Helping Scientists Reconnect their Datasets

Abdussalam Alawini
Portland State University
alawini@cs.pdx.edu

David Maier
Portland State University
maier@cs.pdx.edu

Kristin Tufte
Portland State University
tufte@pdx.edu

Bill Howe
University of Washington
billhowe@cs.washington.edu

ABSTRACT

It seems inevitable that the datasets associated with a research project proliferate over time: collaborators may extend datasets with new measurements and new attributes, new experimental runs result in new files with similar structures, and subsets of data are extracted for independent analysis. As these "residual" datasets begin to accrete over time, scientists can lose track of the derivation history that connects them, complicating data sharing, provenance tracking, and scientific reproducibility. In this paper, focusing on data in spreadsheets, we consider how observable relationships between two datasets can help scientists recall their original derivation connection. For instance, if dataset A is wholly contained in dataset B, B may be a more recent version of A and should be preferred when archiving or publishing.

We articulate a space of relevant relationships, develop a set of algorithms for efficient discovery of these relationships, and organize these algorithms into a new system called ReConnect to assist scientists in relationship discovery. Our evaluation shows that existing approaches that rely on flagging differences between two spreadsheets are impractical for many relationship-discovery tasks, and a user study shows that ReConnect can improve scientists' ability to detect useful relationships and subsequently identify the best dataset for a given task.

Categories and Subject Descriptors

H.4 [**Scientific Databases**]: Scientific Data Management and Analysis

Keywords

Scientific Data Management, Relationship Identification, Spreadsheets

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SSDBM '14 June 30 - July 02 2014, Aalborg, Denmark
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2722-0/14/06Ĕ $15.00.
http://dx.doi.org/10.1145/2618243.2618263.

1. INTRODUCTION

Large amounts of scientific data currently exist as dataset files outside of DBMSs. As scientists perform various activities with these datasets—such as combining data split across several files, filtering or rearranging data to aid their analysis, and sharing datasets with collaborators and receiving versions with modifications—the size and the number of dataset files related to a research project increases. In such cases, we find that scientists lose track of the connections between their files and it becomes increasingly difficult for them to decide on a course of action during activities, such as selecting the version with the most up-to-date or complete data for publishing, archiving, or further analysis. Consequently, scientists might spend more time managing their datasets than analyzing their data [16].

In our experience, much of scientific data takes the form of order-sensitive tables, such as spreadsheets, instrument output, and sensor logs. While our interest in general is understanding the connections between scientific datasets of any format, currently we focus, for tool development and testing, on such ordered-tabular datasets and more specifically on spreadsheets. Thus, in this paper, we use *dataset* and *spreadsheet* interchangeably.

Scientists' activities on such ordered tables can result in various relationships between the tables. Identifying these relationships can suggest the original activities on and connections between datasets. We define a *relationship* between two datasets as a property that depends solely on the current data of these datasets; the *connection* is the derivation history between them, which can depend on past activities. For example, when a scientist adds rows to spreadsheet A and saves it as B, knowing that the relationship between these spreadsheets is *prefix* can help the scientist later determine that the connection between them is that B was a row-extension of A rather than that A was a selection from B. Thus, the scientist may select spreadsheet B for analysis as it is more complete. However, identifying relationships, as our user-evaluation reveals, is a tedious and error-prone process when attempted manually. The goal of this work is to automate the discovery of relationships between datasets to help scientists rediscover the original connections, which will enable them to make informed decisions on how to work with their data.

We developed a methodology for identifying relationships between pairs of spreadsheets by first extracting tabular datasets from them. Then the relationship-identification methodology extracts column features to suggest column correspondence and also to suggest potential relationships

between tabular datasets. Our methodology establishes correspondences between sets of columns in order to generate SQL queries to examine column data for suggested relationships. To validate our methodology, we have implemented and evaluated ReConnect, an interactive tool for identifying relationships between datasets in spreadsheets. In this paper, we show that we have initial answers to the following research questions:

- *Is the proliferation of scientific data a significant problem?*
  Our user study results suggest that scientists often struggle to select and decide how to work with their datasets, especially when working with large datasets where they have lost track of the connections.
- *Can we define a set of observable relationships that will help scientists rediscover connections among their datasets?*
  We propose the concept of *relationship identification* (Section 3) as an aid to managing scientific datasets. We also designed a relationship-identification algorithm and implemented, ReConnect, a robust prototype for relationship identification (Section 4).
- *Can existing tools efficiently extract these observable relationships?*
  We evaluated the effectiveness of ReConnect versus four change-inference tools for spreadsheets in identifying relationships that can exist between datasets (Section 5.2).
- *Does an implementation of this approach measurably improve scientists ability to rediscover meaningful relationships?*
  We conducted a user study (Section 5.1) to evaluate the usefulness of ReConnect and the set of relationships it currently identifies. The results show that ReConnect significantly simplified the task of identifying relationships between subjects' spreadsheets.

1.1 Initial Motivation

Services where scientists can upload tabular data to analyze, visualize, and share are becoming more prevalent. Examples include services such as Fusion Tables [11], Dryad [15], ICPSR [1], and SQLShare [14] and emerging services related to consortia such as DataOne [19] and the Research Data Alliance [2]. We observed that an impediment to using such services is the difficulty in determining which dataset to upload: Which dataset is the most complete? Which two versions of datasets go together? Do two datasets overlap? Thus, we began looking for ways to help scientists with this selection task. The task often requires that scientists determine (or recall) the original connections between datasets. Identifying relationships between these datasets could help with this determination.

1.2 Why Spreadsheets?

We chose spreadsheets for our initial focus for the following reasons. First, spreadsheets are widely used for storing tabular datasets across disciplines, whereas other scientific dataset formats, such as Flexible Image Transport System (FITS), Network Common Data Form (netCDF) and Hierarchical Data Format (HDF) are more domain specific.

Second, based on the file-based datasets we examined, we noticed that many datasets can be characterized as ordered tables at the abstract level. The order of rows and columns can contain information about past activities, which we can help determine by identifying order-based relationships, such as *prefix* (see Section 3.3.1). Spreadsheets are one of the formats that capture dataset order. While relational tables can model order, order is not inherent in them and extra work is required to capture order.

Third, to identify relationships, our methodology relies on extracting indicative features from datasets, such as column type and value frequency. We also wanted to explore using additional kinds of information as indicators for relationships, such as formatting and formulas. Spreadsheets are rich in such information. The trade-off here is that other dataset formats have stronger typing of data, which could aid relationship identification. However, ReConnect is equipped with a basic type extractor for spreadsheets, which allows it to utilize type information in identifying relationships between datasets.

2. BACKGROUND

The *relationship-identification* methodology in ReConnect uses two database techniques—data profiling and schema mapping—to help detect relationships between datasets.

2.1 Column Correspondence

ReConnect relies on SQL queries to test relationships on large datasets in an efficient and scalable manner. But before ReConnect can generate SQL queries, it has to establish a column correspondence between the datasets it is analyzing. Our methodology uses schema mapping techniques, namely semi-automated schema correspondence, to produce an initial column correspondence between two tabular datasets, and then relies on the user to correct or confirm the correspondence.

Schema mapping is considered an important integration requirement for applications such as data warehousing, integrating two data sources, and XML message mapping [3, 13]. In our work, we are not using schema mapping to integrate spreadsheets' data, nor to transform data from one spreadsheet to another. Instead, we apply the attribute (column) corresponding part of schema mapping to enable ReConnect to generate SQL queries that examine columns' data to confirm or discard suggested relationships.

2.2 Relationship Identification

In order for ReConnect to quickly identify relationships between datasets without having to extensively analyze their data, it applies data profiling—the process of collecting and analyzing statistical summaries of data to understand its structure and content [17]. After converting spreadsheets into database tables, ReConnect applies data profiling to analyze instances of column values to determine information such as column data type, length, format, value range, discrete values and their frequencies, variance, uniqueness, and occurrence of null values. The relationship-identification methodology uses this profiling information to aid in identifying column correspondences between two tabular datasets and as an indicator or counter-indicator for suggesting relationships between datasets.

3. RELATIONSHIPS IN SPREADSHEETS

To better understand how identifying relationships between spreadsheets can help scientists make informed decisions on how to work with their data, we first present a

Spreadsheet A

Site	Sample	Bottle gp	Depth
NH-10	**1350**	1-3	70.4
NH-10	**1351**	4-6	60.9

Spreadsheet B

Site	Sample #	Bottle gp	Depth
NH-10	**1350**	1-3	70.4
NH-10	**1351**	4-6	60.9
NH-10	**1352**	7-9	28.7
NH-10	**1353**	10-12	3.9

Spreadsheet C

Site	Sample #	temp	salinity
NH-10	**1350**	8.815	32.6
NH-10	**1351**	8.814	32.6
NH-10	**1352**	8.805	32.5
NH-10	**1353**	8.834	32.5

Figure 1: An example of three related spreadsheets: Spreadsheet A is *contained* in spreadsheet B, and spreadsheets B and C are *complementary*. Relationships sensitive to row and column ordering are also important in this domain; see Section 3.3.1"

motivating example (Section 3.1), which we use as a running example throughout the paper. Next, we introduce the theoretical concept behind the *relationship identification* (Section 3.2). Then we informally define some of the relationships that our methodology currently identifies (Section 3.3).

3.1 A Motivating Example

Jennifer, a marine scientist, collaborates with a group of colleagues on assessing the effects of climate change on the Pacific Ocean. She is responsible for managing and analyzing multiple spreadsheets that contain oceanic data, and wants to upload data to an online service to aid in her analysis. However, she often receives multiple versions of a spreadsheet from different collaborators. Consequently, Jennifer has to inspect spreadsheets manually to identify the best ones to upload.

Figure 1 shows three related spreadsheets. Spreadsheets A and B contain water sample data including site ID, sample number, sample source, and the depth at which the sample was collected. Spreadsheet B was created by extending spreadsheet A with two additional rows (rows 4 and 5). The relationship between A and B is *containment* because spreadsheet A's data is contained in spreadsheet B. Spreadsheet C contains the temperature and salinity readings of the same water samples found in spreadsheet B. Spreadsheet C agrees with B on the *Site* and *Sample #* columns, but the rest of the columns are not related by a column correspondence (*Bottle gp* and *Depth* of B do not match *temp* and *salinity* of C.) The relationship between B and C is *complement*, since the unmatched columns provide complementary information about the water samples in B and C.

Jennifer needs a tool that aids her in determining that spreadsheet B is a more complete version of spreadsheet A, and consequently B is the one that she should upload. Additionally, if she is to upload B, she should also upload C as they complement each other. In this example it is easy to eyeball the relationships, but it would be a tedious and error-prone task to determine relationships between spreadsheets with hundreds of rows. Unfortunately, no current tools specifically target this problem.

3.2 Spreadsheets' Activities, Connections and Relationships

Spreadsheets are produced and modified as a result of scientists' *activities*, and may proliferate as a result of some of these activities. For example, users may combine data from multiple spreadsheets, start a new spreadsheet for each day's observations, fill in missing or null values in an existing spreadsheet, or filter or rearrange data to aid their analyses. Understanding that such activities have occurred can help users detect how spreadsheets are *connected* and how to work with their data.

Table 1: An example of the activities that Jennifer's collaborators may have performed on spreadsheets A, B, and C

Activities	Connection	Relationship
Add two water samples (1352 and 1353) rows to spreadsheet A and save it as B.	B is row-extension of A.	A is row-contained in B.
Select rows with a depth greater than 30 meters from spreadsheet B and save it as A.	A is a selected subset of B.	A is row-contained in B.
Store the bottle label and depth of samples 1350 to 1353 in spreadsheet B, and store the water temperature and salinity of the same samples in spreadsheet C.	Unmatched columns provide complementary information about the matching water samples in B and C.	B and C are complementary.
Sort spreadsheet B' on *Sample #* column and save it as B (see Figure 8).	B is a reordering of B'.	B is row-equal to B'.

Scientists' varied activities typically produce various *relationships* between their spreadsheets. Our methodology can detect these relationships by analyzing the data in spreadsheets A and B without necessarily having knowledge of the history of the activities that produced or modified these spreadsheets.

Table 1 shows activities that Jennifer's collaborators may have performed on spreadsheets A, B, and C. Some activities may produce distinct relationships, as in the case of the sort activity, while other activities may produce the same relationship, as in the case of the adding versus selecting rows activities. Our relationship-identification methodology relies on user feedback to remove such ambiguity in identifying connections. The goal of this work is to provide users with contributing evidence so that they can make the

decision as to how their spreadsheets connect to each other.

It of course may be the case that two spreadsheets were not involved in a common activity. Our methodology can also identify the *incompatible* relationship, which indicates the absence of a connection.

Is our approach appropriate for spreadsheets in general?
Our focus is on spreadsheets that contain scientific datasets, but spreadsheets are also used for many other kinds of tabular data: sales reports, budgets, gradesheets, etc. A natural question is whether our methods work for this data. We can certainly detect instances of the current relationships between data in such spreadsheets. But the kinds of relationships that exist and their likelihood might be different than what is common for spreadsheets for scientific datasets. In this work, we focused on the kinds of activities that scientists perform when operating on scientific datasets stored in spreadsheets.

Consider a spreadsheet containing a project budget, it might have individual entries modified many times—a pattern we do not see much with append-mostly scientific datasets. Our tool currently does not detect a relationship that specifically corresponds to this activity, or other activities uncommon with scientific datasets. In order for ReConnect to identify such relationships, we need to build new routines that can handle these kinds of spreadsheets.

3.3 Relationship Definitions

We have developed a set of relationship definitions by studying and analyzing spreadsheets that we have collected from several sources, such as the EUSES spreadsheet corpus [9], and spreadsheets provided by our scientific collaborators. Table 2 presents some of these relationships.

These definitions are based on how the columns of two spreadsheets correspond, and on the data shared between them (See Figure 7). Our methodology classifies column correspondence as *Full Correspondence, Sub-correspondence, T1 (or T2) Corresponds*, and *No Correspondence*. Full correspondence is when each column in the first spreadsheet (T1) corresponds to a column in the second spreadsheet (T2) and vice versa; sub-correspondence is when a subset of the columns of T1 and T2 correspond; (T1)-corresponds when all of T1's columns correspond to a subset of T2's columns; and no correspondence is when there is no column in T1 that corresponds to any column in T2.

We intentionally do not want to express these conditions in relational algebra because of the following reasons: relational algebra does not support order, so we would not be able to detect order-based relationships, such as *prefix* and *subsequence*; it is not a natural language for describing the activities of spreadsheet users. For example, copying and pasting a rectangle out of a spreadsheet generates a *sub-containment* relationship – which is not a selection followed by a projection because based on set semantics projection would remove duplicates.

3.3.1 Ordered Relationships.

The row order between two spreadsheets can provide useful information about how these spreadsheets relate to each other. In Jennifer's spreadsheets, when the *containment* relationship holds between spreadsheet A and B, other relationships based on row order that might also hold are A is a *prefix* of B or A is a *subsequence* of B, as shown in Figure 2.

In the case of a *prefix* relationship, it seems more likely that A was extended to form B, but in the case of *subsequence* relationship, it is more likely that A is a filtered version (a selection) of the rows in spreadsheet B. This example shows how detecting multiple relationships can help users understand the connection between their spreadsheets. Our tool can detect ordered relationships, such as *prefix, suffix, infix, subsequence, reordered columns, reordered rows,* and *reordered columns/rows.*

Table 2: Description of some of the relationships that the relationship-identification methodology identifies.

Relationship	Description
Row-containment	When A fully corresponds to B, and the rows of spreadsheet A are a subset of the rows of spreadsheet B.
Column-containment	When A (A)-Corresponds to B, and the rows of spreadsheet A are a subset of the rows of spreadsheet B.
Containment	When A is both row-contained and column-contained in B.
Sub-containment	When A sub-corresponds to B, and a portion of the rows of spreadsheet A is a subset of the rows and columns of spreadsheet B.
Augmentation (Fill-in)	When A fully corresponds to B, and all rows of spreadsheets A and B match except for particular cells, and these cells are empty or null cells in A but are filled in B.
Complementation	When A sub-corresponds to B, where the rows in the sub-corresponding columns match, and the remaining columns are not related by column correspondence.
Template	When A fully corresponds to B but their data rows are disjoint.
Equal (Duplicate)	Containment in both direction between A and B.
Incompatible	No correspondence between A and B.

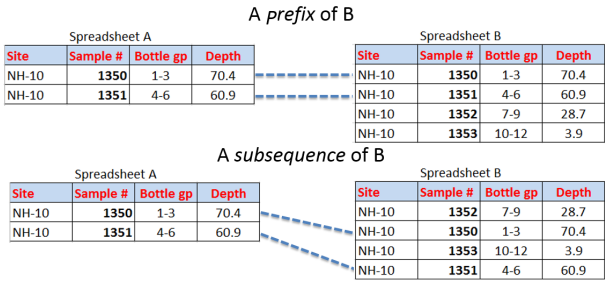


Figure 2: An example of prefix and subsequence relationships

4. AN OVERVIEW OF RECONNECT

As shown in Figure 3, ReConnect's architecture consists of two processes. The first process, *Upload Dataset*, converts a spreadsheet into a database table (Section 4.1). Users improve the accuracy with which tabular datasets are extracted by guiding the conversion process. The second process, *Identify Relationships*, involves two tasks: the first is detecting column correspondence between the two tables (Section 4.2), and the second is analyzing the column correspondence and data profiling statistics to suggest and validate possible relationships between these tables (Section 4.3). Throughout

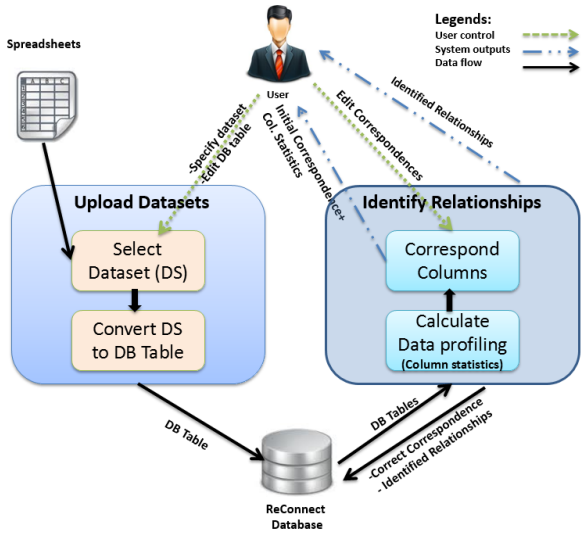
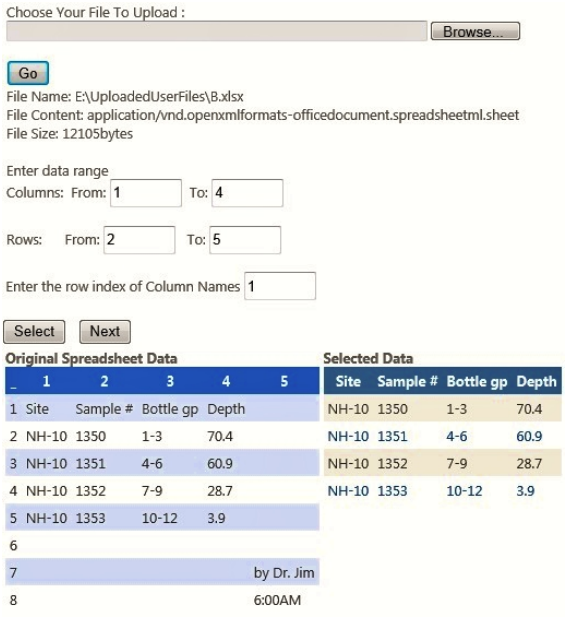


Figure 3: ReConnect's architecture: The process *Upload Dataset* converts a spreadsheet into a database table, and the process *Identify Relationships* detects relationships between pairs of spreadsheets.



_	1	2	3	4	5
1	Site	Sample #	Bottle gp	Depth	
2	NH-10	1350	1-3	70.4	
3	NH-10	1351	4-6	60.9	
4	NH-10	1352	7-9	28.7	
5	NH-10	1353	10-12	3.9	
6					
7					by Dr. Jim
8					6:00AM

Site	Sample #	Bottle gp	Depth
NH-10	1350	1-3	70.4
NH-10	1351	4-6	60.9
NH-10	1352	7-9	28.7
NH-10	1353	10-12	3.9

Figure 4: Table selection for spreadsheet B: Jennifer can specify column and row ranges and the index of the row that contains the column names of her table.

this section, we use the example from Section 3.1 to explain how ReConnect works.

4.1 Converting Spreadsheets to Database Tables

ReConnect converts spreadsheets into database tables in two steps. In the first step, *Select Dataset*, a user selects a tabular dataset within her spreadsheet. (It is also possible for a user to upload several datasets from the same spreadsheet.) As shown in Figure 4, Jennifer selects her table by specifying column (1 to 4) and row (2 to 5) ranges. She then selects the index of the row that contains the column names of her table (row 1 contains B's column names). This step is optional, as some tables may not have headers. When she clicks the *Select* button, ReConnect lets her preview the table.

In the second step, Jennifer can edit database table information such as the table name and column names. She can also specify a primary key for her table. The tool verifies that a selected column satisfies the constraints on primary keys, such as containing neither nulls nor duplicate values. When she is done editing, she can upload her data.

4.2 Column Correspondence

ReConnect must identify correspondences between differently named but conceptually identical columns before it can suggest relationships between two spreadsheets. The column correspondence resulting from this process enables ReConnect to check if two tables are related, and to compute the set of common rows (in the corresponding columns) between two tables. It also enable the tool to generate SQL queries to examine suggested relationships between the tables.

Figure 5 shows the column-corresponding process for A and B spreadsheets from our example in Section 3.1. ReConnect first generates an initial column correspondence by querying the database for column names from table A and B, and then matches these columns by equality. It outputs

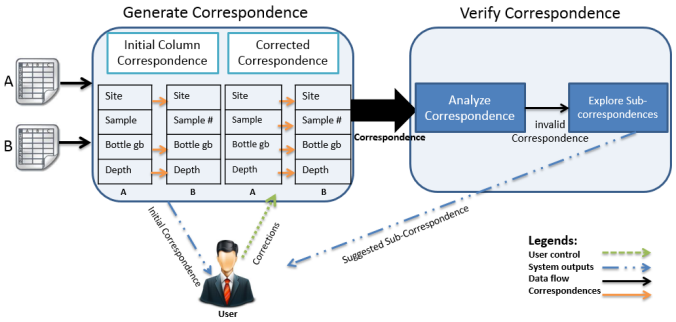


Figure 5: *Column Correspond*, a process for Spreadsheets A and B: ReConnect depends on users' feedback to identify column correspondences accurately.

the correspondences to Jennifer along with column statistics such as the number of null values, the number of unique values, and the maximum and minimum values for each column. These statistics aid her in correcting the proposed correspondences. Next, Jennifer notices that columns *Sample* of A and *Sample #* of B match, so she uses the ReConnect user interface to indicate that these columns are really the same. The example shows user feedback enhancing ReConnect's accuracy in identifying column correspondences.

ReConnect uses Jennifer's column correspondences to compute the rows that are common in A and B: the rows with sample numbers 1350 and 1351 (see Figure 1). Editing column correspondences may result in different sets of common rows, because ReConnect only checks for row matches relative to the corresponding columns. If there were no common rows in these columns, ReConnect would suggest that she explore sub-correspondence, where the tool attempts to match a subset of the columns between A and B looking for a correspondence that produces the largest set of common

rows.

For example, suppose that instead of matching $A.Sample \rightarrow B.Sample$ #, Jennifer mistakenly matched $A.Sample \rightarrow B.Site$, (See Figure 5). If she chose to use the Explore Sub-Correspondence feature, then ReConnect starts by removing the $A.Site \rightarrow B.Site$ correspondence from the column correspondences and computes the set of common rows by matching row values in the remaining corresponding columns: $A.Sample \rightarrow B.Site$, $A.Bottle\ gb \rightarrow B.Bottle\ gb$, and $A.Depth \rightarrow B.Depth$. This sub-correspondence will not produce any common rows. Next, ReConnect removes $A.Sample \rightarrow B.Site$ from Jennifer's column correspondence, and again computes the set of common rows by matching row values in the corresponding columns: $A.Site \rightarrow B.Site$, $A.Bottle\ gb \rightarrow B.Bottle\ gb$, and $A.Depth \rightarrow B.Depth$. This sub-correspondence will produce two common rows: the rows with sample numbers 1350 and 1351. ReConnect continues exploring sub-correspondences until it finds the correspondence that produces the largest number of common rows, which it then suggests to Jennifer.

There are situations where we can have statistically similar columns that in fact do not correspond. For example, assume spreadsheet X has *Initial Temp* and *Final Temp* columns and spreadsheet Y has *Temp1* and *Temp2* columns. Suppose the *Initial Temp* column closely resemble both *Temp1* and *Temp2* (similarly for *Final Temp*). Simple column-to-column comparison does not give much insight on the correct column-correspondence here. However, comparing row values may help detect which of the two correspondences is appropriate, because row values are more distinctive than single column values.

4.3 Relationship Identification

Figure 6 describes the process of identifying relationships, which involves two steps: *Suggest Relationships* and *Validate Relationships*. In the first step, described in Algorithm 1, ReConnect attempts to suggest relationships by analyzing column correspondences and column statistics to check if they are compatible with the conditions for a relationship.

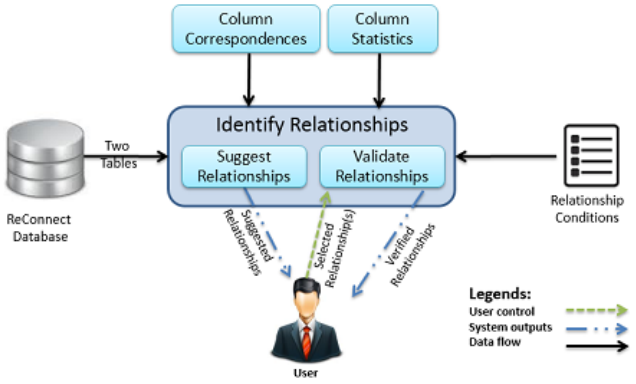


Figure 6: The *Identify Relationships* process: ReConnect analyzes column correspondences, column statistics, and the set of common rows of two tables to check if they match the conditions for a relationship

The result of this step is a "quick and dirty" list of possible relationships—quick because the analysis does not involve the individual data values in the data tables, and dirty because statistics and column correspondences may not be sufficient to validate certain relationships. For instance, if the number of rows of one spreadsheet is greater than the number of rows of another, then it is possible that the relationship between these spreadsheets is *row-containment*. However, for ReConnect to validate such a relationship, it must check whether or not all of the data rows of the first spreadsheet are contained in the second spreadsheet. The purpose of this list of suggestions is to provide users with hints about possible relationships without having to analyze the actual data. It also removes from consideration relationships that cannot possibly hold under the current column correspondence.

Algorithm 1 Suggest relationships algorithm

Input: Column correspondences (*col_corr*); column statistics (*col_stats*) and the set of common rows statistics (*CR_stats*).
Output: List of suggested relationships (*suggested_relts*).

1. *col_corr_class* = classify_correspondence(*Col_Corr*);
   ▷ % classify_correspodence() takes the column correspondences and returns Full Correspondence, T1_Corresponded, Sub-correspondence or No Correspondence%
2. *possible_relts* = get the relationships associated with the *col_corr_class* classification;
3. **for** each relationship in *possible_relts* **do**
4. *relt_conds*= get the relationship conditions;
5. **if** *col_stats* and *CR_stats* are compatible with *relt_conds* **then**
6. *suggested_relts.add*(*relationships*);
7. **end if**
8. **end for**

Figure 7 shows how the *Relationship Identification* methodology classifies relationships based on properties of a column correspondence. Classifying relationships based on column correspondences, such as *Full Correspondence, Sub-correspondence*, and *No Correspondence*, helps ReConnect limit the number of relationships it investigates for a pair of spreadsheets. The figure also shows that relationship conditions are based on row properties such as row count or row-value similarity.

During the second step, *Validate Relationships*, the user can choose the relationship to investigate from the list of suggested relationships. ReConnect validates the selected relationships by generating an SQL query that examines the two tables and then analyzing the results of the query, in addition to column correspondences and statistics. The details of the *Validate Relationships* process are presented in Algorithm 2.

For example, since there is a full correspondence between the columns of Jennifer's spreadsheets A and B, in the *Suggest Relationships* step, ReConnect analyzes column statistics to determine which of the three relationships applies: *Equal, Temple* or *A is row-contained in B*. The tool obtains the row count of A and B ($RC(A) = 2$ and $RC(B) = 4$). Since the number of rows between A and B is different, the *Equal* relationship cannot hold. Because $RC(B) > RC(A)$ ReConnect adds *A is row-contained in B* to the list of suggested relationships. It also adds the *Template* based on the column correspondence.

Suppose that Jennifer selects the *A is row-contained in B*

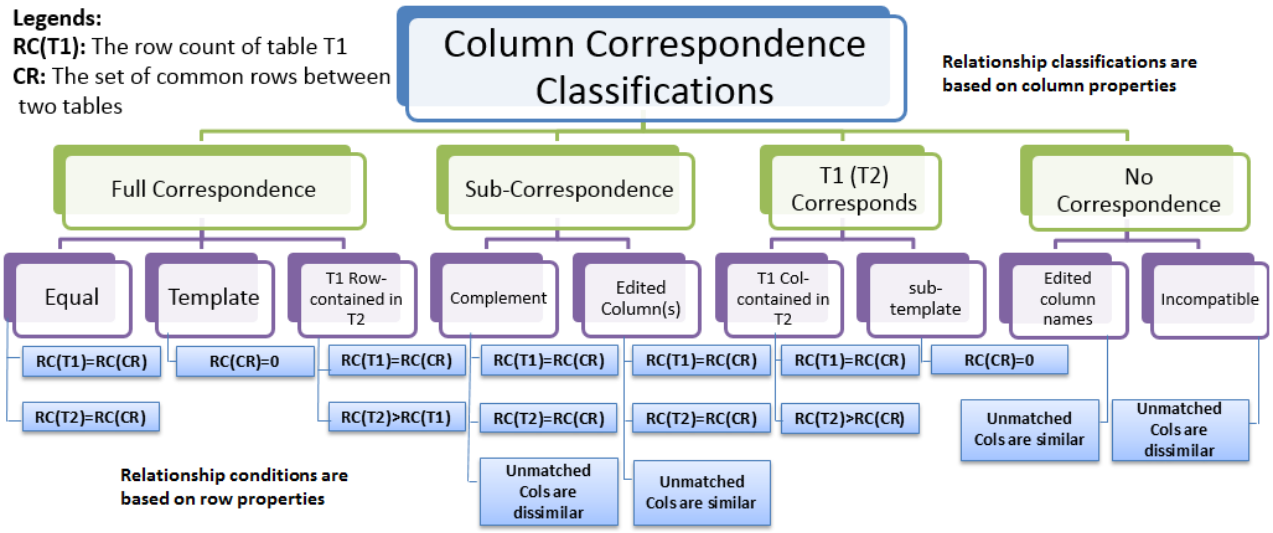


Figure 7: Relationship Identification Overview: Classifying relationships based on column correspondences helps ReConnect limit the relationships to investigate.

Algorithm 2 Validate relationships algorithm

Input: User selected relationship (*suggested_relts*); relationship conditions (*relt_conds*); column correspondence (*col_corr*) and tables A and B.
Output: Confirmation of the relationship.

1. *valid_test*= get the validation tests for the selected relationship *suggested_relts*,
2. **for** for each test in *valid_tests* **do**
3. $test\_qry = generate\_qry(col\_corr, suggested\_relts)$ ▷ % Based on predefined validation test, generate_qry() uses the column correspondence to generate an SQL query that is used to test the tables data for the suggested relationship. %
4. $test\_results = run\_query\_in\_DB(test\_query)$;.
5. $results\_stats = analyze\_results(test\_results)$; ▷ % analyze_results() collects statistics and other information about the results of the test query, which ReConnect uses to validate the suggested relationship. %
6. **end for**
7. **if** all validation tests passed **then**
8. Confirm relationship;
9. **else**
10. Invalidate relationship;
11. **end if**

relationship for validation. The *Validate Relationships* algorithm starts the validation process by retrieving the validation tests associated with the selected relationship. As part of each validation test, ReConnect generates an SQL query that examines certain relationship features. For instance, to count the common rows between A and B, ReConnect generates the following query (which handles repeated rows):

```
SELECT SUM(MIN2(t1.Cnt,t2.Cnt)) AS RC_CR
FROM (SELECT ta.Site, ta.Sample, ta.[Bottle gp],
     ta.Depth, COUNT(*) as Cnt FROM A ta
       GROUP BY ta.Site, ta.Sample, ta.[Bottle gp],
            ta.Depth) t1,
     (SELECT tb.Site, tb.[Sample #], tb.[Bottle gp],
          tb.Depth, COUNT(*) as Cnt FROM B tb
       GROUP BY tb.Site, tb.[Sample #], tb.[Bottle gp],
            tb.Depth) t2
WHERE t1.Site=t2.Site and t1.Sample=t2.[Sample #] and
     t1.[Bottle gp]=t2.[Bottle gp] and t1.Depth=t2.Depth
--MIN2(Arg1,Arg2) returns the minimum value out of the
     two arguments passed to it.
```

The query above joins tables A and B based on the column correspondence, and then counts the set of common rows between the two tables ($RC(CR) = 2$). Since $RC(B) > RC(A)$ and $RC(A) = RC(CR)$, Reconnect reports to Jennifer that the relationship between spreadsheets A and B is A is *row-contained* in B.

5. USER STUDY AND EVALUATION

Our assessment of our work has two parts. First, we conduct a user study to asses ReConnect and the set of relationships it identifies. The goal of this study was to answer the following research questions.

RQ1 Does detecting relationships between spreadsheets help subjects select and manipulate their data?
RQ2 Do the relationships ReConnect detects actually turn up in real-life spreadsheets?
RQ3 When subjects use spreadsheets, do they confront problems that ReConnect seeks to solve?
RQ4 Are there other kinds of relationships we should consider detecting?
RQ5 Do aspects of spreadsheets that ReConnect cannot currently handle interfere with detecting relationships?

Second, we evaluated the effectiveness of ReConnect and four commercial and research change-inference tools for spreadsheets in identifying relationships between spreadsheets.

5.1 User Study[1]

In our study, we asked researchers to attempt to identify relationships between their spreadsheets with and without the help of ReConnect. The purpose of this study was to asses the applicability of the concept of relationship identification, and the usefulness of ReConnect in simplifying the task of detecting relationships between spreadsheets. Our user study consisted of three parts. In the first part, we asked subjects to manually detect relationships between their spreadsheets. In the second part, subjects used ReConnect to do the same task they had performed in part one using the same datasets. In the final part, we conducted interviews with subjects about their experience with the relationship-identification task both with and without the use of ReConnect.

5.1.1 Subjects

To recruit subjects for the study, we sent emails to mailing lists for faculty and graduate students in several science departments in different universities and research labs. We recruited 10 subjects from various science fields such as Biology, Chemistry, Economic, Agriculture, and Computer Science. Macefiled [18] indicates that a size of 10 is effective for early conceptual prototypes. He states that "In the case of studies related to problem discovery in early conceptual prototypes, there are typically factors that drive the optimal group size towards the lower end of this [3-20] range" [18].

Prior to user session, we asked the subjects to send us a pair of spreadsheets they use in their research work. We asked for spreadsheets that the user believed were connected, though we did not require that they be certain of the existence or the exact nature of the connection. Thus, we were able to evaluate ReConnect using real-life research spreadsheets that subjects provided.

5.1.2 Methodology

We conducted the study using a desktop computer with dual wide-screen monitors, which enabled subjects to view their spreadsheets side-by-side to facilitate the manual inspection of relationships. The computer runs Microsoft Excel 2010 on Windows 7 Enterprise Edition. Sessions were conducted on a one-on-one basis, where the first author supervised the session. The session consisted of three parts, which we describe below.

Part One: Detecting of Relationships Manually.

For each subject, we first explained the concept of *relationship identification* using the example spreadsheets shown in Figure 1. We also introduced the set of relationships that ReConnect identifies. Then, each subject was asked to visually inspect the spreadsheets he or she provided looking for relationships that could help him or her understand how the spreadsheets connect to each other. Throughout this part of the study, we took notes on how subjects inspected their spreadsheets, and answered subjects' questions.

Part Two: Detecting Relationships Using ReConnect.

We first demonstrated ReConnect using spreadsheets shown in Figure 1. Next, subjects used ReConnect to convert their spreadsheets into database tables, and then to detect relationships between their spreadsheets. During this session, we provided minimal support for the subjects, and took notes on how they interacted with the tool.

Part Three: Interviews.

In the last part of each session, we interviewed subjects to obtain their feedback about the task of identifying relationships with and without the use of ReConnect. The main goal of this interview was to find out how helpful detecting relationships was in deciding how to work their spreadsheets, and whether there were other relationships of interest.

5.1.3 Results

Even with the aid of wide-screen dual monitors and the vertical and horizontal side-by-side view feature that MS-Excel provides, all subjects found it difficult to visually inspect spreadsheets for relationships. Most difficulties arose from inspecting spreadsheets with a large number of columns and rows or differently ordered rows and columns, and from attempting to detect whether or not two spreadsheets share a subset of their rows or columns (or both). As a result, subjects sometimes reported relationships incorrectly between their spreadsheets. For example, a chemistry researcher thought that she had edited a number of rows in one of her spreadsheets and saved it with a different name. In fact, the rows of the two spreadsheets were identical but appeared in a different order. However, she was able to correctly identify that she had added columns that represented new samples' data.

In a few cases, subjects were able to "guess" such relationships as column-containment and row-containment. However, they were not able to confidently confirm the results of their observations. For instance, while visually inspecting her spreadsheets, a computer scientist remarked, "Based on the number of rows in both spreadsheets, I guess the relationship is row-containment. But to confirm that, I need to write a VBA (Visual Basic Application) script." However, writing VBA scripts might be a difficult task for users with no programming experience. Overall, subjects were unable to confidently identify relationships between their spreadsheets through visual examination.

With ReConnect, subjects quickly and effectively detected useful relationships, which enabled them to recall the activities they performed to transform one version of a spreadsheet into another, and to decide how to further reuse or combine their datasets. For the physics researcher, in addition to confirming the column containment relationship she detected manually in part one, she found that the rows of the spreadsheet with more columns were filtered and reordered to facilitate the analysis of an experiment she was conducting.

Besides detecting relationships, the physics and chemistry researchers found that some of ReConnect's features could be used for data analysis. For instance, after analyzing several correspondences between the columns of her spreadsheets, a chemistry researcher stated "Often, I repeat experiments with minor changes in my experiment configuration, such as increasing the temperature of the sample environment by 10 degrees. Using this tool, I can test several column correspondences to analyze the effect of such

[1]We had IRB approval for our user study

configuration changes on various experimental results."

5.1.4 Discussion

The first two parts of the sessions suggest that ReConnect significantly simplified the task of identifying relationships between spreadsheets, a positive answer to RQ1. Results also suggest that subjects found value in using ReConnect to aid in data-analysis tasks, which is a potential direction for future work. Regarding whether or not the application of relationship identification simplified the task of deciding how to work with data stored in their spreadsheets (RQ2), the results of the interviews appear promising. All subjects agreed that determining relationships removed the burden of comparing and analyzing spreadsheet cells, columns, and rows. Additionally, subjects described several activities they regularly perform on spreadsheets that can produce the types of relationships that ReConnect detects.

Most subjects had positive responses regarding whether, when working with spreadsheets, they face problems that ReConnect can help solve (RQ3). One subject stated that she does not often face such problems because her spreadsheets contain reference information that is rarely updated. However, she stated that she previously confronted similar problems when she worked on a collaborative research project.

Subjects suggested a number of relationships to add to ReConnect's identification capabilities (RQ4). For instance, one subject suggested detecting pairwise column equality within a given percentage range (*near-match* relationship), which could aid her in analyzing her spreadsheets' data. For instance, she might want to detect all the rows whose temperature columns are equal within ± 2%. Regarding whether some aspects of spreadsheets might interfere with detecting relationships (RQ5), only one subject stated that cell formulas could interfere with detecting relationships, as they may present irrelevant information about the original spreadsheets' data.

While the limited number of subjects may not enable us to detect all possible issues with ReConnect, the group size was sufficient for early problem discovery. The importance of this study was that it verified that there are no major issues with the tool, nor with our assumptions about the usefulness of the relationship-identification methodology. It is also worth mentioning that ReConnect response time was not an issue, for the size of spreadsheets our users had.

5.2 Methodological Evaluation

Since there are no commercial or research tools that are aimed at discovering relationships between spreadsheets (or tabular datasets more generally), we looked at other tools that may help scientists with this task. Change-inference tools, which enable users to identify changes between pairs of spreadsheets, much like "diff" utilities on documents, have some capabilities that might help with the task of discovering connections among datasets. The aim of this study was to investigate the effectiveness of our work (ReConnect) compared to change-inference tools in helping users identify relationships between spreadsheets.

5.2.1 Selected tools

We selected one research change-inference tool (SheetDiff [5]) and three commercial tools (DiffEngineX [10], Synkronizer [22], and Excel Inquire [20]) for our experiment. These tools either generate a report of differences and highlight differences between spreadsheets in both spreadsheets, as is the case with Synkronizer, DiffEngineX, and Inquire, or do so only in one spreadsheet, as is the case with SheetDiff.

5.2.2 Methodology

We used two sets of spreadsheets for our investigation. The first set was a collection of related pairs of real-life research spreadsheets that our user study participants had provided. Since the relationships between each of these pairs had been identified and confirmed during the user study, we used this set to evaluate whether or not change-inference tools would help us get to the same results we had obtained in the user study.

The second set contains 10 pairs of spreadsheets that we selected from the EUSES corpus and from other sources. This set was constructed to test combinations of relationships and spreadsheet features that are not covered by the first set. For instance, if we tested the containment relationship between small spreadsheet instances from the first set, then we selected pairs of spreadsheets with large dataset instances that have the containment relationship between them for the second set. We also had modified some of the selected pairs to test for special relationships cases such as *infix*, *prefix/suffix*, *reordered row*, *reordered columns*, and *reordered rows/columns* relationships.

We started by grouping spreadsheets pairs based on the relationship they represent. Next, we used these pairs as the input to each of the selected tools. Each tool was tested with small and large spreadsheets instances. Then, we analyzed the results of each tool to evaluate how readily these results can be used to confirm the existence of a previously identified relationship between the inputted pairs.

5.2.3 Results

The results of our experiment are summarized in Table 3. Regarding the *duplicate* relationship, all tools were able to detect it for both large and small spreadsheet instances. The user can easily identify duplicate pairs when change-inference tools reported no changes. For the *cell reference* relationship, only Inquire has the capability to provide a graphical representation (relationship diagram) of links (formula references) between the current workbook (worksheet or cell) and all other workbooks (worksheets or cells). Such a feature can help users determine relationships between financial spreadsheets, such as budgets and financial statements. However, in the case of scientific spreadsheets, formula references are rarely used.

Regarding the containment relationships, including *row containment*, *column containment*, and *sub-containment*, only ReConnect was able to identify them between both small and large spreadsheet instances. All other tools were able to help in detecting containment relationships only between small instances. For instance, users may identify a *column-containment* relationship by analyzing the additional columns that Synkronizer reported. However, for large spreadsheet instances, change-inference tools generated a large list of changes that was hard to comprehend, and hence complicated the relationship-identification task. For example, DiffEngineX reported 1472 changes between an 80-row spreadsheet and its row-extended and reordered version.

None of the change-inference tools were able to help in detecting complex relationships, such as *reordered rows/-*

Table 3: The methodological evaluation results

Relationship	DiffEngineX	Synkronizer	SheetDiff	Inquire	ReConnect
Duplicate	+	+	+	+	+
Cell/Sheet Reference	−	−	−	+	−
Row Containment	√	√	√	√	+
Column Containment	√	√	√	√	+
Sub-Containment	√	√	√	√	+
Augmentation	√	−	√	√	+
Complementation	−	√	√	√	+
Template	√	−	√	√	+
Infix/Prefix/Suffix	−	√	−	−	+
Subsequence	−	√	−	−	+
Reordered Rows	−	√*	−	−	+
Reordered Columns	−	√*	√*	−	+
Reordered Rows/Cols	−	−	−	−	+
Incompatible	√	√	√	√	+

+ The tool can identify relationships in both small and large instances of spreadsheet pairs.
√ the tool can only identify relationships in small instances of spreadsheet pairs.
− The tool does not help identify relationships in small nor large pairs of spreadsheet pairs.
* Results are presented in terms of missed/added rows/columns. For small spreadsheets instances users may be able to infer that two spreadsheets are actually the same with different row/column order.

columns, where two spreadsheets contain an identical data but have different row and column orders. Spreadsheet B and its reordered version, B', shown in Figure 8, illustrate the issue of order sensitivity. As show in Figure 9, SheetDiff reported that columns *Sample #* and *Bottle gp* each have two unmatched cells (the second and the fourth cell). SheetDiff also reported that column *Depth* of B' has been added to B and that *Depth* of B was deleted without noticing that the deleted and added columns are actually the same with different row order. However, since ReConnect uses schema correspondence, it matched columns *Depth* and *Bottle gp* of spreadsheet B with their counterparts in spreadsheet B' in spite of their order. ReConnect used this correspondence to compute the set of common rows without sensitivity to the rows' order. Because the columns of spreadsheets B and B' are fully corresponded, and the row counts of B and B' are equal to the row count of the set of common rows, ReConnect reported that spreadsheets B and B' are *equal*.

Spreadsheet B

Site	Sample #	Bottle gp	Depth
NH-10	1350	1-3	70.4
NH-10	1351	4-6	60.9
NH-10	1352	7-9	28.7
NH-10	1353	10-12	3.9

Spreadsheet B' (Reorderd version of B)

Site	Sample #	Depth	Bottle gp
NH-10	1350	70.4	1-3
NH-10	1353	3.9	10-12
NH-10	1352	28.7	7-9
NH-10	1351	60.9	4-6

Figure 8: An example of two equal spreadsheets with reordered rows and columns: Change-inference tools report that spreadsheets B and B' are not equal. ReConnect reports that they are *equal* regardless of their row and column order.

Even with small spreadsheet instances, most change-inference tools reports were not useful enough to help in detecting order-sensitive relationships, such as *infix, prefix/suffix, subsequence,* and *reordered columns/rows*. Only Synkronizer and SheetDiff were able to help in detecting some of these relationships in small spreadsheet instances, due to their ability to optimize cell changes into higher level changes (e.g. added row(s), deleted column(s)). However, Synkronizer was able to detect four out of the five order-sensitive relationships, where SheetDiff was able to detect only one. This advantage is because of Synkronizer's ability to identify when rows (or columns) positions have changed between two spreadsheets. However, Synkronizer failed to identify equal spreadsheets when rows and columns were reordered.

	A	B	C	D	E
1	Site	Sample #	Depth	Bottle gp	Depth
2	NH-10	1350	70.4	1-3	70.4
3	NH-10	1351	3.9	4-6	60.9
4	NH-10	1352	28.7	7-9	28.7
5	NH-10	1353	60.9	10-12	3.9

Figure 9: SheetDiff results of comparing spreadsheets B and B' (shown in Figure 8). Yellow is used for changed cells, blue for added columns/rows, and red for deleted columns/rows.

In conclusion, using change-inference tools, the larger a pair of spreadsheet instances was, the longer the list of changes users had to analyze, the more difficult the task of understanding the relationship between their spreadsheets. ReConnect offers a "set at a time" approach to disambiguating a sea of spreadsheets, while the other tools seem to be row-, column- or cell-oriented — which does not scale as spreadsheets become larger. In addition, user-verified schema correspondence allows ReConnect to easily identify order-sensitive relationships. ReConnect is working on a more conceptual level than other change-inference tools, and hence reports changes in a more abstract and compact form.

6. RELATED WORK

6.1 Similar tools

Change-inference tools may help users understand simple relationships between small spreadsheet instances. Using these tools, users may infer relationships by analyzing the change lists they generate. As our evaluation results show (Section 5.2), none of these tools can provide enough help for users in identifying simple relationships in spreadsheets

with hundreds of rows or columns. Furthermore, in the case of two spreadsheets that contain the same data but have different row or column orders, change-inference tools do not detect that they have an identical set of rows.

6.2 Relevant Techniques

The problem of Record Linkage (RL)—identifying records coming from different sources and representing the same real world entity—has received a significant attention from statistic and computer science researchers [8]. We believe that some of the RL techniques may be used to extend our work in two ways. First, field-matching techniques including character-based techniques, such as edit distance, or token-based techniques, such as Q-Grams with tf.idf [12], can be used to improve our column correspondence process by enabling ReConnect to match similar columns with different formats. Second, we may be able to use record-matching techniques, such as Automated Object Matching [23], to realize additional relationships (e.g. a near-match relationship suggested by one of our user study participants). However, we still need to do a higher level analysis of the results of RL techniques in order to detect such relationships. Giving users back a matching list of identical records still puts the burden on them of analyzing the list to understand how the datasets as a whole connect. As we showed in evaluating change-inference tools (Section 5.2), ReConnect is working on an abstract level allowing scientists to understand connections among their datasets without having to analyze individual records.

ReConnect adapts Clio's [13] idea of integrating users' feedback in the schema-mapping process. Clio, a semi-automated tool that maps column names of two database tables, has an interactive user interface that allows users to dynamically provide feedback on proposed schema mappings. As a result, users have full control over column mappings and can map similar schemata differently for various purposes. After computing an initial column correspondence, ReConnect allows users to correct the computed correspondence, which improves the accuracy of the schema correspondence process and allows users to test various correspondences. ReConnect also provides the *explore sub-correspondence* feature, which automatically searches for the correspondence that produces the largest set of common rows between spreadsheet pairs and reports that correspondence to users.

Data profiling is commonly used in data cleaning and data integration [4, 21]. Data-cleaning applications use profiling information to aid in analyzing different aspects of attributes' quality. For example, the max and min could be used to check whether or not the values of a given attribute (data column) are within the expected range. In our work, we use data profiling differently; as an aid to aid in identifying attribute correspondences between two different schemas, which is similar to the way it is used in data integration applications [7].

The Bellman system [6] is a browser for complex databases that provides tools and services to help users di scover the structure of databases. ReConnect collects the same statistics that the Bellman system collects, including the number of rows, the number of distinct values in a column, the number of null values per column, and the ten most common values in a column along with their respective frequencies. However, the Bellman system uses this profiling information to help data analysts understand the structure of a database, whereas ReConnect uses it to help users correspond columns correctly and to guide the process of relationship identification.

7. FUTURE WORK AND FURTHER APPLICATIONS

We have short-term and long-term plans for extending our methodology. In the short term, because scientists often use spreadsheets' metadata, such as cell formulas, formats, and comments, we are working on incorporating this information in the process of relationship identification. For instance, ReConnect might check if an added column contains derived data by checking if the values of that column are computed from formulas. Such an enhancement can also address the issue of formulas' interfere with detecting relationships, which one subject was concerned about in the experiment.

During the second part of our user study, several scientists suggested that ReConnect could combine tabular datasets or remove irrelevant datasets based on the suggested relationships. Thus, we plan to extend our methodology so that it suggests and follows a course of action based on the relationships that it identifies. For instance, the relationship-identification methodology might be able to join complementary information in a table, create data views that reduce data duplication and complexity.

Because scientists often need to work with a large collection of related spreadsheets, we have longer-term plans to explore several directions that involve working with suites of spreadsheets.

The first direction might be multi-spreadsheet connection identification. A connection is identified based on a certain combination of pairwise relationships. A good example would be the concatenation connection. If spreadsheet Y was formed as the concatenation of spreadsheets $\{X_1, ..., X_n\}$, or a subset of them, then the following relationships exists:

a. Some X_i is a *prefix* of Y, where $0 < i \leq n$,
b. some X_j is a *suffix* of Y, where $0 < j \leq n$,
c. zero or more X_k are *infixes* of Y, where $1 < k < n$, and
d. all X_l are pairwise disjoint, where $1 \leq l < m$ (where m is the total number of spreadsheets involved in the concatenation).

The second direction is helping users choose which pairs of spreadsheets to compare from a large collection of spreadsheets. For that, we can utilize the column statistics that ReConnect collects to provide guidance on how similar a pair of spreadsheets are.

A third possible direction is clustering spreadsheets based on similarity measures for fast testing for potential relationships on large spreadsheet repositories, where performing pairwise comparisons for all spreadsheets is not feasible.

We also plan to explore the applicability of the relationship-identification methodology to tabular datasets originating in other settings, such as Relational Database Management Systems (RDBMS). A straightforward way for our tool to detect the same relationships in ordered-table datasets that originate in other formats is to convert these formats into spreadsheet format. However, the relationships that may exist between datasets in different formats could be different in kind and likelihood based on the tools used to generate that format. For example, with spreadsheet tools, cut-

ting and pasting a column is easy, but is not as easy in an RDBMS environment. In an RDBMS, it is easy to remove rows according to a complex predicate, but not as easy in spreadsheet tool. Another example is that researches tend to create a spreadsheet for each run of an experiment or and observation session, whereas in a RDBMS they would not necessarily create a new table in these cases.

While in this work, identifying relationships between tabular datasets is informed by spreadsheets and the ways certain actions are likely to affect spreadsheet data, in the future, we plan to study the activities that scientists commonly perform on other scientific dataset formats and what connections and relationships they give rise to. Such a study will enable us to reveal whether or not the relationships that our methodology currently identifies are applicable to tabular datasets from other sources and whether we will need to define additional relationships.

8. CONCLUSIONS

As technological advances increase the scope and complexity of scientific research, managing data will remain a challenging and time-consuming task. Detecting such relationships as complementation, augmentation, and containment between datasets can enable scientists to work effectively with their data, improve their ability to understand their datasets, and reduce the likelihood of analyzing outdated or incomplete datasets. In this paper, we have demonstrated through the design, implementation, and evaluation of ReConnect that the relationship-identification methodology can help scientists decide how to utilize datasets stored in their spreadsheets. As the experimental results show, using ReConnect, scientists can quickly determine meaningful relationships that facilitate managing their research data.

9. ACKNOWLEDGMENTS

This work is supported by NSF award IIS 1064685 . We thank all our user study participants and our anonymous reviewers for their helpful and appreciated feedback.

10. REFERENCES

[1] Inter-university Consortium for Political and Social Research. http://www.icpsr.umich.edu.

[2] The Research Data Alliance. https://rd-alliance.org/.

[3] Z. Bellahséne, A. Bonifati, and E. Rahm. *Schema Matching and Mapping*. Springer, 2011.

[4] P. A. Bernstein and L. M. Haas. Information Integration in The Enterprise. *Communications of the ACM*, 51(9):72–79, 2008.

[5] C. Chambers, M. Erwig, and M. Luckey. SheetDiff: A Tool for Identifying Changes in Spreadsheets. In *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on*, pages 85–92. IEEE, 2010.

[6] T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining Database Structure; or, How to Build a Data Quality Browser. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 240–251. ACM, 2002.

[7] A. Doan, P. Domingos, and A. Levy. Learning Source Descriptions for Data Integration. In *WebDB (Informal Proceedings)*, pages 81–86, 2000.

[8] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, Jan. 2007.

[9] M. Fisher and G. Rothermel. The EUSES Spreadsheet Corpus: a Shared Resource for Supporting Experimentation with Spreadsheet Dependability Mechanisms. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–5. ACM, 2005.

[10] Florencesoft. DiffEngineX: Compare Excel Work Sheets, 2010. http://www.florencesoft.com/compare-excel-workbooks-differences.html.

[11] H. Gonzalez, A. Y. Halevy, C. S. Jensen, A. Langen, J. Madhavan, R. Shapley, W. Shen, and J. Goldberg-Kidon. Google Fusion Tables: Web-centered Data Management and Collaboration. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1061–1066. ACM, 2010.

[12] L. Gravano, P. G. Ipeirotis, N. Koudas, and D. Srivastava. Text joins in an rdbms for web data integration. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 90–101, New York, NY, USA, 2003. ACM.

[13] M. Hernández, R. Miller, and L. Haas. Clio: A Semi-automatic Tool for Schema Mapping. In *ACM SIGMOD Record*, volume 30, page 607. ACM, 2001.

[14] B. Howe, G. Cole, E. Souroush, P. Koutris, A. Key, N. Khoussainova, and L. Battle. Database-as-a-service for Long-tail Science. In *Scientific and Statistical Database Management*, pages 480–489. Springer, 2011.

[15] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed Data-parallel Programs from Sequential Building Blocks. *ACM SIGOPS Operating Systems Review*, 41(3):59–72, 2007.

[16] Jonathan Wyatt and Ewen Ferguson. Spreadsheets Risk Management: Frequently Asked Questions Guide. Technical report, Protiviti, 2011.

[17] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker. *The Data Warehouse Lifecycle Toolkit*. Wiley, 2011.

[18] R. Macefield. How to Specify the Participant Group Size for Usability Studies: A Practitioner's Guide. *Journal of Usability Studies*, 5(1):34–45, 2009.

[19] W. Michener, D. Vieglais, T. Vision, J. Kunze, P. Cruse, and G. Janée. DataOne: Data Observation Network for Earth-preserving Data and Enabling Innovation in The Biological and Environmental Sciences. *D-Lib Magazine*, 17(1):3, 2011.

[20] Microsoft. What You Can Do with Spreadsheet Inquire, 2013. http://office.microsoft.com/en-us/excel-help/what-you-can-do-with-spreadsheet-inquire-HA102835926.aspx.

[21] E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.

[22] XL-Consulting. Synkronizer: Compares Excel Files Faster than You Can, 2010. http://www.synkronizer.com/.

[23] D. Zardetto, M. Scannapieco, and T. Catarci. Effective automated object matching. *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 0:757–768, 2010.