OPINION ARTICLE

Readable workflows need simple data [version 2; peer review: 3 approved with reservations, 1 not approved]

Claas-Thido Pfaff¹, Helge Bruehlheide², Sophia Ratcliffe¹, Christian Wirth¹, Karin Nadrowski¹

¹Institute of Special Botany and Functional Biodiversity, University of Leipzig, 04103, Leipzig, Germany
²Institute of Biology, Martin Luther University Halle-Wittenberg, 06108, Halle, Germany

Abstract
Sharing scientific analyses via workflows has the potential to improve the reproducibility of research results as they allow complex tasks to be split into smaller pieces and give a visual access to the flow of data between the components of an analysis. This is particularly useful for trans-disciplinary research fields such as biodiversity and ecosystem functioning (BEF), where complex syntheses integrate data over large temporal, spatial and taxonomic scales. However, depending on the data used and the complexity of the analysis, scientific workflows can grow very complex which makes them hard to understand and reuse. Here we argue that enabling simplicity starting from the beginning of the data life cycle adhering to good practices of data management can significantly reduce the overall complexity of scientific workflows. It can simplify the processes of data inclusion, cleaning, merging and imputation. To illustrate our points we chose a typical analysis in BEF research, the aggregation of carbon pools in a forest ecosystem. We propose indicators to measure the complexity of workflow components including the data sources. We illustrate that the complexity decreases exponentially during the course of the analysis and that simple text-based measures can help to identify bottlenecks in a workflow. Taken together we argue that focusing on the simplification of data sources and workflow components will improve and accelerate data and workflow reuse and improve the reproducibility of data-driven sciences.
Introduction
Interdisciplinary approaches, new tools and technologies as well as the increasing availability of online accessible data have changed the way researchers pose questions and perform analyses. Workflow software like Kepler or Pegasus enable access to various sources of scientific data, allow for a visual documentation of scientific analyses and potentially help in breaking down complex analyses into smaller components. However, with an increase in the complexity of analyses and the datasets involved, the complexity of the workflows may easily grow up to a degree that makes them hard to understand and reuse. This is particularly true for data in ecology, which consist of small and highly heterogeneous data files that don’t result from automated loggers but from scientific experiments, observations, or interviews. Current literature deals with different tools to create and manipulate workflows, the data provenance and how semantics can be integrated into workflows. Here we argue that there is a need for quality measures of workflow components, which include scripts, as well as for the underlying data sources. Failure to reuse workflows and available research data is not only a waste of time, money and effort but also represents a threat to the basic scientific principle of reproducibility. Providing feedback mechanisms on the data and workflow component complexity has great potential to increase the readability and the reuse of workflows and its components.

In the following we 1) introduce our concepts of workflow component complexity and identity as well as data complexity. We then use 2) a workflow from the research domain of biodiversity and ecosystem functioning (BEF) to illustrate these concepts. The analysis combines small and heterogeneous datasets from different working groups in the BEF-China experiment (DFG: FOR 891) to quantify the effect of biodiversity and stand age on carbon stocks in a subtropical forest. In the third and last part of the paper we 3) discuss the opportunities for quantifying the complexity and identity of workflow components and data for developing useful features of data sharing platforms to foster scientific reproducibility. In particular, we are convinced that simplicity and a clear focus of research data and scientific workflows are the key to adequate reuse and finally to the reproducibility of science.

Complexity and identity
Here we are interested in workflows that begin with the cleaning, the aggregation, and the imputation of research data. These first steps can make up to 70% of the whole workflow. As data managers and researchers, we want to improve the readability of such workflows whether they are scripts or graphs. Our concept of complexity thus should capture the effort and time needed to understand and reuse such workflows. Regarding the complexity of source code we found similar initiatives that provide quality measures. The Code Climate service for example provides code complexity feedback to programmers in many different programming languages (https://codeclimate.com/?v=b). Their complexity measures take the number of lines of code as well as the repetition of identical code lines into account.

Quantifying data complexity is not as straightforward as workflow component complexity. Datasets used for synthesis in research collaborations often consist of “dark” data, lacking sufficient metadata for reuse. In our experience as data managers of research collaborations, many datasets contain a complete representation of a certain study and thus allow us to answer more than just one single question. This is due to a “space efficient” use of sheets of papers and excel sheets during the field period of the study. Thus, many data columns are used for different measurements, color is used to code for study sites without explicitly naming them in a separate column, which all together constitutes a bad quality data management. Later in the process of writing up, often each specific analysis makes use of a subset of the original dataset only. Data needs to be transformed, imputed, aggregated, or merged with data from other columns to be used in an analysis. Thus, not only the metadata but also the data columns in datasets differ in their quality and usage in a workflow.

Here we argue that data complexity can be quantified by looking at the workflow components needed to aggregate and focus the data for analysis. One of the paradigms of data-driven science is that the analysis should be accompanied by its data. We argue that at the same time, data should be accompanied by workflows that offer cleaning and a meaningful aggregation of the data. Data complexity could then be measured by the complexity of their workflows.

In Garijo et al., they identify common or recurring tasks in workflows including data and workflow oriented motifs. Identifying these motifs or identities of workflow components may allow for an improved sharing of code and workflow components. There are many initiatives that propagate the sharing and the reuse of small pieces of code, including the GitHub service Gist or the Stack Overflow question answer portals. Workflows and components are shared via online platforms like “MyExperiment” which has to date approximately 7500 members presenting about 2500 workflows.
Providing quantitative complexity measures together with automated tagging may further increase component and data reuse and the identification of common tasks may also support the use of semantic technologies to assist in the workflow creation process.\(^7\)\(^8\)\(^9\).

**Example workflow**

**Biodiversity effects on subtropical carbon stocks**

Our example workflow is part of an ongoing study that measures biodiversity effects on subtropical carbon stocks and flows. It represents a typical synthesis in collaborative research projects as it combines eight datasets collected by seven independent research groups collaborating within the BEF-China research platform (www.bef-china.de, DFG: FOR 891). The data is archived, harmonized, and exchanged using the BEFdata web application\(^10\). The meta-data is exported in EcologicalMarkup-Language (EML) format which is used to import the data into the Kepler Workflow system\(^2\), Figure 1. The data describes carbon pools from soil, litter, woody debris, herb layer plants, trees and shrubs that surpass 3 cm diameter at breast height. The data has been taken in the years 2008 and early 2009 on the observational plots of the research platform. The plots span a gradient from 22 to 116 years of plot age and 15 to 35 tree species\(^14\). Our example workflow cleans, imputes and merges the data and terminates in a linear model relating biomass pools to plot age and plot diversity. It shows that carbon pools increase with stand age, however, in plots with high species richness this increase was less steep (p-values for stand age: 0.0006, species richness: 0.0568 and their interaction: 0.0236).

**Workflow design**

We used the Kepler workflow system (version 2.4) to build our workflow. The components in Kepler fall into two categories: “actors”, which handle all kinds of data related tasks, and “directors”, which direct the execution of components in the workflow. Workflow components can perform anything from data import over the transformation of data or the execution of complex scripts containing statistical procedures\(^2\). The components in Kepler can “talk” to each other via a port system. Output ports of components hand over their data to input ports of another component for further consumption\(^2\).

The Kepler “SDF” director was used for execution as it handles sequential workflows. The data was imported using the “eml2dataset” actor as it can import data along the conventions of EML\(^15\). It reads the information available in the metadata file and uses it to automatically set up output ports reflecting the columns in the data to allow a direct consumption by other components in the workflow. For the data manipulation in Kepler we used the “RExpression” actor. It offers an interface to the R statistics environment\(^16\) and thus...
allows arbitrary complex R scripts to be embedded into the workflow. We aimed for uniform workflow components thus setting a limit of 5 lines of code per workflow component as a rule of thumb.

Quantifying workflow complexity
To quantify the complexity of the components we used the number of code lines (loc), the number of R commands (cc) and R packages used (pc), as well as the number of input and output ports (cp) of the components (Equation 1). We further calculated a relative component complexity as the ratio of absolute complexity to total workflow complexity given by the sum of all component complexities (Equation 2).

\[
ac = pc + loc + cc + cp
\]  
(1)

\[
rac = \frac{ac}{\sum_{i=1}^{n} ac_i}
\]  
(2)

As each component in the workflow starts its operation only if all input port variables have arrived, the longest port connection of a component back to a data source defines its absolute position in the workflow sequence (Figure 2). We could thus explore total workflow complexity, individual component complexities, the number of components, and the number of identical tasks (see below) along the sequence of the workflow. For this we used linear models which have been compared using the Akaike Information Criteria (AIC) to select for the most parsimonious model.

Quantifying component identity
Based on our analysis we defined 12 tasks or identities handled by the components a priori (Table 1). We then used text mining tools to characterize the components in the workflow automatically. For this we used the presence/absence of R commands and libraries as qualitative values, the number of input and output ports, the number of datasets a component is connected with, as well as the count of code lines. This allowed us to match the defined identities with

**Figure 2.** This figure shows assigned component positions using the example of the herb layer dataset of the workflow. The absolute position in a workflow is defined by the distance back to the data source. The numbers on the components display the distance count back to the data source. The assignment of positions starts with 0.

<table>
<thead>
<tr>
<th>Identities</th>
<th>Description</th>
<th>Motif</th>
</tr>
</thead>
<tbody>
<tr>
<td>data source</td>
<td>Access (remote or local data)</td>
<td>Data retrieval</td>
</tr>
<tr>
<td>type transformation</td>
<td>Transform the type of a variable (e.g. to numeric)</td>
<td>Data preparation</td>
</tr>
<tr>
<td>merge data</td>
<td>Match and merge data</td>
<td>Data organization</td>
</tr>
<tr>
<td>data aggregation</td>
<td>Aggregate of data</td>
<td>Data organization</td>
</tr>
<tr>
<td>create new vector</td>
<td>Create a vector filled with new data</td>
<td>Data curat./clean</td>
</tr>
<tr>
<td>data imputation modify a vector</td>
<td>Impute data (e.g linear regressions on data subsets)</td>
<td>Data curat./clean</td>
</tr>
<tr>
<td>modify a vector</td>
<td>Modify a complete vector by a factor or basic arithmetic operation</td>
<td>Data organization</td>
</tr>
<tr>
<td>create new factor</td>
<td>Create a new factor</td>
<td>Data organization</td>
</tr>
<tr>
<td>data extraction</td>
<td>Extract data values (e.g. from comment strings)</td>
<td>Data curat./clean</td>
</tr>
<tr>
<td>sort data</td>
<td>Sort data</td>
<td>Data organization</td>
</tr>
<tr>
<td>data modeling</td>
<td>All kinds of model comparison related operations (ANOVA, AIC)</td>
<td>Data analysis</td>
</tr>
</tbody>
</table>
the gathered characteristics of the components. We used non metric multidimensional scaling (NMDS)\(^4\) to find the two main axes of variation in the multidimensional space defined by the characteristics. We then performed a linear regression to identify which of the characteristics and which of identities could explain the variation of the two NMDS axes. Furthermore we compared the complexity of identities. For this we used a Kruskal-Wallis test and a post-hoc Wilcoxon test, since residuals where not normally distributed (Shapiro-Wilk test).

**Quantifying quality and usage of data**

For our analysis we only used a subset of the data columns available in each data source. Thus we quantified the “data usage” of a data source as the ratio of data columns used for the analysis to the total number of data columns in that data source \(3\). In contrast, the usage of data columns in relation to the whole workflow is quantified by the total number of workflow components processing the data, before and after the critical component in the workflow that signifies where the data preparation of a column ends (Equation 3). Similarly, the quality of a data column in relation to the workflow is quantified by the number of workflow components that deal with the data preparation (Equation 4). Thus, the higher the quality value is, the lower the column’s quality in terms of effort needed to prepare it for the analysis. Using these metrics we can compare datasets based on the usage and quality of their data. We did this using Kruskal-Wallis and post-hoc Wilcoxon tests.

\[
usage = \sum_{i=1}^{n} con_{i} + \sum_{j=1}^{n} inf_{j}
\]

(3)

\[
quality = \sum_{i=1}^{n} prep_{i} + 1
\]

(4)

**The beginning matters - results from our workflow meta analysis**

The workflow analysing carbon pools on a gradient of biodiversity consisted of 71 components in 16 workflow positions consuming the data of 8 datasets (Table 3). The data in the workflow was manipulated via 234 lines of R code. The number of code lines per component ranged between 1 (e.g component plot_2_numeric) and 23 (component impute_missing_tree_heights) with an overall mean of 3.3 (± 3.98 SD). See Figure 2 and Figure 3 for a graphical representation of the workflow.

Although we aimed to keep the components streamlined and simple, the absolute and relative component complexity varied markedly. The absolute complexity ranged between 4 and 41 with an overall mean of 9.25 (± 6.77 SD), (Summary: Min. 4.0, 1st Qu. 4.0, Median 8.0, Mean 9.2, 3rd Qu. 12.0, Max. 41.0). Relative component complexity ranged between 0.69% (e.g component calculate_carbon_mass_from_biomass) and 7.03% (e.g component add_missing_height_broken_trees) with an overall mean of 1.59 (± 1.16 SD), (Summary: Min. 0.69, 1st Qu. 0.69, Median 1.37, Mean 1.59, 3rd Qu. 2.05, Max. 7.03).

Total workflow complexity decreased exponentially from the beginning to the end (Figure 4). The exponential decrease means that the decrease in complexity is steeper in the beginning of the workflow than at the end of the workflow, showing that complexity at the end of the workflow did not differ as much as at the beginning of the workflow. From the three models relating the sum of relative component complexities to workflow position, the one including position as logarithm (AIC = 64.69) was preferred over the one including a linear and a quadratic term for position (AIC = 65.67, delta AIC = 0.71), and the one including position as linear term only (AIC = 72.26, delta AIC = 7.3).

At the same time, relative complexity increased in the course of the analysis (Figure 5), since our model with an intercept and a linear term for the workflow position (AIC = 325.85) was preferred. However, we will argue later, that this increase was mainly due to a group of workflow components of extreme simplicity at the very beginning of the workflow visible in the bottom left of Figure 5. These “data type transformation” components convert textual columns into numeric columns. This was necessary as there was columns with intermixed content of categorical and numerical values. As we will outline later, we took this as an opportunity to program a feature for our BEF-Data data management portal, that splits columns mixed with text and numbers to numeric and categorical columns for the EML output.

We could group workflow components according to their assigned identities using text mining. The non metric multidimensional scaling had a stress value of 0.17 using 2 main axes of variation. Several of the parameters, including specific commands of R code, were correlated to axes scores (Table 2). Our defined identities could be significantly separated in the parameter space (\(r^2\ 0.58, p\)-value 0.001): the first axis spans between the tasks “data aggregation” and

---

*Figure 3. The usage and quality measure on an example dataset.* The components marked with a P represent preparation steps of a variable. Here we see three preparation steps so the quality is 4. The components marked with C and I represent direct consumption and an indirect influence. Those together build the variable usage together with all following influenced components.
**Figure 4.** Relative workflow complexity along workflow positions could be best described by an exponential model including position as logarithm (R-squared=0.9, F-statistic: 29.16 on 3 and 10 DF, p-value: smaller than 0.001 *****). This figure shows the model back transformed to the original workflow positions. The gray shading displays the standard error.

**Table 2.** Characteristics of workflow components used to assess variation or similarity between components by means of non metric multidimensional scaling (NMDS). Characteristics include lines of code, use of packages, as well as specific commands (see text for further detail). Figure 6 plots the first two axes of the NMDS. r², Pr(≥ r) and sig. describe R square, Probability, and significance level of a correlation with the characteristic as dependent and the NMDS scores of both axes (NMDS1, NMDS2) as independent variables. For example, “count of codelines” separates workflows components in the NMDS plot, so that components with more codelines are plotted in the upper left quadrat of the plot in Figure 6

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>NMDS1</th>
<th>NMDS2</th>
<th>r²</th>
<th>Pr(≥ r)</th>
<th>sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>abline</td>
<td>-0.595118</td>
<td>0.803639</td>
<td>0.0879</td>
<td>0.024</td>
<td>*</td>
</tr>
<tr>
<td>as.numeric</td>
<td>0.229379</td>
<td>-0.973337</td>
<td>0.3536</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>attach</td>
<td>-0.485526</td>
<td>0.874222</td>
<td>0.0535</td>
<td>0.113</td>
<td></td>
</tr>
<tr>
<td>data.frame</td>
<td>-0.902072</td>
<td>-0.431586</td>
<td>0.8074</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>ddply</td>
<td>-0.802096</td>
<td>-0.597195</td>
<td>0.3456</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>detach</td>
<td>-0.485526</td>
<td>0.874222</td>
<td>0.0535</td>
<td>0.113</td>
<td></td>
</tr>
<tr>
<td>grep</td>
<td>-0.211367</td>
<td>0.977407</td>
<td>0.2885</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>ifelse</td>
<td>-0.663695</td>
<td>0.748004</td>
<td>0.3222</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>is.na</td>
<td>-0.759668</td>
<td>0.650428</td>
<td>0.2800</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>length</td>
<td>-0.601172</td>
<td>0.799120</td>
<td>0.0526</td>
<td>0.182</td>
<td></td>
</tr>
<tr>
<td>lm</td>
<td>-0.199313</td>
<td>0.979936</td>
<td>0.0578</td>
<td>0.157</td>
<td></td>
</tr>
<tr>
<td>match</td>
<td>0.445922</td>
<td>0.895072</td>
<td>0.0057</td>
<td>0.849</td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>-0.994320</td>
<td>0.106435</td>
<td>0.1346</td>
<td>0.016</td>
<td>*</td>
</tr>
<tr>
<td>none</td>
<td>0.977510</td>
<td>0.210891</td>
<td>0.5095</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>plot</td>
<td>-0.689676</td>
<td>0.724118</td>
<td>0.0478</td>
<td>0.254</td>
<td></td>
</tr>
<tr>
<td>predict</td>
<td>-0.595118</td>
<td>0.803639</td>
<td>0.0879</td>
<td>0.024</td>
<td>*</td>
</tr>
<tr>
<td>sort</td>
<td>-0.417914</td>
<td>-0.908487</td>
<td>0.0071</td>
<td>0.954</td>
<td></td>
</tr>
<tr>
<td>strsplit</td>
<td>0.016877</td>
<td>0.999858</td>
<td>0.0633</td>
<td>0.059</td>
<td></td>
</tr>
<tr>
<td>subset</td>
<td>-0.788938</td>
<td>-0.614472</td>
<td>0.0212</td>
<td>0.821</td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td>-0.658793</td>
<td>-0.752324</td>
<td>0.1278</td>
<td>0.004</td>
<td>**</td>
</tr>
<tr>
<td>summary</td>
<td>0.022488</td>
<td>-0.999747</td>
<td>0.0043</td>
<td>0.969</td>
<td></td>
</tr>
<tr>
<td>unique</td>
<td>-0.485526</td>
<td>0.874222</td>
<td>0.0535</td>
<td>0.113</td>
<td></td>
</tr>
<tr>
<td>unlist</td>
<td>0.016877</td>
<td>0.999858</td>
<td>0.0633</td>
<td>0.059</td>
<td></td>
</tr>
<tr>
<td>vector</td>
<td>-0.269756</td>
<td>0.962929</td>
<td>0.4583</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>which</td>
<td>-0.421341</td>
<td>0.906902</td>
<td>0.4582</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>write.csv</td>
<td>0.022488</td>
<td>-0.999747</td>
<td>0.0043</td>
<td>0.969</td>
<td></td>
</tr>
<tr>
<td>count of R functions</td>
<td>-0.796351</td>
<td>0.604834</td>
<td>0.4893</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>count of codelines</td>
<td>-0.530470</td>
<td>0.847704</td>
<td>0.5392</td>
<td>0.001</td>
<td>***</td>
</tr>
<tr>
<td>domain count</td>
<td>0.920778</td>
<td>-0.390087</td>
<td>0.0142</td>
<td>0.704</td>
<td></td>
</tr>
<tr>
<td>count packages per component</td>
<td>-0.781994</td>
<td>-0.623286</td>
<td>0.4004</td>
<td>0.001</td>
<td>***</td>
</tr>
</tbody>
</table>

**Figure 5.** Relative component complexities along the workflow of the carbon analysis. The points are slightly jittered to handle over plotting. Although workflow component complexity slightly increases towards the end of the workflow, this increase is largely due to the many components at the beginning of the workflow with low complexity (lower left corner). R-squared: 0.09, F-statistic: 6.57 on 1 and 61 DF, p-value: 0.01285.

“modify a vector” while the second one spans between the tasks “data extraction” and “data type transformation” (Figure 6).

Our workflow identities had similar complexity, with only one exception: the task “data type transformation” was less complex...
Table 3. The workflow positions listed along with the unique component tasks they contain and the count of components per position.

<table>
<thead>
<tr>
<th>Position</th>
<th>Tasks</th>
<th>Component count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>data source</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>data type transformation, data extraction, create new vector</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>merge data, data imputation, modify a vector, create new vector, data aggregation, sort data</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>create new factor, merge data, data aggregation</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>merge data, create new vector, data imputation, modify a vector</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>create new vector, merge data, modify a vector</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>merge data, data imputation, modify a vector</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>data imputation, create new vector, data aggregation</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>create new vector</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>merge data</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>modify a vector</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>data aggregation</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>modify a vector</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>create new vector</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>data modeling</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6. Workflow components (points) in reduced component characteristics space (Table 2). We used non metric multidimensional scaling (NMDS, see text for further detail) to reduce the parameter space to two axes. Table 2 lists the regression results of the axes scores on the component characteristics, which are plotted in smaller text here. Table 1 lists the a priori tasks, which are plotted in large labels here. Points are jittered by a factor of 0.2 horizontally and vertically to handle over plotting.
(Kruskal-Wallis chi-squared = 41.97, df = 9, p < 0.001) than the tasks “create new vector”, “data aggregation”, “data imputation” and “merge data” (Figure 7). Again, “data type transformation” was only used at the beginning of the workflow to transform columns mixing numbers and text to numbers.

Data usage in relation to data sources was higher in smaller data sources. “Wide” data sources, those consisting of many columns, contributed less to the analysis than “smaller” data sources with fewer columns. While on average 37.4% of the columns in the data sources were used, a linear regression showed that the number of columns not used increased with the total number of columns available per data source (Figure 8).

At the same time, data usage in relation to the workflow was similar for all data sources. Data column usage within the workflow ranged between a minimum of 1 and a maximum of 16 with an overall mean of 6.38 (± 4.25 SD). Although usage was different between datasets (Kruskal-Wallis, chi-squared = 18.05, df = 7, p-value = 0.012), a post hoc group wise comparison could not identify the differences (Wilcoxon test). The data column quality, the amount of processing steps needed to transform data for the analysis, was also similar for all data sources. It ranged between a minimum of 1 and a maximum of 10 with an overall mean of 3.54 (± 2.44 SD). There were no differences in the data column quality between data sources (Kruskal-Wallis, chi-squared = 10.9, df = 7, p-value = 0.14).

Figure 7. The median, 25% and 75% quantiles of the relative component complexities for the component tasks. Letters refer to: a=create new factor, b=create new vector, c=data aggregation, d=data extraction, e=data imputation, f=data modeling, g=data type transformation, h=merge data, i=modify a vector, j=sort data. The small dots are the relative complexities, the diamonds the means. The whiskers are 25% quantile – 1.5 * IQR and 75% quantile + 1.5 * IQR, big black circles are outliers. Signific.: * = 0.05, ** = 0.001.

Figure 8. The linear regression shows the dependency between the total column count of the datasets and the count of unused columns. The gray shaded area represents the standard error. Linear model with total columns as predictor for unused columns: R-squared: 0.925, F-statistic: 74.02 on 1 and 6 DF, p-value: 0.0001.
Discussion

Our paper aims to raise the attention, that simplifying datasets already goes a far way in reducing workflow complexity. We provide illustrations for the amount of effort needed to merge datasets at the beginning of the workflows and suggest the need of feedback mechanisms for data providers on the complexity of their datasets. We further show that specific workflow tasks can be identified using text mining, which could be used for social sharing mechanisms in workflow or scriptlet generation.

The example workflow that we use to illustrate our points shows that components dealing with the cleaning, the imputation, the aggregation and the merging of data contribute the most to the complexity of the workflow. Similarly, Garijo et al. found that these steps can make up to 70% of a whole scientific workflow. In our example the complexity decreases exponentially along the positions in the workflow and thus with the ongoing preparation of data towards the actual analysis (Figure 4). A simplification of the underlying data could significantly reduce the amount of steps needed to prepare the data for analysis and thus the overall complexity of the workflow. Here we argue that the simplicity of data could be fostered by feedback mechanisms to data providers that inform about the usage and the quality of their datasets. While this mechanism could help to improve data that is already available the information could also be used to develop guidelines for good quality datasets in structure and constitution of columns. Additionally the information could guide the development of data management tools that assist researchers or data curators in the process of creating good quality data. Furthermore the information could be employed in tracking down the ownership of data products which remains as unsolved problem and a major concern in sharing data.

In our workflow, “wide” datasets consisting of many columns contribute less to the analysis than smaller datasets with fewer columns (Figure 8). The more data columns a dataset has, the more difficult it is to understand and to describe. The high number of columns in datasets, however, results from the effort to provide comprehensive information about a study in one single file. This often includes different experimental designs and methodologies. These datasets typically result from copying the field notes that are related to the same research objects, but combine information from different experiments. For example, a field campaign on estimating the amount of woody debris on a study site might count the number and a major concern in sharing data.

We show that we could identify workflow tasks using text mining techniques (Figure 6). The identification of common and recurring tasks in workflows may serve several purposes. First the mechanism could be employed to automatically detect and tag workflow components to help progress towards a semantic enhanced workflow environment. Secondly, it could also identify bottlenecks in a workflow, i.e. components that need improvement in terms of simplification. Furthermore the mechanism could be employed to guide the development of a semantic framework that can assist researchers in the process of creating workflows guiding the exploration of useful and compatible components for a certain analysis. In our case, we could identify a task of very low complexity (data type transformation). It constituted the second axis of our NMDS analysis (Figure 6). This task converts text vectors into numeric vectors. Having text vectors that could actually be interpreted as numbers stems from a “weakness” in the EML standard describing the previously mentioned highly complicated field data as it does not allow a per column definition of categorical values for numeric columns. However, it is very common that scientists comment missing values or numbers below or above a measuring uncertainty threshold using text.

In a larger context we are witnessing a growing loss of data which is mostly related to illegible and highly complicated datasets and due to missing metadata. This is especially true for the small and heterogeneous data constituting the long tail of big data. The concern of losing valuable data has led to the invention of several data management tools like DataUP or BEFdata. They help to annotate data in Excel and use Excel sheets as exchange format with the database. This is an important step since many researchers are not well trained in data management and they mainly use Excel for data management and storage. At the same time portals are emerging that allow publishing datasets where the Ecological Archives http://www.hindawi.com/dpis/ecology/, Figshare (http://figshare.com/) or F1000Research are only a few of the alternatives. Data journals are emerging (e.g. Scientific Data http://www.nature.com/sdata/ and Dataset Papers in Science, http://www.hindawi.com/journals/dpis/) which try to provide an impact for data to give researchers credit for all of their work and not only for publications. Workflows are shared via online platforms like “MyExperiment” which has to date approximately 7500 members presenting about 2500 workflows. However, there is only a simple rating mechanism for the workflows which does not identify the complexity or the quality of a workflow and its components.

We here exemplify how to quantify the complexity as well as the quality and the usage of data in scientific workflows, using simple qualitative and quantitative measures. Our means are not meant to be exhaustive but rather could serve as a starting point for discussion towards the development of more sophisticated complexity feedback mechanisms for data providers and workflows creators. Our example workflow strongly relies on the interface component of Kepler connecting to the R statistical environment for the purpose of data manipulation and analysis. Thus the means we provide to measure complexity and quality are adapted to that specific workflow situation. However, adapting our means to further components that work as interfaces to other programming languages should be straightforward. Further complexity attributes could be the inclusion of the variable types of workflow components or a ratio capturing the enrichment or reduction of data consumed by the component. Providing complexity measures at the level of workflow components might help in adapting workflows towards a better readability and reusability and thus improve their value for
reuse. Additionally it can guide the restructuring and simplification of data for better use in workflows, understandability and reuse.

Summary
Providing feedback to researchers about the complexity of their data may be an important step towards improving the quality of scientific data. We show that simple text-based measures can be helpful in quantifying the complexity of data and the workflow. Offering complexity measures can help to identify complicated components in workflows that need improvement. Measures about the usage of data can also help to better propagate the ownership of derived data products as it allows to track down the contribution of each dataset. Identifying common and recurring tasks in workflows could facilitate building up libraries of standard components and semantic frameworks to guide scientist in the process of the workflow creation.

Data availability
figshare: Data used to quantify the complexity of the workflow on biodiversity-ecosystem functioning, http://dx.doi.org/10.6084/m9.figshare.1008319

Author contributions
C.T.P., K.N., S.R., C.W. and H.B. substantially contributed to the work including the conceptualization, the acquisition and analysis of data as well as the critical revision of the draft towards the final manuscript.

Competing interests
No competing interests were disclosed.

Grant information
The data was collected by 7 independent projects of the biodiversity - ecosystem functioning - China (BEF-China) research group funded by the German Research Foundation (DFG, FOR 891).

Acknowledgements
Thanks to all the data owners from the BEF-China experiment who contributed their data to make this analysis possible. The research data is not all publicly available currently but will be in future. The datasets are linked and the ones publicly available are marked accordingly and can be downloaded using the following links. By dataset these are: Wood density of tree species in the Comparative study plot (CSPs); David Eisenberg, Martin Böhne, Helge Brueelheide. Tree size in the CSPs in 2008 and 2009: Bernhard Schmid, Martin Baruffol. Biomass of herb layer plants in the CSPs, separated into functional groups (public): Alexandra Erfmeier, Sabine Both. Gravimetric Water Content of the Mineral Soil in the CSPs; Stefan Trogisch, Michael Scherer-Lorenzen. Coarse woody debris (CWD): Collection of data on dead wood with special regard to snow break (public): Goddert von Oheimb, Karin Nadrowski, Christian Wirth. CSP information to be shared with all BEF-China scientists: Helge Brueelheide, Karin Nadrowski. CNS and pH analyses of soils depth, increments of 27 Comparative Study Plots: Peter Kühn, Thomas Scholten, Christian Geißler.

References


Open Peer Review

Current Peer Review Status:  ?  ?  ?  ×

Version 2

Reviewer Report 19 February 2015

https://doi.org/10.5256/f1000research.6072.r7388

© 2015 Soergel D. This is an open access peer review report distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

David Soergel
Google Inc., Mountain View, CA, USA

Wrong arguments in support of a valid point

It is appropriate that this is now an opinion piece rather than a research paper. Given that, it is essential to state clearly what the opinion is, and what practical consequences it has. As far as I can tell, the opinion is that real-world data are messy and require a lot of cleaning. That much is obvious and requires no demonstration. But what do the authors actually want people to do about it?

Surely the authors are not suggesting that dataset providers should proactively prune their datasets, for instance by removing columns that they consider less interesting or that are rarely used (perhaps based on some feedback mechanism). I hope it goes without saying that such an opinion would be unscientific and dangerous, and should not be published.

I trust the authors instead mean that dataset providers should carefully distinguish data arising from different experiments and experimental designs, by providing them in separate files with adequate metadata, so that the meaning and provenance of each measurement is clear. That is of course true and should be entirely obvious, but it does merit repeating.

To the extent that scientists do in fact mix up their datasets as the authors describe, that is very bad. But the principal argument against it is not that it requires additional work on the part of downstream data consumers (as represented here by workflow complexity). The real argument is that mixed datasets, insufficient metadata, columns of mixed type, and other forms of poor bookkeeping lead to bad science and wrong results, regardless of the workflow system or analysis methods that are used.

So, while I am very much in favor of exhorting scientists to collect and publish their data in ways that are clean, rigorous, simple, and reusable, I think the arguments presented here miss the point as to why those are important goals.

Workflow metrics
Measuring workflow complexity is an entirely separate issue, but interesting in its own right. Any proposal and validation of workflow complexity metrics should be a separate paper. However the measures proposed here lack justification, and neglect the existing large literature on software metrics.

The authors say that "complexity" should reflect how long it takes a user to understand a component, but then propose an entirely arbitrary measure (Eq. 1) with no reference to this "readability" criterion. Why should a line of code and an R package import count the same?

The authors are motivated by the laudable goal of making workflow components more reusable, and assert that component complexity inhibits reuse. But is complexity really the main barrier, or even any barrier at all, to component reuse? What about basic interoperability (i.e., having input and output ports of matching data type)? What about sufficient metadata and documentation of the components? What about search and discovery of available components? etc. etc. Consider the analogous process of importing some library package in R or any other language. Such libraries are often extremely complex, but are designed and marketed with reuse in mind, and many enjoy widespread adoption.

**Workflow component classification**

Use of the terms "Identities" and "Identical tasks" is extremely misleading. The authors classified the components into very general classes such as "data extraction". This does not make the components identical! Table 1, Column 1 header should read "Classes" (and similarly throughout the text).

Did the authors manually label the 71 components with their respective classes? If so, that is not clear from the text (manual labeling is not what "a priori" means, and that's the only related bit I can find).

The "text mining" methods are poorly described. Do the authors mean the NMDS applied to term presence/absence vectors? In any case, this is orthogonal to the rest of the paper and entirely unnecessary. Perhaps the authors hope that their method is generalizable, so that it can be run on large numbers of workflows; indeed they state "we further show that specific workflow tasks can be identified using text mining." But in fact they don't show that. In order to make such a claim, they would need to provide some validation that the automated classification is meaningful, typically by comparing the class predictions with manually labelled data. Is that what figure 6 is meant to demonstrate? If so, 1) that is unclear from the text and caption, 2) the authors do not anywhere actually predict component classes based on feature vectors, and 3) the classes are not at all separated in the figure, so it seems unlikely that any associated class predictions would be correct.

**Inappropriate and overused statistical computations**

Overall, this paper tries to blind the reader with gratuitous use of statistical methods that are not justified and have no clear purpose. If the authors can clearly explain how the use of each statistical method supports their argument, they should do so. If not, the methods should not appear. This applies at least to every mention of a statistical test (Kruskal-Wallis, Wilcoxon, etc.), every mention of AIC, and to the NMDS analysis.

**Competing Interests:** No competing interests were disclosed.
I have read this submission. I believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Reviewer Report 16 February 2015

https://doi.org/10.5256/f1000research.6072.r7389

© 2015 Hettne K. This is an open access peer review report distributed under the terms of the Creative Commons Attribution Licence, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Kristina Hettne
Department of Human Genetics, Leiden University, Leiden, The Netherlands

The title and the abstract correctly summarize the article. The authors investigate an increasingly important subject, namely the role of well-designed scientific workflows in the reproducibility of science. The first step is indeed to create a workflow since that essentially should enable experimental reproducibility, but if the workflow itself is too complex it seems reasonable that that aim can indeed be missed. The tone in the abstract is right for an opinion piece.

Generally, the introduction is adequate for this type of paper, but would indeed be stronger if more relevant references would be included (I second reviewer Barry Demchak here). For example, I was surprised that a sentence dealing with semantic enrichment of workflows did not mention SADI (Wilkinson, Vandervalk and McCarthy, 2011).

I appreciate the section “Complexity and Identity” where the authors outline what they mean by these terms in the context of scientific workflows. I however agree with reviewer Barry Demchak that it is limited in its current version and would benefit from mentioning other types of complexity. If his section would be followed by a section about the analysis strategy, including a motivation for the statistical analyses performed, the rest of the article would hopefully read more easily.

The result section would benefit from use of subheadings. The point that I believe that the authors are trying to make about how to feedback complexity information to data providers is somewhat lost. They come back to this in the Discussion section, but are not being very specific in which type of information this feedback would contain. Could it simply be a recommendation about the number and type of columns in the dataset or would it also contain the background information leading to this recommendation? Also, it would be interesting to know which text mining tools the authors used to characterize the components in the workflow. Inclusion of this information would increase the reproducibility of their analysis.

Competing Interests: No competing interests were disclosed.

I have read this submission. I 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.

Reviewer Report 03 February 2015

https://doi.org/10.5256/f1000research.6072.r7387
The title is appropriate, and entices a reader interested in creating "high quality" workflows.

The abstract reasonably describes the problem and the general approach, and prepares me to understand evaluation methodologies I'll likely find useful across a large number of scientific projects. Somewhat disorienting, though: the characterization of giving "visual access to the flow of data" conflates a common presentation of workflows (i.e., visually) with the real nature of workflows, which is the thrust of the paper.

The overall topic is of great interest, and has been dealt with (somewhat inconclusively) in the computer science literature exhaustively. Citing some of that work would lend a good foundation for the discussion. The unique value of this paper is the use of metrics and statistical analysis to make points about complexity of computation and data. While I support this, the paper would be much stronger if it could justify and give stronger foundation to the choices and formulation of metrics -- intuitively, to me, they seem to conflate the complexity of a workflow component with the overall complexity of the workflow. (The resolution lies in composition/decomposition, encapsulation, and reusability arguments from computer science or mathematics.) Also, data complexity metrics seem to focus on identifying extraneous data, which is trivially filtered -- attention to other types of complexity (e.g., data that cross-references other data) would be useful. I would also like to know whether there are other dimensions to data complexity.

Given a stronger foundation for metrics, the statistical analysis approach is conceptually valuable. To drive the points, closer attention to the statistics being used would be helpful, but only with a much larger sample space. Additionally, I would like to see a section on the analysis strategy, as the use of some of the statistical techniques (e.g., AIC) seems unintuitive.

Housekeeping: Figures 2 and 3 seem to have extraneous numbers (e.g., "1" or "2") or garbled text (by "4" in Figure 2). The text claims 12 tasks in Table 1, but there are 11 tasks.

As a scientific paper, it could easy and usefully be twice as long if it addressed the points above. As an opinion piece, justification of the metrics and placing them on a sound theoretical foundation would be valuable, and would enable reducing the statistical analysis.

Within the context above, I second all of reviewer Missier's comments, and give appreciation to both the authors and Missier for the progress so far.

The great potential value of this work would be its effective targeting of the biology community, which is often best served by concrete proposals for best practices, accompanied by specific examples.

**Competing Interests:** No competing interests were disclosed.
I have read this submission. I 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.

Paolo Missier
School of Computing Science, Newcastle University, Newcastle upon Tyne, UK

The title and abstract do indeed summarise the purpose and content of the paper adequately.

While the goals of the work are laudable, the framework proposed to go about them is not convincing. I see two main problems, firstly with using a single case study to drive the definition and analysis of data and process complexity, and thus to derive results can hardly have general validity. Secondly, by basing the analysis on some questionable assumptions. In what follows, I try to elaborate on these points.

The paper makes a strong case for simplicity of data and components, however that is based on a single sample. This is hardly justified, and at odds with the wealth of quantitative research methods machinery deployed to analyse workflows and data and derive the metrics proposed in the paper.

It would be good to clarify whether the paper’s focus is on the method -- whereby the case study is just an illustrative example, and without any pretense of drawing general conclusions, or on the actual results, which given the very limited evaluation, are questionable.

Regarding assumptions, it is stated “data complexity could be measured by the complexity of their workflow”. How general is this meant to be? I am not sure a process-independent notion of data complexity is given in the paper, but I believe it should be, to clarify the argument. Here complexity seems to be based on how many different usages (and reuse) the data supports, which is fine perhaps, but only one of many possible criteria.

I am also suspicious of process complexity criteria based on lines of code, especially in workflows that are composed of discrete components, often pre-existing and part of libraries. Kepler is idiosyncratic in this, as it assumes most actors are ad hoc programs. More generally, workflow is about coarse-grained composition (eg of third party services), and local coding decisions matter a lot less than in hand-crafted code.

LOC is a very crude measure of complexity. Just as old, but perhaps more appropriate, is the notion of “function points" whereby you express complexity in terms of functionality realised by a component -- regardless of how much code is required to implement a certain function.
LOC alone is also at odds with the idea that languages like R sit on powerful packages, which make for succinct but expressive code. How do you compare R code that implements a whole algorithm in R with one that simply invokes a lib function to achieve the same result?

One could also argue, reading on pg 7 (col 2), that you may be measuring personal coding style rather than actual process complexity.

Other assumptions along the way seem contrived and overfit the (single) example, for instance "output ports of a data source in the workflow directly relate to data columns in the data set". (pg 5,6) In the same section, questionable conclusions follow from this assumption.

So overall, I think the quantitative methods used in the paper are interesting, but they are applied to a framework where a number of initial assumptions are questionable, and seem to be driven by one single example.

**A few specific comments:**
- pg 3 - Complexity: The point is about programs with control structures, but scientific workflows traditionally are dataflows. So does the same notion of complexity apply here?
- pg 4: I feel there is probably too much detail on the science and its results here, which is not the focus of the paper and can be distracting (and uninteresting unless you know the specific science).
- pg 5 col 2: Need to explain AIC.
- pg 7: I found table 2 interesting and generally useful. In contrast, Table 3 is a bit of a mystery to me.

**Competing Interests:** No competing interests were disclosed.

**I have read this submission. I 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 03 Nov 2014

**Claas-Thido Pfaff, University of Leipzig, Germany**

Dear Paolo Missier,

First of all thank you for your valuable input which gave us the opportunity to sharpen the focus of our paper. Your main argument was that we cannot prove our points because we are using a single case study. At the same time you said we should clarify whether our focus is on the results of the analysis - based on only one use case - or the metrics derived for illustrating complexity. However, our main focus is neither on the specific results of this use case, nor on the metrics. We are writing an opinion paper, and both, the use case and the metrics, are illustrations of our opinion. As you say in your comment, - and we take this as a compliment -, we want to “make a strong case for simplicity of data and workflow components”.

Although it is not our intention to use the case study as proof, our paper is accompanied by many statistical analyses and plots. This may fool the reader in believing that we want to present a research article. However, we think that our plots are very useful for other data managers and
scientists in illustrating why it is worthwhile to invest energy into simplifying datasets. This is especially the case for files from the long tail of big data, which are handcrafted, and relatively small data sets resulting from fieldwork and not from automated sensors. To be able to illustrate the problem of merging these files - which is our day to day work as hybrids of data managers and researchers - we chose this case study, as it is representative for our work and the work of our fellow data managers we spoke to. We also think that it is highly useful to illustrate our difficulties in data re-use.

Reworking our text in response to your questions, we scaled down the method descriptions and put a stronger focus on the opinion parts of the paper. We completely reworked the text in many passages and provide here some examples:

For example, in the abstract we changed the sentence (page 1):

“We illustrate our points using a typical analysis in BEF research...”

to:

“To illustrate our points we chose a typical analysis in BEF research...”.

At the beginning of the introduction we sharpened the opinion aspect of the paper, instead of the sentence (page 2):

“However, there is a lack of papers that discuss workflow components within an analysis including data processing.”

we now write :

“Here we argue that there is a need for quality measures of workflow components, which include scripts, as well as for the underlying data sources. Failure to reuse workflows and available research data is not only a waste of time, money and effort but also represents a threat to the basic scientific principle of reproducibility. Providing feedback mechanisms on the data and workflow component complexity has the great potential to increase the readability and the reuse of workflows and its components.”

and other small changes like:

before:

“We thus suggest that focusing on simplifying ...”

after:

“We argue that focusing on simplifying...”

We further added a new paragraph to the discussion on our methods. We explain, that we want to illustrate the possibility to use simple text mining techniques in providing immediate feedback to data providers or workflow creators. We also add additional avenues that could be taken to quantify complexity of further workflow components or scriptlets (last paragraph discussion):
“We here exemplify how to quantify the complexity as well as the quality and the usage of data in scientific workflows, using simple qualitative and quantitative measures. Our means are not meant to be exhaustive but rather could serve as a starting point for discussion towards the development of more sophisticated complexity feedback mechanisms for data providers and workflows creators. Our example workflow strongly relies on the interface component of Kepler connecting to the R statistical environment for the purpose of data manipulation and analysis. Thus the means we provide to measure complexity and quality are adapted to that specific workflow situation. However, adapting our means to further components that work as interfaces to other programming languages should be straightforward. Further complexity attributes could be the inclusion of the variable types of workflow components or a ratio capturing the enrichment or reduction of data consumed by the component. Providing complexity measures at the level of workflow components might help in adapting workflows towards a better readability and reusability and thus improve their value for reuse. Additionally it can guide the restructuring and simplification of data for a better use in workflows, a better understandability and reuse.”

We further agree, that we have provided too little explanation of what we mean by “complexity”. We thus added a paragraph in the introduction, section complexity and identity, to define the aspect of complexity we are concerned about (section complexity and identity, first paragraph):

“Here we are interested in workflows that begin with the cleaning, the aggregation, and the imputation of research data. These first steps can make up to 70% of the whole workflow. As data managers and researchers, we want to improve the readability of such workflows whether they are scripts or graphs. Our concept of complexity thus should capture the effort and time needed to understand and reuse such workflows. Regarding the complexity of source code we found similar incentives that provide quality measures. The Code Climate service for example provides code complexity feedback to programmers in many different programming languages (https://codeclimate.com/?v=b). Their complexity measures take the number of lines of code as well as the repetition of identical code lines into account.”

Our operationalisation of data complexity is based on this approach to workflow complexity. We explain in the same section (paragraph 2 - 3):

“Quantifying data complexity is not as straight forward as workflow component complexity. Datasets used for synthesis in research collaborations often consist of “dark” data, lacking sufficient meta- data for reuse….

… Here we argue that data complexity can be quantified by looking at the workflow components needed to aggregate and focus the data for analysis. One of the paradigms of data- driven science is that the analysis should be accompanied by it’s data. We argue that at the same time, data should be accompanied by workflows that offer meaningful aggregation of the data. Data complexity could then be measured by the complexity of their workflows.”

More technically speaking, you asked for a process independent complexity measure and criticised the use of line of codes, asking how we deal with hidden complexity when using whole script packages with only one line of code. However, the overwhelming majority of data merging efforts we see in our work as data managers are script based, and are not meant to be reused in the same way as software programs. For this reason, function points do not make sense for them. In addition, we do not only use lines of code in our complexity measure. We also include the
number of packages used, for example.

In our paper, in the part on the Example workflow, section Quantifying workflow complexity, we explain:

To quantify the complexity of the components we used the number of code lines (loc), the number of R commands (cc) and R packages used (pc), as well as the number of input and output ports (cp) of the components (equation 1).

However, we are aware that we only use simple and crude methods to assess complexity. As this is not a research paper, but an illustration for an opinion paper, we do not want to focus on the methods too much. On the other hand, we think that it would be good to develop complexity measures for these type of data merging scripts as well as their components and data sources. For this reason we added a whole new paragraph on our methods to the discussion, as stated above.

We agree that our measure of complexity is within one personal coding style only. This has the disadvantage that there is only one person or coding style, but the advantage that the differences between the complexities of components is not confused by different coding styles. In most cases, data merging efforts will be done by one person only. We do not want to generalise for all researchers as to which commands or packages they choose. But from our experience, our coding example is representative for data merging exercises. Independently from coding style, most effort goes into the first data cleaning and aggregation steps, including the effort to understand the different data sets. Whatever means we find to give a feedback on how much effort is needed to reuse this data, it is worthwhile giving it back to the data providers.

In the following, we answer to specific comments:

Paolo Missier: "Other assumptions along the way seem contrived and overfit the (single) example, for instance "output ports of a data source in the workflow directly relate to data columns in the data set". (pg 5,6) In the same section, questionable conclusions follow from this assumption."

We agree that this formulation is misleading. Since we use the EML actor of Kepler to import data, the “output ports” are always the data columns. We did not want to imply a causality here. Data columns appear as output ports in the Kepler actor, because this is how the EML actor works. We reformulate this sentence accordingly. Indeed, the paragraph works without even using the whole sentence (see page 3, section: quantify quality and usage of data)

Before:

“As explained above, output ports of a data source in the workflow directly relate to data columns in the data set. Thus, the number of available ports of a data source is the “width” of a dataset, or the number of data columns. Thus, the usage of a data column in relation to the data source was calculated as the ratio of ports actually used in the workflow to the ports that were not used. This allowed us to relate the number of unused ports to the number of available ports of a data source."

Now:

“For our analysis we only used a subset of the data columns available in each data source. We therefore quantified the “data usage” of a data source as the ratio of data columns used for the
Paolo Missier: "pg 3 - Complexity: The point is about programs with control structures, but scientific workflows traditionally are dataflows. So does the same notion of complexity apply here?"

No, it doesn’t. We now provide a definition of complexity that clarifies that we are interested in the amount of effort and time that has to be invested in data or workflow reuse (see above).

Paolo Missier: “pg 4: I feel there is probably too much detail on the science and its results here, which is not the focus of the paper and can be distracting (and uninteresting unless you know the specific science).”

We have reordered and shortened the paragraphs on our workflow example. However some of the information is interesting for the general reader and are required for the overall understanding. For example that the data sources come from independent projects and are archived in a common platform as well as some basics on workflows. But we have shortened the information on the scientific analysis to one paragraph.

Page 3, section: biodiversity effects on subtropical carbon stocks:

“Our example workflow is part of an ongoing study that measures biodiversity effects on subtropical carbon stocks and flows. It is typical for synthesis tasks in collaborative research projects in that it combines eight datasets collected by seven independent research groups collaborating within the BEF-China research platform (www.bef-china.de). Data is archived, harmonized, and exchanged using the BEFdata web application (citation!!!). Data is exported in EML format and as such imported into the Kepler Workflow system.

The data describes carbon pools from soil, litter, […] from the years 2008 and 2009 on the observational plots of the BEF-China research platform spanning a gradient from 22 to 116 years of plot age and 15 to 35 tree species. Our example workflow merges the data and terminates in a linear model relating biomass pools to plot age and plot diversity. It shows that carbon pools increase with stand age, however, in plots with high species richness this increase was less steep (p-values)."

Paolo Missier: “pg 5 col 2: Need to explain AIC.”

AIC is explained and cited in the methods part (Akaike Information criterion). page 3, right column, section: quantify component identity:

“For this we used linear models which have been compared using the Akaike Information Criteria (AIC) to select for the most parsimonious model.”

Paolo Missier: ”pg 7: I found table 2 interesting and generally useful. In contrast, Table 3 is a bit of a mystery to me.”

Table 1, 3 and Figure 6 are different perspectives on the same topic. Figure 6 shows the ordination result using multidimensional scaling (NMDS) of workflow component characteristics. The NMDS results in 2 axes that span the highest variation of components in the characteristics space. Thus NMDS1, the first axis, spans the highest variation between components. The component identities
in Table 1 as well as the component characteristics in Table 3 were later compared to the axes scores of the NMDS axes. These are the measures given in Table 2. We have changed the text in the captions to point out the relatedness of the figure and the tables. We additionally included and example in how to interpret measures in Table 2 when comparing them with Figure 6.

Table 1:

“Workflow component identities defined a priori and their relation to the data oriented motifs identified by (10). Figure 6 plots the a priori defined identities of the workflow components to the characteristics we measured from each component a posteriori. Characteristics include lines of codes or specific commands (Table 3).”

Table 2:

“Characteristics of workflow components used to assess variation between components by means of non metric multidimensional scaling (NMDS). Characteristics include lines of code, use of packages, as well as specific commands (see text for further detail). Figure 6 plots the two axes of the NMDS. r2, Pr(>r) and sig. describe R square, Probability, and significance level of a correlation with the characteristic as dependent and the NMDS scores of both axes (NMDS1, NMDS2) as independent variables. For example, “count of code lines” separates workflow components in the NMDS plot, so that components with more code lines are plotted in the upper left quadrant of the plot in Figure 6. Signif. codes…”

Figure 6:

“Workflow components (points) in reduced component characteristics space (Table 3). We used non metric multidimensional scaling (NMDS, see text for further detail) to reduce the parameter space to two axes. Table 3 lists the regression results of the axes scores on the component characteristics, which are plotted in smaller text here. Table 2 lists the a priori tasks, which are plotted in large labels here. Points are jittered by a factor of 0.2 horizontally and vertically to handle over plotting.”

**Competing Interests:*** No competing interests were disclosed.
The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com