (nr) protein database using the BLASTP

tool, which is implemented in Galaxy[4]. Those peptides which do not have a 100% alignment and sequence match to known sequences within the database qualify as verified variant sequences, which are then passed on for further analysis[3,4].

The workflow for carrying out this analysis of putative variant peptide sequences is shown in Figure 2. This workflow takes as input the peptide spectrum matches (PSMs) containing matches to putative variant amino acid sequences, and analyzes these using BLASTP, producing a list of verified PSMs to true variant sequences. Figure 2A outlines the initial workflow, which contained 9 total steps and required multiple text manipulations with Galaxy tools. The text manipulations format the input tabular file for BLASTP analysis, extracting and re-formatting information from the PSM input. A number of manipulations are also required on the BLASTP alignments: querying the tabular files for peptides with alignment identities less than 100%, those with any gaps in the sequence alignment or those which lacked full-length matching of the known peptides to the putative variant sequence.

When Query Tabular is used, the individual text manipulation steps are not needed, and the number of steps is reduced from 9 to 4 (Figure 2B). We have made this workflow available for demonstration purposes at z.umn.edu/proteogenomicsgateway. Supplementary File 1 provides instructions on accessing and using this workflow.

Metaproteomics

Metaproteomic workflows seek to identify peptide sequences expressed by a community of microorganisms, usually bacteria. These sequences are further analyzed to characterize the taxonomic distributions of the bacteria present in the community; the peptides are also mapped to protein groups which have known biochemical functions, such that the peptides can be indicators of specific functional responses of the community to external perturbations[11,12].

In one established metaproteomics Galaxy workflow[6], the microbial peptides must be verified by matching to the NCBI nr database, using the BLASTP tool (Figure 3). A number of text manipulation steps are required to make the file of identified peptide sequences compatible with BLASTP. The BLASTP-aligned sequences are outputted in a tabular file, and this file must be further manipulated via several steps in order to create a tabular file in correct format for downstream functional and taxonomic analysis. In all, this workflow ends up requiring many text manipulations in order to achieve desired results. Figure 3A highlights these numerous manipulation steps.

Query Tabular greatly simplifies this metaproteomics workflow. As shown in Figure 3B, use of Query Tabular eliminates many of the initial steps required to generate a tabular input compatible with BLASTP. It also greatly simplifies the second part of the workflow where the BLASTP outputs are further manipulated

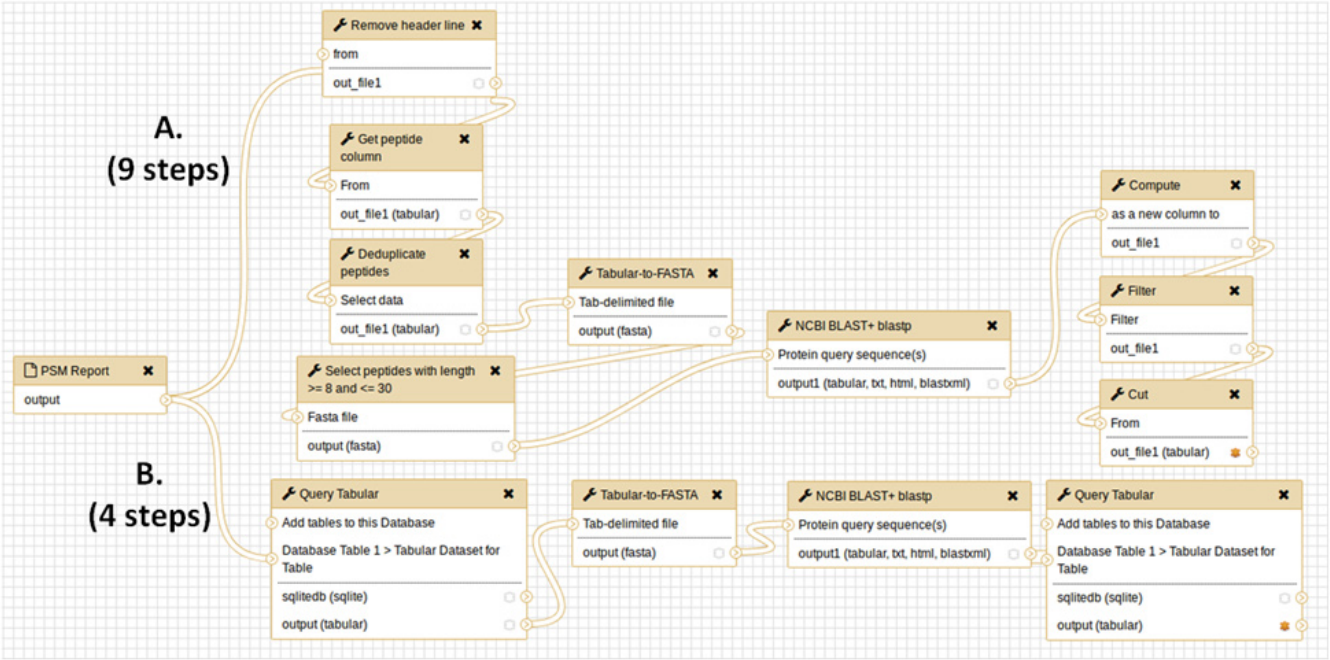

Figure 2. A proteogenomics workflow using Query Tabular. This workflow takes as input peptide spectrum matches (PSMs) of putative variant peptide sequences and further analyzes them using BLASTP to verify sequences which are truly variants compared to the reference proteome. **A**) The initial workflow comprised of nine total steps, including multiple text manipulation steps in Galaxy; **B**) The simplified workflow when using Query Tabular, which reduces the number of steps to 4 to obtain the same results.

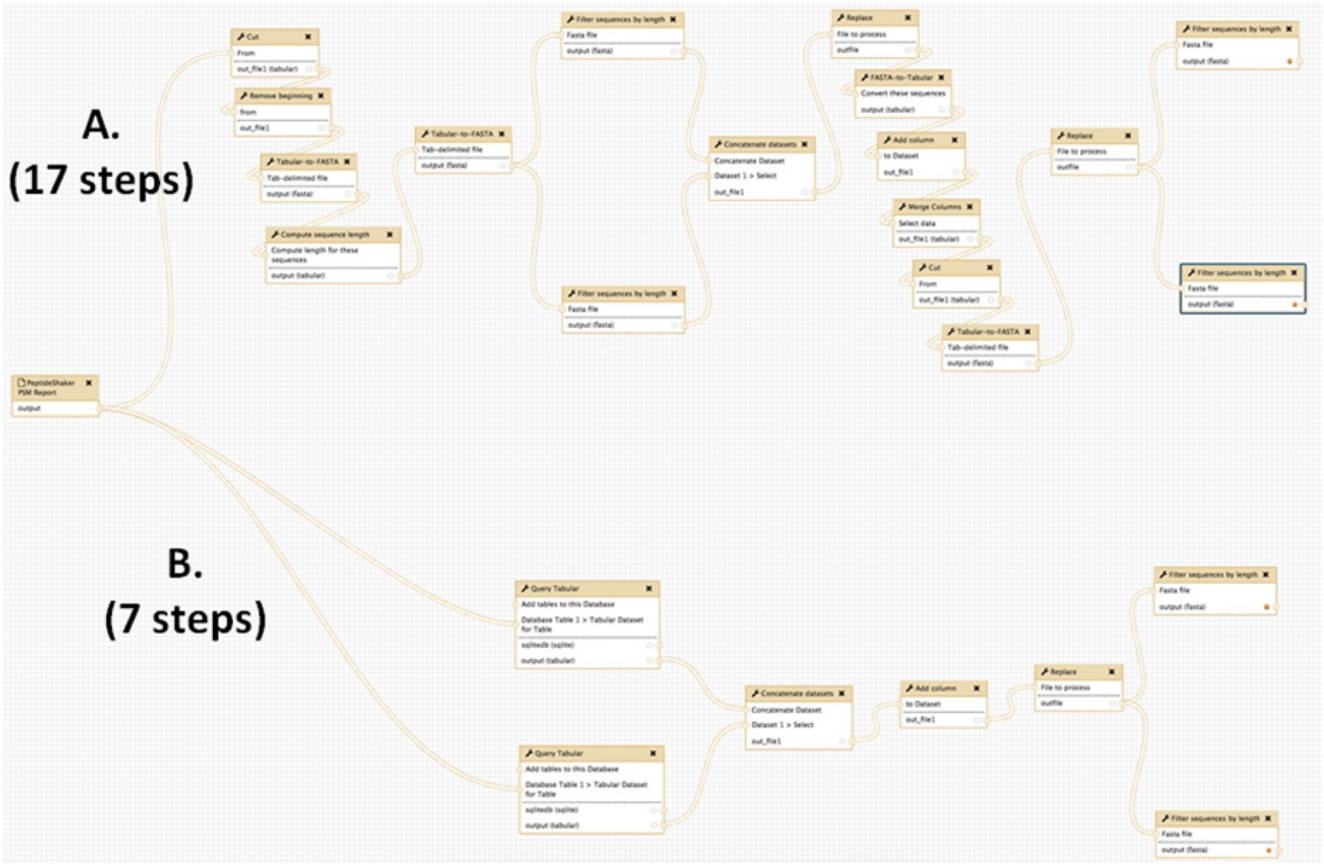

Figure 3. A metaproteomics workflow using Query Tabular. This workflow verifies the presence of detected microbial peptides by matching peptides against the NCBI nr protein sequence database using the BLASTP tool. (**A**) Using conventional Galaxy text manipulation tools, the workflow requires 17 steps to achieve desired outputs. (**B**) When utilizing Query Tabular, desired results are obtained in seven steps.

to generate a tabular file which is required for further taxonomic and functional analysis. In all, using Query Tabular reduced the length of the workflow from 17 steps to 7. We have made this workflow available for demonstration purposes at z.umn.edu/metaproteomicsgateway. Supplementary File 1 provides instructions on accessing and using this workflow.

Metabolomics

A Galaxy-based metabolomics workflow provides an example where Query Tabular was used to enable efficient data correction and analysis that was not possible with other existing Galaxy tools. This workflow utilizes VKMZ, a metabolomics tool under development which predicts and plots metabolites from liquid chromatography (LC)-MS data. Metabolite predictions are made by comparing the neutral mass of observed signals to a dictionary of known mass-formulas. When a signal's neutral mass is within a given mass error range of a known mass, a prediction is made.

For the use-case presented here, targeted metabolomics data were collected on a low resolution LC-MS instrument. Low mass standards in the data, used to provide more accurate mass assignments to observed signals, had a systematic mass shift caused by using an instrument calibration method for high mass molecules. Figure 4 shows the two-part SQL query inputted in the Query Tabular tool and used to correct this shift, operating on the tabular data generated from MS data by VKMZ, which assumes charge (z) is 1. The inner-query determines the average relative mass error for molecules with low mass-to-charge (mz) values (molecular mass <250 Daltons) in the data. The outer-query adjusts all detected molecules within this same mz range by the average mass error. Before making mass adjustment with Query Tabular, VKMZ was able to predict 85.7% of the features for the standards. After the mass adjustment, VKMZ was able to correctly predict all features for the standards. This two-step manipulation, with dependency of the outer-query on the result from the inner-query, is concise and would require generation of a nested, multiple step workflow within the larger workflow if using existing text manipulation tools in Galaxy. We have made this workflow available for demonstration purposes at z.umn.edu/metaproteomicsgateway. Supplementary File 1 provides instructions on accessing and using this workflow.

Conclusions

We have described a new Galaxy tool, Query Tabular, which significantly improves the development and application of multi-step workflows in Galaxy. Leveraging a SQLite database,

```
1  SELECT sample_id,
2    polarity,
3    mz * (1 + avg_rel_err) AS mz,
4    rt,
5    intensity,
6    mz AS raw_mz
7  FROM t1,
8    (
9      SELECT avg(predicted_delta / predicted_mass) AS avg_rel_err
10     FROM t1
11     WHERE mz < 250
12   ) AS t2
```

Figure 4. Example query utilizing the Query Tabular tool for a metabolomics data analysis workflow. The two part SQL query corrects mass errors in low resolution MS-based metabolomics data, using an inner- and outer-query. The inner-query (lines 9-11) determine the average mass error for mz values of detected molecules below 250 Daltons. The outer-query (all other lines) adjusts all mz values in this range based on the determined mass error. Chromatographic retention time (rt) and signal intensities are also assigned values for the molecules detected by LC-MS.

and utilizing regular expressions, the tool can minimize the need for lengthy workflows using conventional Galaxy-based text manipulation tools. This eases the process of workflow development, producing more efficient workflows which can be utilized and understood more easily by non-expert bench researchers. We have provided use-case examples in the area of multi-omics (proteogenomics and metaproteomics) demonstrating the value of Query Tabular in this way. Via an example in metabolomics, we also demonstrate how Query Tabular can enable new manipulations and analyses of textual data within a single, simplified workflow, that would otherwise require separate workflow development if attempted using existing Galaxy tools. The Query Tabular tool has also proven useful and versatile for developing workflows used for multi-omic informatic training workshops (http://galaxyp.org/workshops/) and online training via the Galaxy Training Network (http://galaxyproject.github.io/training-material[13]). The free and open tool is available to any Galaxy user, and should provide a valuable addition to the Galaxy tool box for developing analysis workflows.

Data availability

All data underlying the results are available as part of the article and no additional source data are required.

Software availability

The Query Tabular suite of tools can be added to a Galaxy server from the Galaxy Tool Shed: https://toolshed.g2.bx.psu.edu/view/iuc/query_tabular/1ea4e668bf73.

Source code available from: https://github.com/galaxyproject/tools-iuc/tree/master/tools/query_tabular.

Archived source code at time of publication: https://doi.org/10.5281/zenodo.1439296[14].

License: MIT license.

Adding tools from the Tool Shed is an administrative function of a Galaxy server, and as a security precaution is restricted to users designated as admins for the server. From the Galaxy server, an admin simply searches for the tool in the toolshed and clicks the install button.

As we described above, we have also made available example workflows for demonstration purposes using Query Tabular on outputs from proteogenomics data (z.umn.edu/proteogenomicsgateway) and metaproteomics & metabolomics data (z.umn.edu/metaproteomicsgateway). Supplementary File 1 contains instructions on how to access these example workflows.

Grant information

This work was supported in part by NSF award 1458524 and NIH award U24CA199347 to T.J. Griffin and the Galaxy for proteomics (Galaxy-P) research team.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

We thank the Supercomputing Institute at the University of Minnesota for support and maintenance of software and hardware infrastructure used in the development of this tool. We also thank the Jetstream team at the University of Indiana for support and maintenance of software and hardware infrastructure used for hosting publicly accessible Galaxy instances described in this manuscript. Additionally, we would like to thank Kevin Murray from the Department of Veterinary Population Medicine at the University of Minnesota for contributing the metabolomics data set.

Supplementary material

Supplementary File 1. Detailed instructions on accessing and operating the demonstration workflows which utilize Query Tabular.

Click here to access the data.

References

1. Afgan E, Baker D, Batut B, *et al.*: **The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update.** *Nucleic Acids Res.* 2018; **46**(W1): W537–W44.
PubMed Abstract | Publisher Full Text | Free Full Text
2. Boekel J, Chilton JM, Cooke IR, *et al.*: **Multi-omic data analysis using Galaxy.** *Nat Biotechnol.* 2015; **33**(2): 137–9.
PubMed Abstract | Publisher Full Text
3. Chambers MC, Jagtap PD, Johnson JE, *et al.*: **An Accessible Proteogenomics Informatics Resource for Cancer Researchers.** *Cancer Res.* 2017; **77**(21): e43–e46.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Jagtap PD, Johnson JE, Onsongo G, *et al.*: **Flexible and accessible workflows for improved proteogenomic analysis using the Galaxy framework.** *J Proteome Res.* 2014; **13**(12): 5898–908.
PubMed Abstract | Publisher Full Text | Free Full Text
5. Blank C, Easterly C, Gruening B, *et al.*: **Disseminating Metaproteomic Informatics Capabilities and Knowledge Using the Galaxy-P Framework.** *Proteomes.* 2018; **6**(1): pii: E7.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Jagtap PD, Blakely A, Murray K, *et al.*: **Metaproteomic analysis using the Galaxy framework.** *Proteomics.* 2015; **15**(20): 3553–65.
PubMed Abstract | Publisher Full Text
7. Davidson RL, Weber RJ, Liu H, *et al.*: **Galaxy-M: a Galaxy workflow for processing and analyzing direct infusion and liquid chromatography mass spectrometry-based metabolomics data.** *GigaScience.* 2016; **5**: 10.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Guitton Y, Tremblay-Franco M, Le Corguillé G, *et al.*: **Create, run, share, publish, and reference your LC-MS, FIA-MS, GC-MS, and NMR data analysis workflows with the Workflow4Metabolomics 3.0 Galaxy online infrastructure for metabolomics.** *Int J Biochem Cell Biol.* 2017; **93**: 89–101.
PubMed Abstract | Publisher Full Text
9. Weber RJM, Lawson TN, Salek RM, *et al.*: **Computational tools and workflows in metabolomics: An international survey highlights the opportunity for harmonisation through Galaxy.** *Metabolomics.* 2017; **13**(2): 12.
PubMed Abstract | Publisher Full Text | Free Full Text
10. Nesvizhskii AI: **Proteogenomics: concepts, applications and computational strategies.** *Nat Methods.* 2014; **11**(11): 1114–25.
PubMed Abstract | Publisher Full Text | Free Full Text
11. Hettich RL, Pan C, Chourey K, *et al.*: ***Metaproteomics*: harnessing the power of high performance mass spectrometry to identify the suite of proteins that control metabolic activities in microbial communities.** *Anal Chem.* 2013; **85**(9): 4203–14.
PubMed Abstract | Publisher Full Text | Free Full Text
12. Rudney JD, Jagtap PD, Reilly CS, *et al.*: **Protein relative abundance patterns associated with sucrose-induced dysbiosis are conserved across taxonomically diverse oral microcosm biofilm models of dental caries.** *Microbiome.* 2015; **3**: 69.
PubMed Abstract | Publisher Full Text | Free Full Text
13. Batut B, Hiltemann S, Bagnacani A, *et al.*: **Community-Driven Data Analysis Training for Biology.** *Cell Syst.* 2018; **6**(6): 752–758.e1.
PubMed Abstract | Publisher Full Text
14. Johnson JE: **query_tabular (Version 3.0.0).** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1439296

Open Peer Review

Current Peer Review Status: ? ✔ ?

Version 1

Reviewer Report 19 November 2018

https://doi.org/10.5256/f1000research.17980.r39108

?

Margaret E. Staton
Department of Entomology and Plant Pathology, Institute of Agriculture, University of Tennessee , Knoxville, TN, USA

Johnson *et al.* present a new Galaxy tool, Query Tabular, that enables a Galaxy user to load a tab-separated value (tsv) file and then make SQL-based manipulations of that data. The tool leverages a sqlite database and is publicly available.

Is the rationale for developing the new software tool clearly explained?

Yes. The authors created this tool to make sophisticated text manipulations possible within Galaxy, enabling workflows with many fewer steps than previously possible. However, the rationale that this will be used by the “non-expert bench researcher” is weak - SQL is a computational language that is not a common skill among bench researchers. Galaxy itself was built to help researchers who do not have in depth computational skills to still be able to run sophisticated informatics analysis.

However, I think the tool may be used by a slightly different group of Galaxy users - Galaxy server administrators or informaticians who build and maintain workflows for others, or the tool developers that are interested in embedding their own or others' tools into useful workflows. This is a slightly different user group than the average user, but very important for making Galaxy powerful. Pointing this out in the manuscript would provide a more compelling reason for the tool and is more in line with the use cases.

Is the description of the software tool technically sound?

1. The implementation section is short but covers the three basic functions.

2. In the methods, the original list of functions is in the order line filtering, loading the table, then querying. The next paragraphs discuss the functions in the reverse order, making it

confusing.

3. Details on filtering the input file are scant; it would be nice to have a figure that illustrates the user interface for that part of the tool. This function appears as an unexpanded box in Figure 1.

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

1. The tool is publicly available through the Galaxy tool shed, which is the central place for Galaxy tools.

2. The instructions for the tools inside Galaxy are good and helpful for figuring out how to use them. I was able to use the provided Jetstream instance to test “Query Tabular”, “SQLite to Tabular”, and “Filter Tabular”, all worked with my very simple tests. I do not think it’s feasible for the jetstream instance to persist indefinitely - could the workflows be shared via the main public Galaxy instance instead to help future readers?

3. A README for the github repo would be helpful for other developers.

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Is there any way for a user to see what database tables are in their history and how they are structured (i.e. column names and data types in each column?). It would be difficult to debug why an SQL query is not working if it’s impossible to see the table and its data somehow (I ran into this problem trying the tool out). For example, someone who works with an SQL database from the command line would use sqlcmd, or through a web server, something like PhpLiteAdmin. Something like that would be very helpful inside the Galaxy interface.

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 19 Dec 2018

Tim Griffin, University of Minnesota, Minneapolis, USA

We thank the reviewer for the comments. Below in bold text we provide responses to these comments and revisions we have made in the updated version.

Johnson et al. present a new Galaxy tool, Query Tabular, that enables a Galaxy user to load a tab-separated value (tsv) file and then make SQL-based manipulations of that data. The tool leverages a sqlite database and is publicly available.

Is the rationale for developing the new software tool clearly explained?

Yes. The authors created this tool to make sophisticated text manipulations possible within Galaxy, enabling workflows with many fewer steps than previously possible. However, the rationale that this will be used by the “non-expert bench researcher” is weak - SQL is a computational language that is not a common skill among bench researchers. Galaxy itself was built to help researchers who do not have in depth computational skills to still be able to run sophisticated informatics analysis.
However, I think the tool may be used by a slightly different group of Galaxy users - Galaxy server administrators or informaticians who build and maintain workflows for others, or the tool developers that are interested in embedding their own or others' tools into useful workflows. This is a slightly different user group than the average user, but very important for making Galaxy powerful. Pointing this out in the manuscript would provide a more compelling reason for the tool and is more in line with the use cases.

>We agree with this comment, and we have modified our description of the target audience as those who are more advanced Galaxy users and developers, with knowledge of SQL (see comments for reviewers above). We have removed the mention of “non-expert bench researchers” as being the main beneficiaries of this tool from the Conclusions section.

Is the description of the software tool technically sound?

The implementation section is short but covers the three basic functions.

In the methods, the original list of functions is in the order line filtering, loading the table, then querying. The next paragraphs discuss the functions in the reverse order, making it confusing.

Details on filtering the input file are scant; it would be nice to have a figure that illustrates the user interface for that part of the tool. This function appears as an unexpanded box in Figure 1.

>We have expanded Figure 1 to show the view of the interface when a user selects functions in the main tool (Table options and Filtering options).

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

The tool is publicly available through the Galaxy tool shed, which is the central place for Galaxy tools.

The instructions for the tools inside Galaxy are good and helpful for figuring out how to use them. I was able to use the provided Jetstream instance to test “Query Tabular”, “SQLite to Tabular”, and “Filter Tabular”, all worked with my very simple tests. I do not think it’s feasible for the jetstream instance to persist indefinitely - could the workflows be shared via the main public Galaxy instance instead to help future readers?
A README for the github repo would be helpful for other developers.

>As we have mentioned for the reviewer comments above, we have now deposited the workflows and input data in a Github repository:
https://github.com/galaxyproteomics/query_tabular_supplementary_material
This repository also contains a README file as suggested.

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Is there any way for a user to see what database tables are in their history and how they are structured (i.e. column names and data types in each column?). It would be difficult to debug why an SQL query is not working if it’s impossible to see the table and its data somehow (I ran into this problem trying the tool out). For example, someone who works with an SQL database from the command line would use sqlcmd, or through a web server, something like PhpLiteAdmin. Something like that would be very helpful inside the Galaxy interface.

>This is an excellent suggestion, although beyond the scope of our Query Tabular tool which is the focus of this paper. This would most likely require the development of a new Galaxy tool to make this possible, which is something that we will consider pursuing in

future developments.

Competing Interests: No competing interests

Reviewer Report 13 November 2018

https://doi.org/10.5256/f1000research.17980.r39107

Maria A. Doyle
Research Computing Facility, Peter MacCallum Cancer Centre, Melbourne, VIC, Australia

Johnson et al's paper describes Query Tabular, a tool for simplifying text manipulation in the Galaxy platform. The paper describes the tool and shows example use cases for proteogenomics, metaproteomics and metabolomics. The Query Tabular tool can create an SQLite database that can then be queried, saving on multiple text manipulation steps. Example workflows are provided for the use cases.

I really like that the tool enables a user to create a database easily from text files. The instructions in the Supplementary Material for running the example workflows were easy to follow and all the workflows ran without issue. The tool also installed easily from the toolshed and ran without error using a example provided in the tool help section.

Minor suggestions for future revisions:

- Part of the text is a bit confusing, Query Tabular is referred to as a single tool in most of the text, but then it is also described as "the Query Tabular tools" in the Implementation section.
- The user needs to know some SQL to use the tool which could be highlighted more. It could perhaps be noted that this tool might provide a way to introduce/teach SQL to Galaxy users, as it enables them to create a database easily without the need to install anything, so they could just focus on learning SQL queries, like the examples provided by Data Carpentry: https://datacarpentry.org/sql-ecology-lesson/
- The SQL query is shown for workflow 3 (Figure 4) but not for workflows 1 and 2. While the queries are available within the example workflows, it could be helpful to show them all in the text, especially for users not familiar with SQL.
- In the workflows described, it's not always obvious what some of the steps are for (e.g. the Compute step in Figure 2, why there are two Filter sequences by length steps in Figure 3.) Perhaps the generically-named steps in the workflow figure could be changed to have more descriptive names, similar to what some of the steps already have (e.g. "Deduplicate peptides") And/or perhaps a table could be provided describing what the individual steps

are for.
- For the workflows, while the inputs and outputs are available in the example workflows provided, it could be helpful to provide screenshots in the text showing what the input and outputs for the use cases look like.
- The metabolomics workflow utilizes a tool called VKMZ that's described as a tool under development. Would be good to note if the workflow is very specific to this tool or also relevant for other metabolomics analyses.
- Are there known limits on how big tables can be e.g. are tables with tens/ hundreds of thousands/ millions of rows possible.
- The workflows are available in the authors' Galaxy but it could be noted that the tool is available in other public Galaxies (e.g usegalaxy.eu) for people running analyses there. And perhaps the example workflows could be added to the workflows tested there: https://github.com/usegalaxy-eu/workflow-testing
- Would be good to provide a link to the training material that shows further examples and details on how Query Tabular can be used e.g. https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/metaproteomics/tutorial.html

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 19 Dec 2018

Tim Griffin, University of Minnesota, Minneapolis, USA

We thank the reviewer for the comments. Below in bold text we provide responses to these comments and revisions we have made in the updated version.

Johnson et al's paper describes Query Tabular, a tool for simplifying text manipulation in the Galaxy platform. The paper describes the tool and shows example use cases for proteogenomics, metaproteomics and metabolomics. The Query Tabular tool can create an SQLite database that can then be queried, saving on multiple text manipulation steps. Example workflows are provided for the use cases.

I really like that the tool enables a user to create a database easily from text files. The instructions in the Supplementary Material for running the example workflows were easy to follow and all the workflows ran without issue. The tool also installed easily from the toolshed and ran without error using a example provided in the tool help section.

>We thank the reviewer for the kind comments and positive impressions of the Query Tabular tool.

Minor suggestions for future revisions:
--Part of the text is a bit confusing, Query Tabular is referred to as a single tool in most of the text, but then it is also described as "the Query Tabular tools" in the Implementation section.

>We have clarified this in the Methods section under "Implementation". We now clarify that Query Tabular is a single tool, but contains three different modules, in the form of python scripts, which carry out different functions. We describe these three functions, and also clarify that Query Tabular is the main Galaxy tool that provides all of this functionality and is the focus of the description manuscript.

-- The user needs to know some SQL to use the tool which could be highlighted more. It could perhaps be noted that this tool might provide a way to introduce/teach SQL to Galaxy users, as it enables them to create a database easily without the need to install anything, so they could just focus on learning SQL queries, like the examples provided by Data Carpentry: https://datacarpentry.org/sql-ecology-lesson/

>We appreciate the suggestion to include this link to help in learning SQL queries, which we have now included in the text in the Conclusions section. We have also clarified in the text that Query Tabular does require some SQL knowledge, and its use is targeted towards more advanced Galaxy users with SQL knowledge. For those readers without SQL knowledge, this link will provide a useful resource for training.

-- The SQL query is shown for workflow 3 (Figure 4) but not for workflows 1 and 2. While the queries are available within the example workflows, it could be helpful to show them all in the text, especially for users not familiar with SQL.

>We have now expanded Figures 2 and 3 and show the queries utilized in these workflows within the inset boxes. The figure legends have been updated to reflect these additions to the figures.

-- In the workflows described, it's not always obvious what some of the steps are for (e.g. the Compute step in Figure 2, why there are two Filter sequences by length steps in Figure 3.)

Perhaps the generically-named steps in the workflow figure could be changed to have more descriptive names, similar to what some of the steps already have (e.g. “Deduplicate peptides”) And/or perhaps a table could be provided describing what the individual steps are for.

>We have clarified in the text description of the figures that Figures 2 and 3 are meant to show the steps involved in these workflows with or without using Query Tabular, offering a visual depiction of how Query Tabular simplifies these complex workflows. Given this purpose to the figure, we decided not to go into detail on each specific step shown in the workflows.

-- For the workflows, while the inputs and outputs are available in the example workflows provided, it could be helpful to provide screenshots in the text showing what the input and outputs for the use cases look like.

>We have revised Figures 2 and 3 and now show small snippets of the tabular input and output data formats for these workflows.

-- The metabolomics workflow utilizes a tool called VKMZ that’s described as a tool under development. Would be good to note if the workflow is very specific to this tool or also relevant for other metabolomics analyses.

>The workflow shown is specific to data manipulations necessary for the VKMZ tool. Query Tabular however is generally useful for any other data manipulations that may be required for a metabolomics workflow. We have added a statement about the general applicability of Query Tabular in the Conclusions section.

-- Are there known limits on how big tables can be e.g. are tables with tens/ hundreds of thousands/ millions of rows possible.

>We have yet to encounter limits in terms of table size – we have used on data with millions of rows successfully. We would note that it is important to create indices on tables when dealing with a large number of rows or columns. We state this in the Operation section of the methods.

-- The workflows are available in the authors’ Galaxy but it could be noted that the tool is available in other public Galaxies (e.g usegalaxy.eu) for people running analyses there. And perhaps the example workflows could be added to the workflows tested there: https://github.com/usegalaxy-eu/workflow-testing

>We have added the workflows and input data for each to a Github repository, where these can now be accessed and downloaded (https://github.com/galaxyproteomics/query_tabular_supplementary_material). We also mention in the Software Availability section that the Query Tabular tool is available in the Tool Shed, and can be used on local instances or instances such as usegalaxy.eu.
We are also in process of adding the three demonstration workflows to the Github site established for testing workflows (https://github.com/usegalaxy-eu/workflow-testing).

These are listed under names "F1000_Metaproteomics_QueryTabular", "F1000_Proteogenomics_QueryTabular", etc.

-- Would be good to provide a link to the training material that shows further examples and details on how Query Tabular can be used e.g. https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/metaproteomics/tutorial.html

>In the future, we will pursue adding Query Tabular to the Galaxy Training Network in the future, under the category of Text Manipulation. We now do point out in the text that the Galaxy Training Network material does contain a metaproteomics tutorial that utilizes Query Tabular, and provides a detailed description of using this tool within this workflow. This provides readers another example of using Query Tabular on complex data, in addition to our Use Cases described in this manuscript.

Competing Interests: No competing interests

Reviewer Report 05 November 2018

https://doi.org/10.5256/f1000research.17980.r39106

?

Daniel Blankenberg
Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic, Cleveland, OH, USA

Summary:

The authors describe a set of Galaxy tools collectively referred to as "Query Tabular" (but composed of 3 individual tools "Query Tabular", "SQLite to Tabular", and "Filter Tabular"). This tool allows user-defined database operations to be performed on tabular files within Galaxy through the use of generated sqlite intermediate files. Tabular files are very common outputs of bioinformatics tools, and a significant number of Galaxy tools exist for manipulating these types of files. By enabling the the use of SQL statements to transform tabular files, a great deal of effort that currently requires several tools from the 'standard' Galaxy text manipulation toolbox can be performed in far fewer steps.

General comments:

Query Tabular does enable powerful manipulations to be performed, and it can simplify a workflow which may otherwise have many simple text manipulation tools connected together to achieve a similar result. A significant caveat is that the most powerful functions require the user to have working knowledge of SQL ('simple' things like filtering do not). For pre-canned workflows, this is not a problem, but for a typical 'bench scientist' attempting to use Query Tabular this may

prove to be a formidable barrier to usage when developing their own analysis pipelines. This isn't necessarily a problem with the tool, just a fact of the intended tool design, but it does place it into more of the power-user category. This does enable someone with SQL knowledge to easily do a bunch of neat things inside of Galaxy, and it might be beneficial to include a link to a resource with general help on writing SQL and perhaps provide an additional resource that provides examples of some typical operations relevant to common Galaxy tools.

Essential changes:

1. Figure 1: Have "Table Options" section expanded to show that table name is being set and that header line is being used for column names. It would also be helpful to provide a small snippet (~5 lines or so) of the input tabular file that is selected in an additional panel.

2. Provide direct downloads for each example input file and exported workflow (perhaps at Zenodo or a Github repository, etc). Currently a reader needs to visit and register an account at two separate Galaxy servers to gain access to these examples.

Minor suggestions:

1. It might be useful to provide a link to, or list by name, the specific Galaxy Training material tutorial that currently makes use of Query Tabular (https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/metaproteomics/tutorial.html) when mentioning the online training available.

2. More examples of 'real-world' usage could be helpful, especially to users that are less experienced with SQL. Perhaps as a linked external Github page, or similar.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of

expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 19 Dec 2018

Tim Griffin, University of Minnesota, Minneapolis, USA

We thank the reviewer for the comments. Below in bold text we provide responses to these comments and revisions we have made in the updated version.

General comments:
Query Tabular does enable powerful manipulations to be performed, and it can simplify a workflow which may otherwise have many simple text manipulation tools connected together to achieve a similar result. A significant caveat is that the most powerful functions require the user to have working knowledge of SQL ('simple' things like filtering do not). For pre-canned workflows, this is not a problem, but *for a typical 'bench scientist' attempting to use Query Tabular this may prove to be a formidable barrier to usage when developing their own analysis pipelines.* This isn't necessarily a problem with the tool, just a fact of the intended tool design, but it does place it into more of the power-user category. This does enable someone with SQL knowledge to easily do a bunch of neat things inside of Galaxy, and it might be beneficial to include a link to a resource with general help on writing SQL and perhaps provide an additional resource that provides examples of some typical operations relevant to common Galaxy tools.

>The suggestion about clarifying the target user audience for this tool is well-taken. It is true that Query Tabular does require working knowledge of SQL, and as such higher level users and developers of Galaxy benefit most directly from this tool. We have acknowledged this in the Conclusions section, and also we point readers to training material for those unfamiliar with SQL.

Essential changes:
1) Figure 1: Have "Table Options" section expanded to show that table name is being set and that header line is being used for column names. It would also be helpful to provide a small snippet (~5 lines or so) of the input tabular file that is selected in an additional panel.

>We have expanded Figure 1 to show the "collapsed" view of the Query Tabular tool, as well as expanded views of the Table options and Filtering menus which open when selected in the tool. We have also provided a view of snippets of the tabular data that comprise input and output for the use case workflows shown in Figure 2 (proteogenomics) and Figure 3 (metaproteomics). The figure legends have also been updated reflecting these changes.

2) Provide direct downloads for each example input file and exported workflow (perhaps at Zenodo or a Github repository, etc). Currently a reader needs to visit and register an account at two separate Galaxy servers to gain access to these examples.

>We have deposited workflow files (.ga format) for the 3 use cases and also the example input data for these into an accessible Github repository at:

https://github.com/galaxyproteomics/query_tabular_supplementary_material
We describe access to the workflow files and data in this Github repository in the Software Availability section. This repository also contains a README file describing the use case workflows and input data.

Minor suggestions:
1. It might be useful to provide a link to, or list by name, the specific Galaxy Training material tutorial that currently makes use of Query Tabular (https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/metaproteomics/tutorial.html) when mentioning the online training available.

>We have added a link to a newly created training tutorial on proteogenomics that is part of the GTN (https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/proteogenomics-novel-peptide-analysis/tutorial.html). We have added this link to the Conclusions section and we also mention this link as another example of application of Query Tabular for complex data manipulations in the introductory text of the Use Case section.

2. More examples of 'real-world' usage could be helpful, especially to users that are less experienced with SQL. Perhaps as a linked external Github page, or similar.

>We have emphasized in the text the Galaxy wrapper which provides some simplified explanation of how the Query Tabular tool operates (see second line of text in Use Cases section). For those wanting more exposure to more complex data workflows which mimic those encountered by many researchers, we point them to the three use case workflows which were designed as representative, more complex examples where this tool has value and are now available for direct download and install from a Github repository (https://github.com/galaxyproteomics/query_tabular_supplementary_material). Additionally, at the start of the Use Case section, we now point the readers to the proteogenomics workflow on the Galaxy Training Network which provides another "real-world" application of Query Tabular (https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/proteogenomics-novel-peptide-analysis/tutorial.html). We also point readers to Galaxy Training Network material focused on metaproteomics (https://galaxyproject.github.io/training-material/topics/proteomics/tutorials/metaproteomics/tutorial.html) which also utilizes Query Tabular and provides a detailed explanation of the tool and its use within this complex, real-world workflow.

Competing Interests: No competing interests