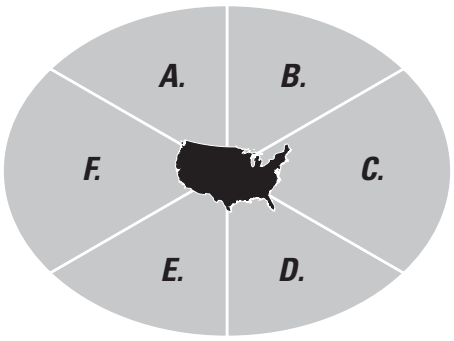
USGS
science for a changing world

Prepared in cooperation with the U.S. Environmental Protection Agency

Determining the Sources of Fine-Grained Sediment Using the Sediment Source Assessment Tool (Sed_SAT)

Open-File Report 2017–1062

U.S. Department of the Interior
U.S. Geological Survey



Cover. Photographs showing sources of fine-grained sediment: *A*, Lake Tahoe area, NV, *B*, Difficult Run, VA, *C*, Lake Tahoe area, NV, *D*, Linganore Creek, MD, *E*, Difficult Run, VA, and *F*, Linganore Creek, MD. Photos by Allen C. Gellis, U.S. Geological Survey.

Determining the Sources of Fine-Grained Sediment Using the Sediment Source Assessment Tool (Sed_SAT)

By Lillian E. Gorman Sanisaca, Allen C. Gellis, and David L. Lorenz

Prepared in cooperation with the U.S. Environmental Protection Agency

Open-File Report 2017–1062

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
RYAN K. ZINKE, Secretary

U.S. Geological Survey
William H. Werkheiser, Acting Director

U.S. Geological Survey, Reston, Virginia: 2017

Suggested citation:
Gorman Sanisaca, L.E., Gellis, A.C., and Lorenz, D.L., 2017, Determining the sources of fine-grained sediment using the Sediment Source Assessment Tool (Sed_SAT): U.S. Geological Survey Open File Report 2017–1062, 104 p., https://doi.org/10.3133/ofr20171062.

ISSN 2331-1258 (online)

Acknowledgments

We would like to thank USGS colleagues Anna Baker and Leah Kammel for testing earlier versions of the program as well as Shannon Jackson and Matthew Cashman for testing later versions, and Tanja Williamson, Durga Ghosh, and Karen Ryberg for their edits and suggestions.

Contents

Figures

Tables

Determining the Sources of Fine-Grained Sediment Using the Sediment Source Assessment Tool (Sed_SAT)

By Lillian E. Gorman Sanisaca, Allen C. Gellis, and David L. Lorenz

Abstract

A sound understanding of sources contributing to instream sediment flux in a watershed is important when developing total maximum daily load (TMDL) management strategies designed to reduce suspended sediment in streams. Sediment fingerprinting and sediment budget approaches are two techniques that, when used jointly, can qualify and quantify the major sources of sediment in a given watershed. The sediment fingerprinting approach uses trace element concentrations from samples in known potential source areas to determine a clear signature of each potential source. A mixing model is then used to determine the relative source contribution to the target suspended sediment samples.

The computational steps required to apportion sediment for each target sample are quite involved and time intensive, a problem the Sediment Source Assessment Tool (Sed_SAT) addresses. Sed_SAT is a user-friendly statistical model that guides the user through the necessary steps in order to quantify the relative contributions of sediment sources in a given watershed. The model is written using the statistical software R (R Core Team, 2016b) and utilizes Microsoft Access® as a user interface but requires no prior knowledge of R or Microsoft Access® to successfully run the model successfully. Sed_SAT identifies outliers, corrects for differences in size and organic content in the source samples relative to the target samples, evaluates the conservative behavior of tracers used in fingerprinting by applying a "Bracket Test," identifies tracers with the highest discriminatory power, and provides robust error analysis through a Monte Carlo simulation following the mixing model. Quantifying sediment source contributions using the sediment fingerprinting approach provides local, State, and Federal land management agencies with important information needed to implement effective strategies to reduce sediment. Sed_SAT is designed to assist these agencies in applying the sediment fingerprinting approach to quantify sediment sources in the sediment TMDL framework.

Introduction

Identifying the sources of fine-grained instream sediment flux in a watershed is an essential part of developing total maximum daily load (TMDL) management strategies designed to reduce suspended sediment in streams (Gellis, Noe, and others, 2015). Fine-grained silt and clay are of particular concern as they are a source of habitat degradation, affect water supply intakes and reservoirs, and commonly carry pollutants (Larsen and others, 2010).

The sediment fingerprinting method uses trace element concentrations from fine-grained sediment samples (<63µm) in potential source areas in a watershed to determine a clear signature or "fingerprint" for each potential source through a series of computational and statistical steps (Gellis and Walling, 2011). Potential target samples include but are not limited to suspended sediment, bed, passive, and floodplain deposits (Miller and others, 2015). Tracer concentrations found in target samples can then be compared to the source fingerprints through a mixing model to determine relative source contributions (Collins and others, 2010; Gellis, Noe, and others, 2015). This approach allows us to answer the important question of whether the fine-grained sediment in target samples is derived from upland areas (such as agricultural fields, forests, or construction sites) or the channel (e.g. streambanks) sources, allowing for more effective, targeted management strategies (Gellis, Noe, and others, 2015).

Sed_SAT

The Sediment Source Assessment Tool (Sed_SAT) (Gorman Sanisaca and others, 2017) was developed as a user-friendly interface that steps through the challenging and time-intensive computational procedures involved in applying the sediment fingerprinting method to quantify the relative contribution of watershed-derived sediment sources. The systematic structure and easy-to-use push-button interface in Sed_SAT

make the sediment fingerprinting method more accessible as it can be applied more easily and obtain results more quickly. All computations and statistics are run in the open source statistical software R (R Core Team, 2016b), version 3.3.0, while the interface is based in Microsoft Access® and applies Visual Basic for Applications®.

Sed_SAT was developed in cooperation with the U.S. Environmental Protection Agency's (EPA) Regions 3 and 5 and is a companion to the EPA Manual, "A Manual to Identify Sediment Sources" (Gellis, Fitzpatrick, and others, 2016). Reading the EPA manual before engaging in a sediment source study using Sed_SAT is strongly recommended. Chapter 6 of the manual includes guidance on how to collect, prepare, and analyze source and target sediment samples for sediment source identification using the sediment fingerprinting method.

Navigating the Instruction Manual

The instruction manual was designed to be an interactive document that can be accessed and searched from within the Sed_SAT program. All details on the procedures completed at each step are found in the body of the manual. Screenshots and details on navigating each screen are located in the appendixes. Important options and elements of each screen are highlighted with colored boxes that correspond to the bullets below the screenshot.

The InstructionManual.pdf file is generated by InstructionManual.docm, a macro-enabled Microsoft Word® document that can be accessed from every screen in the program. Clicking the Instruction Manual hyperlink below the Program MAP icon (fig. 6) opens the InstructionManual.pdf file to the section of the manual that gives background on the current step, and clicking the Screen Help hyperlink opens the manual to the appendix that gives details on navigating the current screen. The InstructionManual.pdf file can only be accessed from within the program if the path to AcroRd32.exe has been set by the user in the first step of the program setup on the Set PATHs screen (see *Set PATHs*) and the content has been enabled in the InstructionManual.docm file (see *Enable Content*.)

Downloading Sed_SAT

Minimum System Requirements

1. Microsoft Windows® 7 or above
2. Microsoft Office® 2007–13 (32 bit)
    i. Microsoft Access®
    ii. Microsoft Word®
    iii. Microsoft Excel®
3. Adobe Acrobat Reader® (AcroRd32.exe)

The program can be run on either a 32-bit or 64-bit operating system. The program has not been tested using Parallels Desktop® for Mac®.

Download all files and folders in the Sed_SAT repository on USGS BitBucket, located at https://my.usgs.gov/bitbucket/projects/SED/repos/sed_sat/, into a parent directory that does not contain any spaces or special characters. DO NOT rename folders, subfolders, or files as this will cause program failure. The file structure is shown in *appendix 1*.

Getting Started

When starting Sed_SAT for the first time, the user will need to follow the Enable Content instructions and the Activating Microsoft Access Visual Basic References instructions if an error were to occur when enabling content. This is a one-time process for each initiation of the model from a given location on the user's system.

Enable Content

Prior to running the model, you must enable content in the files that contain Visual Basic for Applications® code and (or) macros. The following files contain macros:

- Instruction Manual.docm (may need to Enable Editing first)
- Blank.accdb
- Sed_SAT.accdb

Failing to complete this step will result in program failure. Files containing Visual Basic for Applications® code must be listed as "trusted documents" before any of the code can run. This adds protection for the user by preventing the code from running upon opening a document before the user decides to trust the document. For more information on the enable content warning see Enable Content Warning (Microsoft®, 2007). To list documents as "trusted documents" complete the following steps:

1. Enable content in the SedimentFingerprinting_R\Manual\InstructionManual.docm file
    a. If open, close the InstructionManual.pdf file (it will be updated by the InstructionManual.docm file)
    b. Open the SedimentFingerprinting_R\Manual\InstructionManual.docm file
    c. Click "Enable Content" in the yellow ribbon on the top of the widow (the file will close itself)
2. Enable content in the Blank.accdb file (fig. 1)
    a. Start by opening the Blank.accdb file. This file contains no information. It acts only as a springboard to send the user back to the core file Sed_SAT.accdb

b. Close the pop-up window by clicking the "X" in the top right corner

c. Click "Enable Content" in the yellow ribbon on the top of the widow (if an error occurs after enabling content, this is due to a missing Visual Basic for Applications® reference; see *Activating Microsoft Access Visual Basic References* to add required Visual Basic for Applications® references)

d. Click "Stop" on the bottom of the pop-up (this will close the Blank.accdb file)

3. Enable Content in the Sed_SAT.accdb file

a. Open the Sed_SAT.accdb file

b. Close the Pop-up window by clicking the "X" in the top right corner

c. Click "Enable Content" in the yellow ribbon on the top of the widow (if an error occurs after enabling content, this is due to a missing Visual Basic for Applications® reference; see *Activating Microsoft Access Visual Basic References* to add required Visual Basic for Applications® references)

d. All content is now enabled, and the program is ready to run

Enabling content is a one-time process as long as the program remains in one location. Each time the program is moved to a new location you will need to enable content in the new location in all three files in the order above.

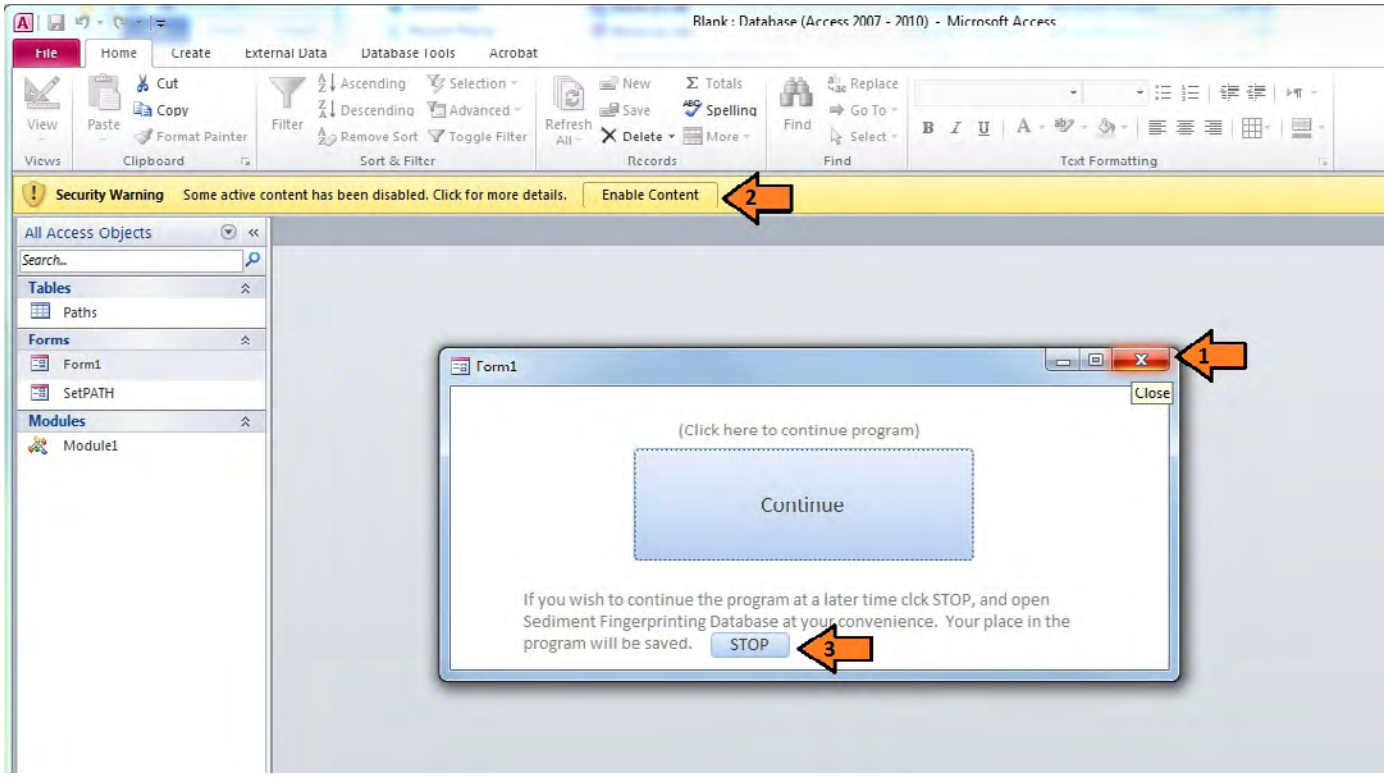


Figure 1. Steps to enable content in the Blank.accdb file.

Activating Microsoft Access Visual Basic References

The Sed_SAT user interface is written in Visual Basic for Applications® (VBA) and therefore requires some Visual Basic for Applications® references. If a reference is missing, the user will see an error when enabling content. The following references are required:

1. Visual Basic for Applications®
2. Microsoft Access® 15.0 Object Library (15.0 is for Microsoft Access® 2013, 14.0 and 12.0 are used for Microsoft Access® 2010 and 2007 respectively)
3. OLE Automation
4. Microsoft Office® 15.0 Access® database engine Object Library (15.0 is for Microsoft Access® 2013, 14.0 and 12.0 are used for Microsoft Access® 2010 and 2007 respectively)
5. Microsoft Word® 15.0 Object Library (15.0 is for Microsoft Word® 2013, 14.0 and 12.0 are used for Microsoft Word® 2010 and 2007 respectively)

If an error occurs after enabling content, the Visual Basic® Editor window will open displaying a message that indicates a missing Visual Basic for Applications® reference. If no error occurs, skip this section and continue to *Preparing Data for Sed_SAT*. To add missing references, follow these steps:

1. Click “OK” to close the error message (fig. 2)
2. Click “Run”->”Reset,” which will close the VBA Editor (fig. 3)
3. Close the open form window by clicking the X in the upper right corner (fig. 4)
4. Press Alt+F11 to open the Visual Basic® Editor
5. Click “Tools”->”References” (fig. 5)
6. Uncheck any references that start with “MISSING”
7. Select the required Visual Basic for Applications® references listed above
8. Click “OK”
9. Close and reopen the database

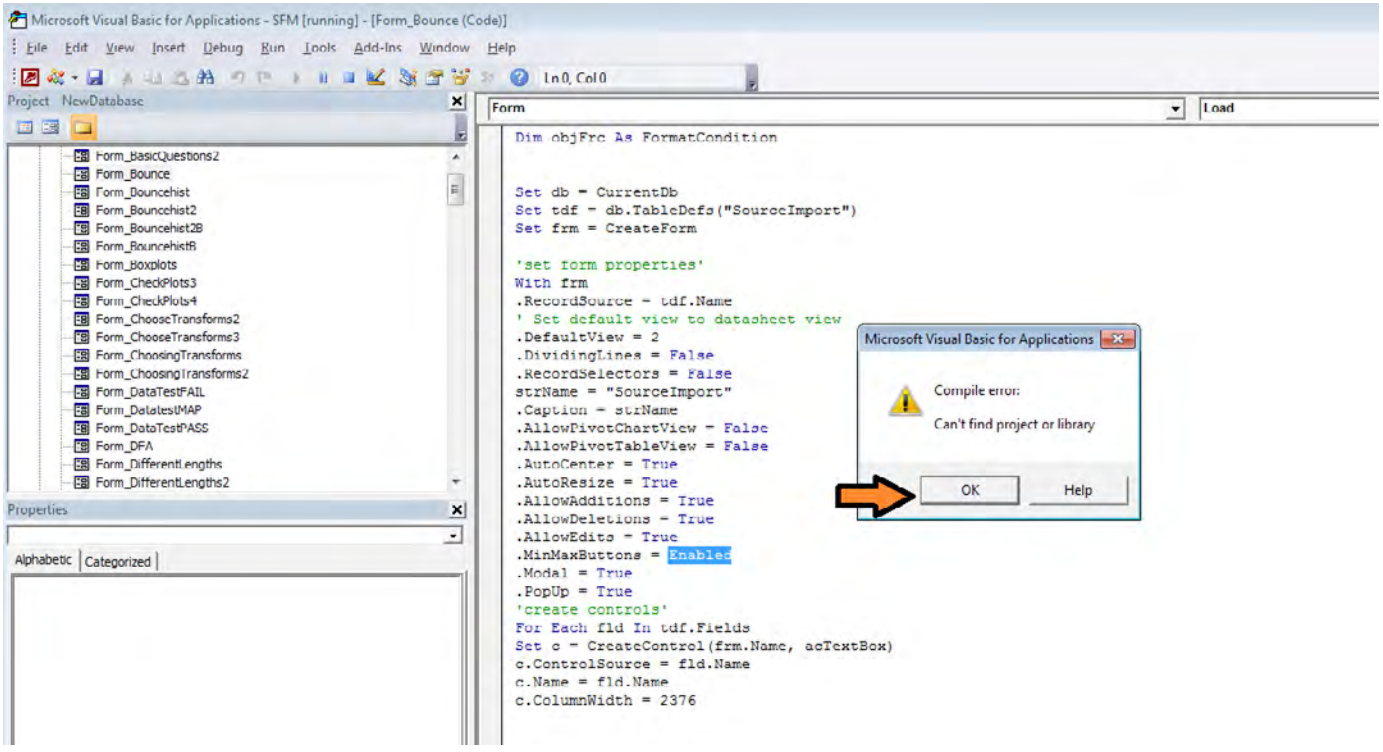


Figure 2. Step 1 in activating acquired Visual Basic for Applications references.

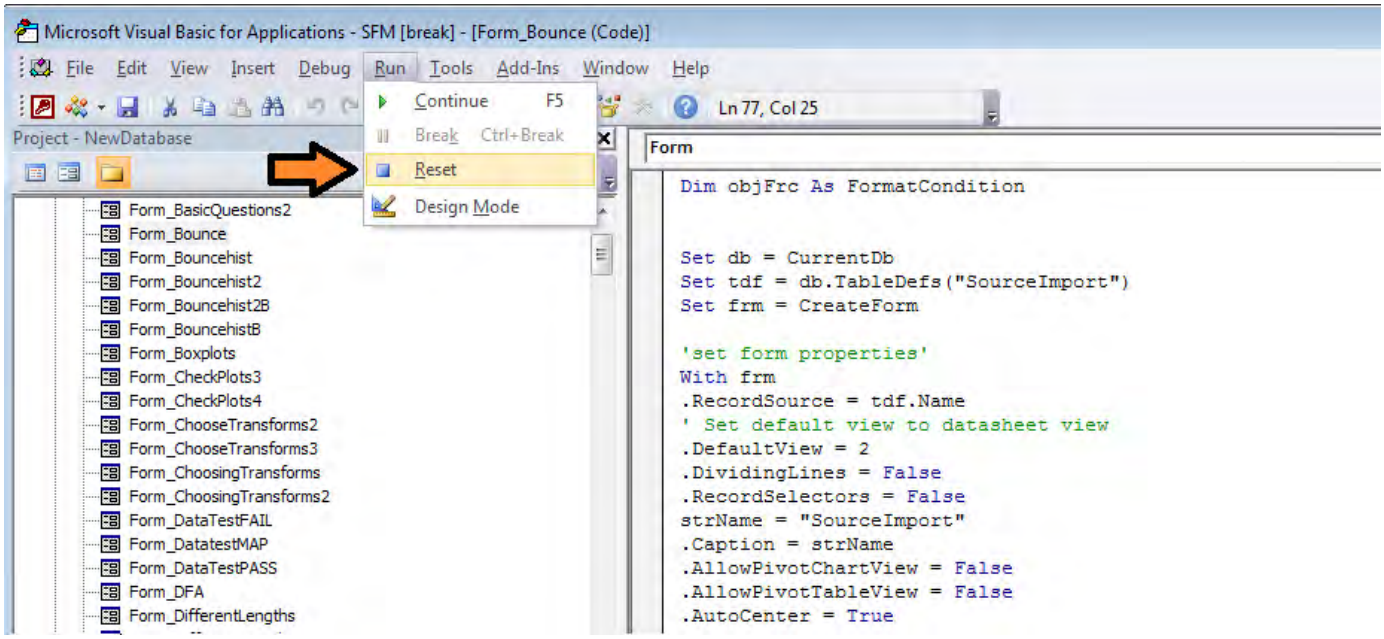


Figure 3. Step 2 in activating required Visual Basic for Applications references.

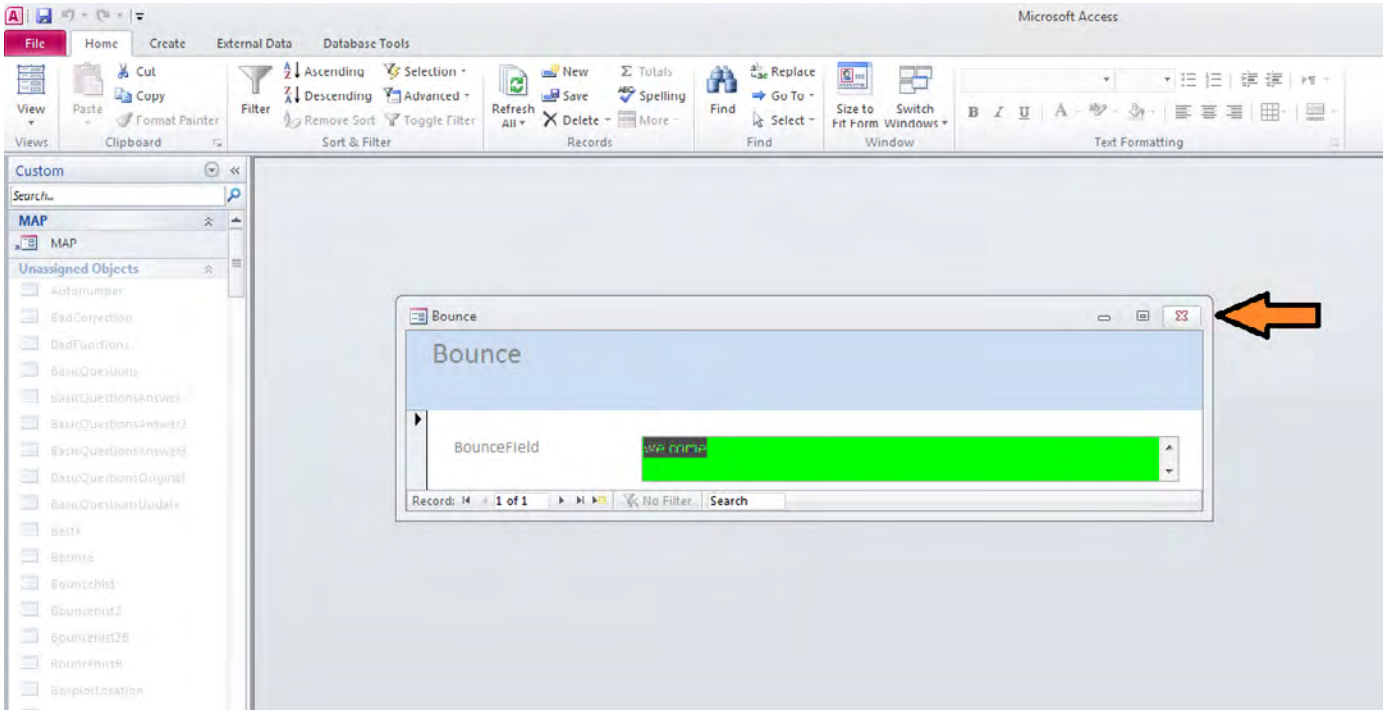


Figure 4. Step 3 in activating required Visual Basic for Applications references.

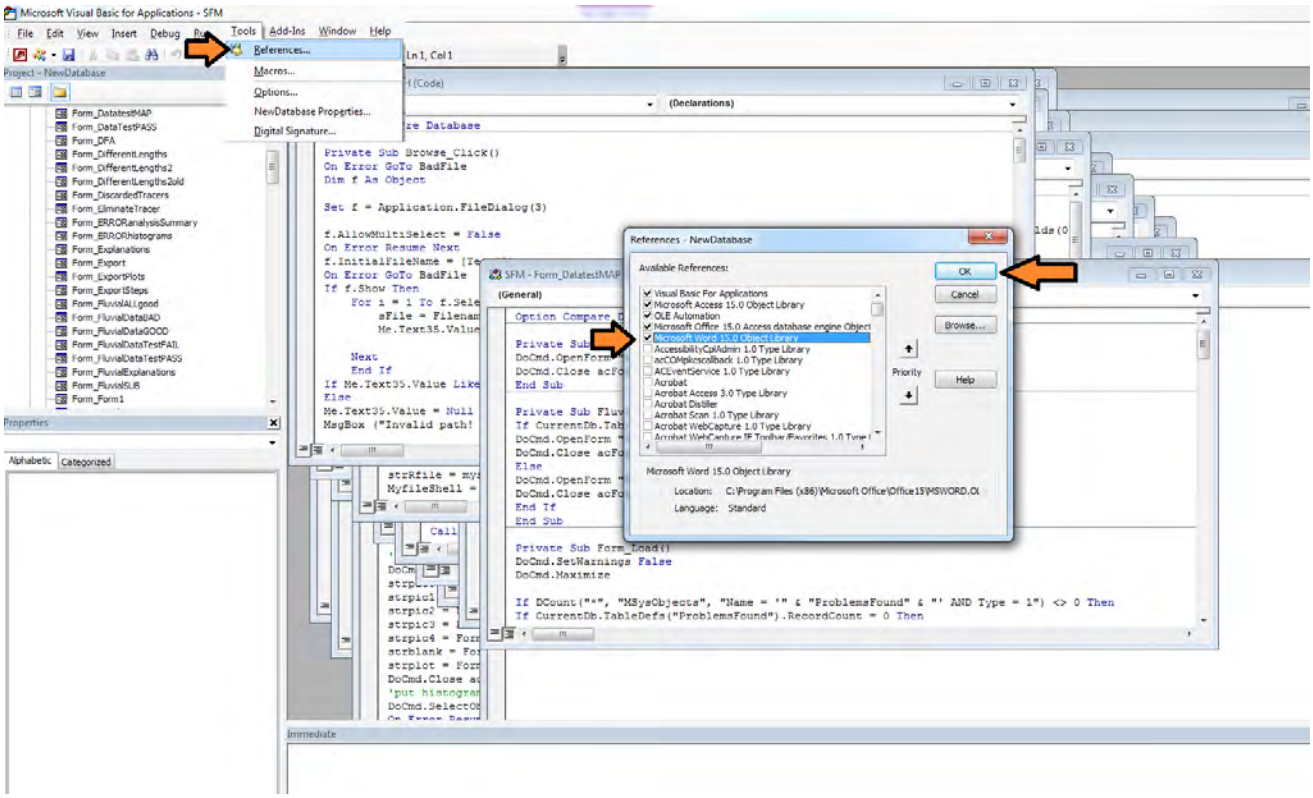


Figure 5. Steps 5–8 in activating required Visual Basic for Applications references.

Preparing Data for Sed_SAT

In order for Sed_SAT to function, the data *must* be properly formatted (links to example datasets given in the *Example Datasets* section). The specific text identifying column/field headers is unimportant, but the relative location of columns in the source and target datasets must be identical and is a core requirement for the program to run successfully. Column/field headers should be unique; no field name should match another field name within the same dataset. Field names must not contain special characters "():;%&*$#@!?><{}[]+=\~/|". Detailed formatting rules are given as part of the program when you click START on the Welcome Screen.

Example Datasets

The example datasets provided can be run through the program for practice purposes and (or) used as a guide to properly format import datasets. Links to the datasets are given below. The files are located in the "ExampleDatasets" folder and come preloaded in the program. However the user can practice importing the datasets from any location. The example datasets include size data (Surface_Area), organic content data (Total Organic Carbon, TOC), elemental analysis (metals), stable isotopes (δ13C, δ15N), and radionuclides (^{137}Cs, excess ^{210}Pb). The source dataset contains censored values (nondetects) flagged by missing values. Negatives and zeros are true values not nondetects. These datasets are examples only and are not meant to restrict the types or the number of tracers that can be analyzed, the type of nondetect flag, or the file type. See *Import Source and Target Data* for supported file types. Minor formatting errors are included in the example datasets; these errors are designed to trigger the *Data Testing Module* screens. These are all acceptable formatting errors that can be fixed within the program without having to alter the original files.

The column headers for target, source, and reporting limit data do not have to match exactly, but the tracers *must* be in the same relative location. For example, the fifth column in the example source data has the header "Al," but the header of the fifth column in the example target data is "Aluminum." Because both column headers signify that the tracer is aluminum, there will be no problems analyzing the data.

> ***Example Datasets Hint***
>
> Throughout the instruction manual relevant information on processing the example datasets within a given step of the program will be given in an "Example Datasets Hint" box.

See *appendix 2* for more details on the columns in the example datasets.

Data Formatting Rules

1. The sample name or ID must be in the first column for both the *source* and *target* datasets. This column MUST NOT contain missing values (i.e. blank or null). Sample names must be unique; no two samples can have the same name. Sample names should not contain special characters, "():;%&*$#@!?><{}[]+=\~/|-".

2. The source type (for example, bank or cropland) is required in the *source* dataset and must be in the second column. In the *target* dataset, the second column should contain the sample type (for example, suspended or bed material); however, if sample type information is not available, the user can include a column with "target" repeated for each sample as the second column. The second column is REQUIRED in BOTH the *source* and *target* datasets, and no missing values are permitted.

3. If your dataset contains size data (grainsize or surface area), it must go in the third column (this column may contain some but not all missing values.) If your dataset does not contain size data, but contains organic content data (either Total Organic Carbon [TOC] or Loss on Ignition [LOI]), it must go in the third column. TOC cannot contain missing values, but missing values are permitted in LOI. For more information on size and organic content data see *appendix 3*.

4. If your dataset contains both size and organic content data, organic content data must go in the fourth column. See *appendix 3* for more details on how size and organic content data are treated in the program.

5. Tracers, with the exception of TOC, may contain missing values in the *source* dataset only if the missing values indicate values below the reporting limit of the tracer (nondetects). Nondetects are not permitted in the *target* dataset and must be addressed by the user prior to importing the *target* dataset.) Nondetects in the *source* dataset can be estimated within the program using an imputation procedure in the first part of the program.

6. Tracers MUST be in the same relative position in both the *source* and *target* datasets. Results will be meaningless if this *requirement* is not met.

7. Text is not allowed in the tracer data unless it indicates a nondetect (i.e. "<0.6") in the *source* dataset ONLY. There should be only numerical data in the *target* dataset tracer columns.

8. There must be at least two tracers without nondetects.

9. There should be no completely blank columns or rows. For example if importing data from Microsoft Excel, columns/rows should be deleted not cleared.

10. There MUST be >3 samples per *source* group (recommended >10).

11. All samples MUST be unique; replicates/duplicates should be evaluated and removed prior to importing data.

Navigating Sed_SAT

Screen Resolution

Sed_SAT was designed with screen resolution set to 100 percent, but it will function at any screen resolution. If at any point you cannot view the current screen, simply resize the window or use the scroll bars at the bottom and left edges of the screen to see all content. If you are unsure whether or not all content is visible, refer to the appendixes for screen details.

Program MAP

Sed_SAT is designed to run in a linear fashion from beginning to end. The Program MAP (fig. 7) is designed to allow the user to navigate steps, view data tables/results that have already been completed, and (or) start running the model at a previously completed step. If this the first time the program is run, the user will see a Welcome Screen; to start the program, click "Start" or view the Program MAP. The Program MAP can be accessed from any screen in the program by clicking the Program MAP icon (fig. 6).

The Program MAP is NOT designed to allow the user to skip steps. Skipping steps using the Program MAP will at best cause the program to fail and at worst cause corrupted results. Throughout the normal progression of the program, however, opportunities to skip various steps will be made available. Skipping steps using this method is acceptable and will yield appropriate results.

Figure 6. Program MAP icon.

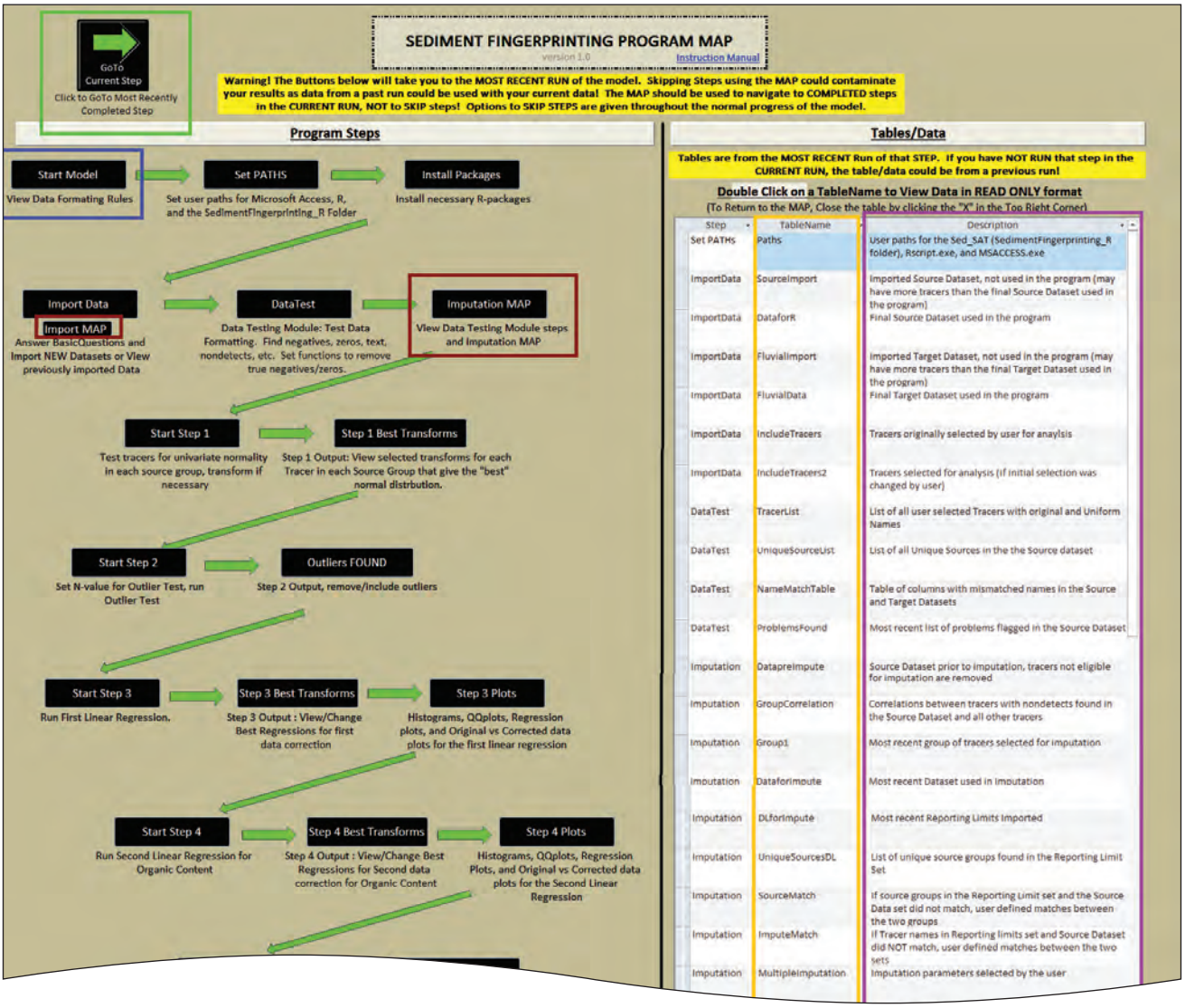
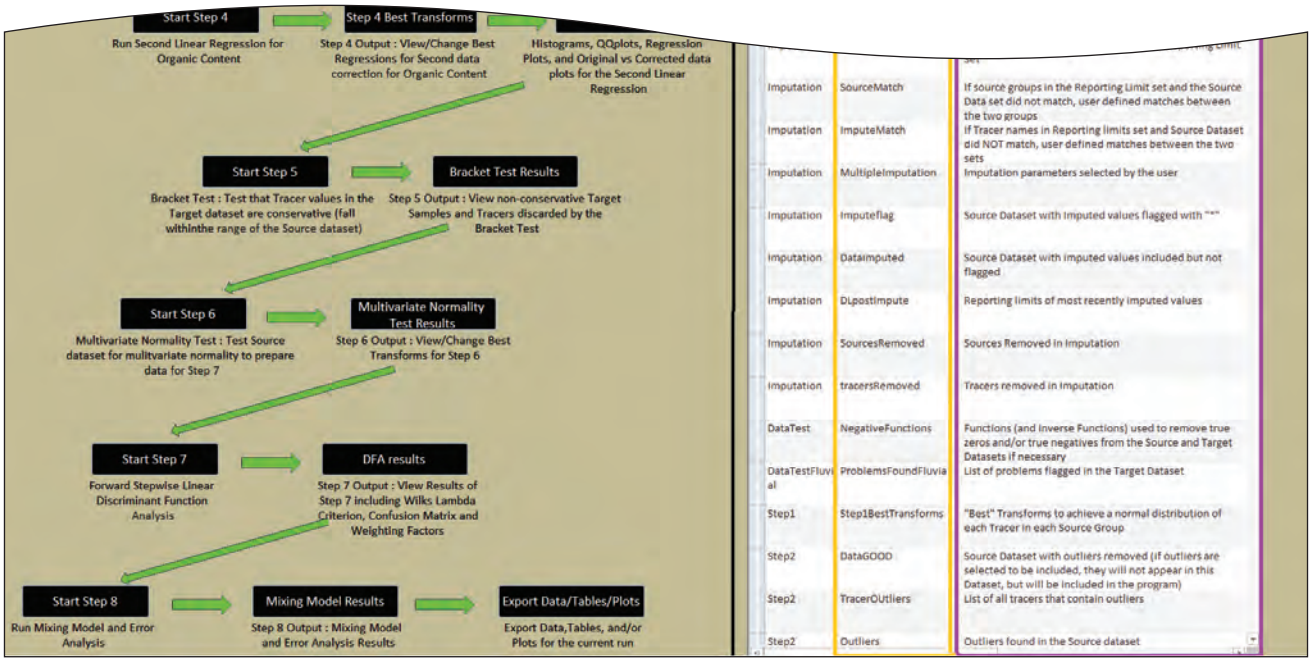


Figure 7. Sed_SAT Program MAP. (Submaps can be accessed from the main Program MAP and are shown in figures 8, 9, and 10.)



EXPLANATION

Click to navigate to the most recently completed step in the program

Click the black buttons to navigate to a given step

Click to navigate to submaps (Import MAP, fig. 8, and Imputation MAP, fig. 9)

Output table descriptions

Double-click TableName to open a read-only version of the table

Figure 7. Sed_SAT Program MAP. (Submaps can be accessed from the main Program MAP and are shown in figures 8, 9, and 10.)—Continued

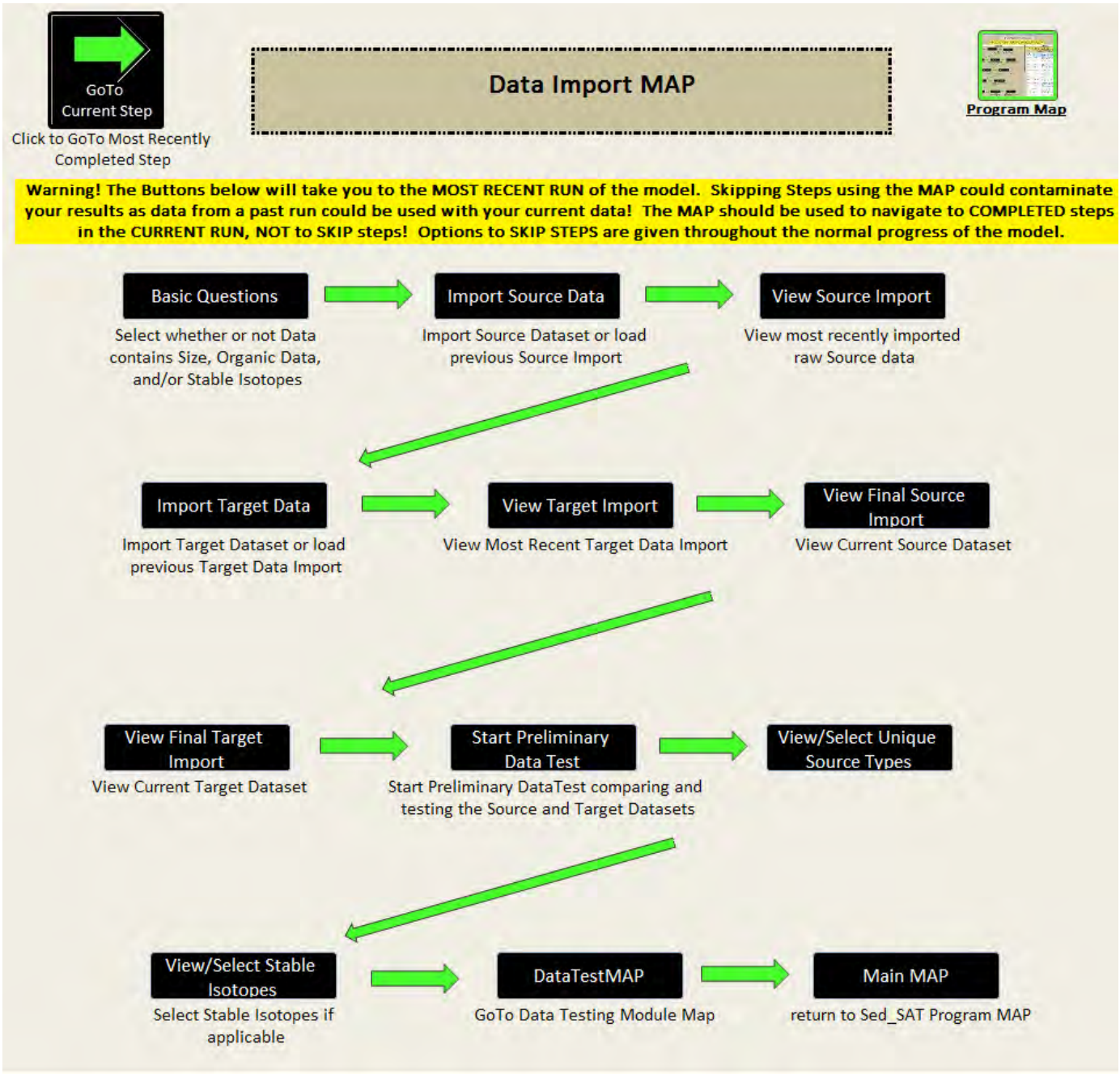


Figure 8. Map of steps to import data into Sed_SAT.

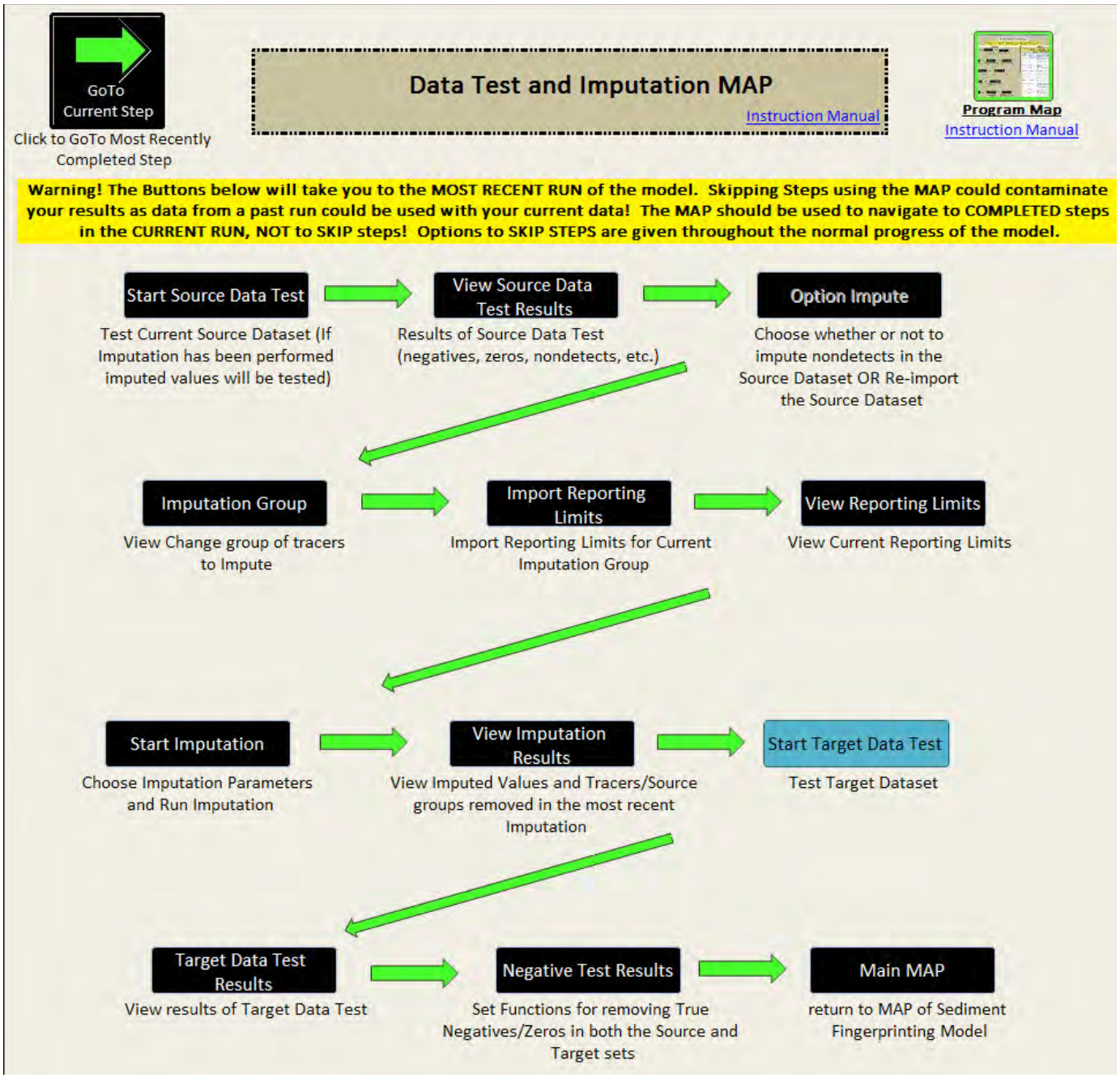


Figure 9. Map of steps in the *Data Testing Module*. This map can be accessed through the Import MAP shown in figure 8.

Buttons

As the program runs, each step's output is updated and overwritten. All buttons that run an Rscript that overwrites data are Teal, and all buttons that simply navigate to an output are Black. This color scheme allows the user to differentiate between buttons that will alter/delete data and that shouldn't be run out of order from buttons that simply navigate.

The Blank File

The Blank.accdb file is a mostly empty database with the sole purpose of returning the user to the primary Sed_SAT.accdb file. This allows the Sed_SAT.accdb file to be updated by the Rscripts being run. The only screen in the Blank.accdb file is shown below in figure 10.

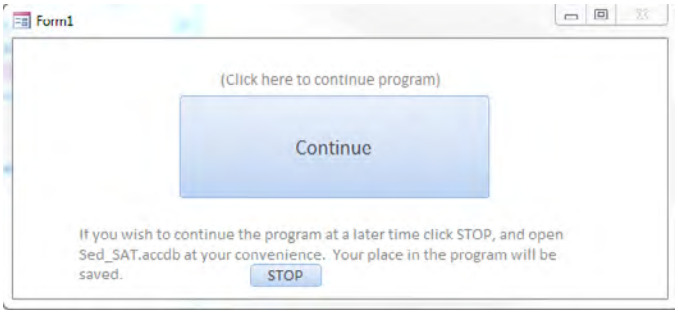


Figure 10. The only screen in the Blank.accdb file.

Set PATHs

Before any data can be imported or analyzed, paths to all required programs, and the path to the SedimentFingerprinting_R folder must be confirmed for the computer being used. The program will attempt to detect all required programs, but if any path is blank or incorrect the user must enter it. Using the Browse buttons is highly recommended to avoid program failure due to typing mistakes. The Sed_SAT program must be located in a path with no spaces in the parent directories. If spaces exist in the path to Sed_SAT the user will be prompted to move the program and repeat the *Enable Content* steps. At the end of the first run of the program, the user will have the opportunity to export the "Paths" table either to Microsoft Excel® or Microsoft Access® (see *Export Data/Tables/Plots*); this can be imported in subsequent program runs at the Set PATHs step to avoid browsing for paths every time. Examples of paths are given below the path input boxes on the Set PATHs screen as a guide to where programs may be located on the hard drive.

Sed_SAT is designed for use on both 64-bit and 32-bit operating systems as long as 32-bit Microsoft Access® is installed, but on a 64-bit operating system special care must be taken when selecting paths to Microsoft Access®.

1. Path to 32-bit MSACCESS.EXE: this is where Microsoft Access® is located on the hard drive.
    a. For 32-bit computers, browse to the MSACCESS.EXE file in the Program Files under Microsoft Office®.
    b. For 64-bit computers, browse to the MSACCESS.EXE file in the Program Files (x86) under Microsoft Office®.
2. Path to SedimentFingerprinting_R folder: this is where the SedimentFingerprinting_R folder was saved. Just navigate to the folder's location and select the folder.
    a. Spaces in the parent directory are NOT permitted.
    b. Same for both 32-bit and 64-bit
3. Path to 32-bit AcroRd32.exe: this is where Adobe® Acrobat Reader® is located on the hard drive.
    a. For 32-bit computers, the file is located in the Program Files folder.
    b. As of 2016, Adobe Acrobat Reader is available only as a 32-bit program, but should that change, the user of a 64-bit operating system should take care to give the path to the 32-bit version of the program located in the Program Files (x86) folder.

Do not try to continue past the SetPATHs screen if you have not enabled content in the Blank.accdb file (see *Enable Content*). This will cause the program to fail. For more details on navigating the SetPATHSs screen see *appendix 4*.

R Packages

All of the various statistical tests and calculations in Sed_SAT are run using the open source statistical software R (R Core Team, 2016b), version 3.3.0, which is included as part of the Sed_SAT program. Regardless of which versions of R the user has installed on the computer being used, only the version of R included in Sed_SAT will be invoked. Working versions of all required R-packages are included in the SedimentFingerprinting_R/R-3.3.0/library folder. Packages are free libraries of code written by the community of active R users that allow for additional functionality in specific topics (2015a; Hornik, 2016). Information on the R-Packages included in Sed_SAT can be found in *appendix 5*.

Import Data

Import steps are outlined on the Import MAP (fig. 8). The Import MAP can be opened using the button below the Import Data button on the *Program MAP* (fig. 7).

Basic Questions

Two basic questions that dictate the path the program will take are:

1. Does the dataset include size data?
2. Does the dataset include organic content data
   a. What type of organic content data (Loss on Ignition or Total Organic Carbon)?
   b. Does the dataset include stable isotopes (not corrected for organic content)?

It is important that the user answer the questions according to which type of data are included in the dataset. If the dataset contains organic content data, a second window will appear asking for the type of organic content data. On some systems the second box may appear behind the Basic Questions screen. If this occurs, simply click on the second screen to enter the organic content type. Do not use the basic questions to skip corrections for size or organic content (*Start Step 3: First Linear Regression* and *Start Step 4: Second Linear Regression for Organic Content*) if the dataset does in fact contain these fields.

> ***Example Datasets Hint***
>
> The example datasets contain both size and organic content data (Total Organic Carbon).

To view additional information about how size and organic content data are utilized within the program, click the  on the Basic Questions screen or see *appendix 3*.

Organic Content and Stable Isotope Data

Selecting "Yes" to the question "Does your dataset include organic content data?" will trigger a pop-up asking the user to select the type of organic content data that appears in the dataset and whether or not the dataset contains stable isotope data. Because Loss On Ignition (LOI) is expressed as a percentage rather than as an actual concentration value, it is treated differently in comparison to Total Organic Carbon (TOC) in that it is used only for the purpose of applying correction factors for organic content in *Start Step 4: Second Linear Regression for Organic Content* and not a tracer used as part of the "fingerprint." This is why missing values are permitted in LOI but not in TOC.

Stable isotopes should not be corrected in *Start Step 4: Second Linear Regression for Organic Content* for organic content as the concentrations are affected by relative proportions of organic content; therefore, correcting these tracers would introduce bias into the dataset (Gellis, Noe, and others, 2015). If your dataset contains both organic content and stable isotopes, you will be prompted to select which tracers are stable isotopes after your datasets are imported. These tracers will not be corrected for organic content. See *appendix 6* for details on the Stable Isotopes screen.

> ***Example Datasets Hint***
>
> The example datasets contain the stable isotopes Delta_15N and Delta_13C_Organic.

Import Source and Target Data

First ensure that your source and target datasets are formatted according to the *Data Formatting Rules*. To import data, Browse to the file you want to import. Supported file types include .accdb, .xls, .xlsx, and .csv. For Microsoft Excel® files, select the sheet to import; for Microsoft Access® files, select the table to import; and for CSV files, select the separator for the fields. The user can also choose to use a previously imported source and (or) target dataset rather than importing a new dataset.

> ***Example Datasets Hint***
>
> The example datasets come preloaded in the program and therefore the user does not need to import them. To run the example datasets click "Use Previously Imported Source Data."

For details on the Import Data screens see *appendix 7*.

Data Testing Module

The first step after the datasets have been imported is to verify that the datasets are formatted correctly according to the *Data Formatting Rules*. Both the source and target datasets are tested and cannot be edited within the program; this is to avoid accidental edits that could cause program failure. See figure 9 for a map of the steps in the Data Testing Module. Potential problems flagged by the Data Testing Module include:

Problem 1: Number of fields in the source dataset ≠ number of fields in the target dataset.

Problem 2: Tracers not in the same relative location (column) in both the source and target datasets. For example, if Aluminum is in the fourth column in the source dataset it cannot be in the fifth column in the target dataset.

Example Datasets Hint

Problem 2 is triggered because the column names in the source data and target data datasets are different. But because they refer to the same quantity, simply check the box that indicates that columns are a match.

Problem 3: Non-unique sample names. Each sample in both the source and target datasets MUST have a unique name.

Problem 4: Non-unique source types. Text indicating unique source types must be exactly the same text (including case) throughout the source dataset (i.e. bank and BANK will be seen as different).

Problem 5: Less than three samples per source group. Recommended >10 samples per source group.

Example Datasets Hint

Problem 4 exists in the Source Data and Multiple Reporting Limit example datasets. Assume that "bank" is a typing mistake and should be "BANK."

Problem 6: Number of Tracers < number of Sources. There must be more tracers than sources for the program to function.

Problem 7: Missing/null/blank values

- In the source dataset, missing values are permitted only in size data, LOI data, or if the missing values indicate nondetects. Missing values that are not in size data or LOI data that are not nondetects are not permitted. Missing values in the *Example Datasets* are to be considered nondetects.

Example Datasets Hint

Problem 7 will be triggered by the example datasets. Assume that all missing values indicate censored data or nondetects.

- In the target dataset, missing values are permitted only in size data or LOI data.

Problem 8: Zeros and negatives

- Although true negatives/zeros are permitted in both the source and target datasets, the Data Testing Module will flag these values as potential problems.

Example Datasets Hint

Problem 8 will be triggered by the stable isotope data in the example datasets. These are "true negatives" and "true zeros."

- Negatives and (or) zeros that are not true concentration values but indicators for nondetects are permitted in the source dataset ONLY.

Problem 9: Text in tracer data (i.e. "<")

- Text in tracer data columns is permitted in the source dataset as long as the text is a flag for a nondetect. Text indicating a nondetect in the target dataset is not permitted.
- Text that does not indicate a nondetect is not permitted in tracer data columns in the source datasets.
- Text is not permitted in size or organic content data in either the source or target datasets.

See *appendix 8* for details on navigating the Data Testing Module.

Nondetects

A nondetect or censored value is defined as a value between zero and the reporting limit of the laboratory instrument used to measure concentration (Helsel, 2012). All nondetects must have the same indicator/flag per tracer (i.e. missing/null, text, negatives, or zeros) in the imported source dataset. Flagged nondetects are permitted in the source dataset only. Different tracers can have different nondetect indicators, but the indicator must be consistent within each tracer column. The program imputes nondetects only in tracer columns, not

in size data or organic content data. For more information on how nondetects are processed, refer to the discussion below, *Imputation of Nondetects in Source Data*. The Data Testing Module will search for values that are missing, text, negatives, or equal to zero, and prompt the user to input an explanation for their occurrence. If the values flagged are nondetects, select "Nondetect" as the "User Explanation."

"True Negative" and "True Zero" Values

"True negative" and "true zero" values are not treated differently from positive values throughout the program because they cannot be transformed using all transformations and remain in the set of real numbers (i.e. the square root of a negative number is not in the set of all real numbers). A value is considered a "true zero" or "true negative" if its value is truly equal to zero or a negative number; therefore, by definition, nondetects are not true negatives or true zeros. For example, true negative and true zero values can be found in δ13C and δ15N stable isotope data. For more information on how true negatives and true zeros are handled by the program, refer to *Start Step 1: Test for Univariate Normal Distributions* and *Start Step 3: First Linear Regression*. The Data Testing Module will search for all negative and zero values and prompt the user for an explanation for their occurrence. If the values are true negatives or true zeros, select the appropriate explanation.

> ***Example Datasets Hint***
>
> The stable isotope data in the example datasets contains "true negatives" and "true zeros."

Imputation of Nondetects in Source Data

Basics

What is a Nondetect?

A nondetect is a censored value reported to be below the reporting limit of the instrument used to measure tracer concentration. Nondetects are not zero values, however, and by definition the value of a nondetect falls somewhere between zero and the reporting limit (Helsel, 2012).

What is a Reporting Limit?

Reporting limits can be either quantitation limits or detection limits (Helsel, 2012). The quantitation limit is the value below which a single numerical value cannot be reported with confidence (Helsel, 2012). The detection limit is the value below which no concentration can be detected by the instrument (Helsel, 2012). Using either the detection limit or the quantitation limit as the reporting limit is acceptable as long as the choice remains consistent throughout the dataset (Helsel, 2012). In other words, all the tracers used in the analysis of the data must have the same type of reporting limit.

Imputation

Single and multiple imputation are methods for estimating values for nondetects based on the multivariate structure of the existing data and the reporting limits of the nondetects (Little and Rubin, 1987; Lubin and others, 2004). Single and multiple imputation are both presented as choices for imputing nondetects in the source dataset. For more details on single versus multiple imputation, see the discussion *Imputation Parameters* below. Because imputation is a multivariate procedure that considers not only the distribution of individual tracers but also the relationships between the tracers (Little and Rubin, 1987; Lubin and others, 2004), it can only be applied to a dataset, not a univariate vector (Palarea-Albaladejo and Martin-Fernandez, 2015). In Sed_SAT, the source dataset can be imputed for each source type. Each source type is assumed to be unique and unrelated to the other source types; therefore, the tracer relationships should be maintained and imputation can be run. Similarly, the target dataset cannot be imputed because each sample is considered unique and therefore independent. It is up to the user to decide how to manage any nondetects in the target samples prior to importing the data into Sed_SAT.

If the source dataset contains nondetects, values can be imputed using the zCompositions R-package (Palarea-Albaladejo and Martin-Fernandez, 2014) prior to *Start Step 1: Test for Univariate Normal Distributions*. zCompositions is a package designed to impute values that are below the reporting limit in compositional and subcompositional data, and as such it can only be used to impute values in the portion of the dataset that can be considered compositional or subcompositional (Palarea-Albaladejo and Martin-Fernandez, 2014). Compositional data are data whose components can be considered parts or percentages of a whole (Aitchison, 1986). Subcompositional data constitute a subset of compositional data and can be analyzed in the same way as compositional data (Aitchison, 1997). More information on the functions used for imputation can be found in Imputation Parameters below.

Imputation is the only option built into the program to estimate values for nondetects in the source dataset. To run the program without using imputation, all nondetects will have to be estimated or removed (not recommended) by the user prior to importing the source dataset.

Imputation MAP

This Imputation MAP (fig. 9) gives the details of the steps in the imputation process. These steps should not be run out of order, as doing so will cause the program to fail.

Selecting Tracers for Imputation

The first step in the imputation procedure is the selection of the group of tracers that will be used to impute nondetects. At least two tracers without nondetects and one tracer with nondetects must be selected as part of an imputation group. Only those tracers that can be considered part of a compositional or subcompositional dataset can be included (Palarea-Albaladejo and Martin-Fernandez, 2014). Size, organic content, stable isotope, and radionuclide data are not considered compositional or subcompositional data and should not be included when imputing values below the reporting limit in the source dataset. If nondetects exist in the portion of the source dataset that contains stable isotope or radionuclide data, it is left up to the user to decide how to manage these nondetects prior to running the program.

The user can choose to impute the entire set of compositional or subcompositional source data or select a smaller group of highly correlated tracers. The user will be given the chance to select groups of tracers for imputation until all nondetects have been imputed. It is recommended that the user try to impute the entire set of compositional or subcompositional data first and then if the desired result is not achieved try smaller groups. The group size can affect the threshold of missing parameters shown in table 1 and table 2. If the user has opted to estimate, substitute, or use any other method to fill nondetects, tracers containing these values should never be used to impute values for other nondetects and should not appear in any imputation group regardless of whether or not the tracer meets the compositional data requirement.

> ***Example Datasets Hint***
>
> All nondetects in the example datasets fall within the compositional data requirement. Selecting all elemental analysis tracers in the Imputation Group is recommended. Do not select radionuclides as they do not meet the compositional data requirement. Click "Select All" in the "Group to Impute" pane on the right, then deselect the radionuclides, 210PbXs and Cs137.

See *appendix 9* for details on the Preparing for Imputation and Imputation Group Selection screen.

Importing Reporting Limits

Reporting limits for all nondetects that are to be imputed must be provided. The user must select what type of reporting limits, either single or multiple, are to be used. "Single reporting limits" means that there is only one reporting limit per tracer, whereas "multiple reporting limits" allow for a different reporting limit for every sample. Single and multiple reporting limits should be formatted according to the example datasets Single Reporting Limits and Multiple Reporting Limits respectively. The type of reporting limit is largely dependent on the data received from the laboratory analysis (Helsel, 2012). Generally, the laboratory results will include either single or multiple reporting limits that comprise either detection or quantitation limits. If access to both multiple and single reporting limits is possible, multiple reporting limits is recommended but not strictly necessary if the analytical laboratory provided only single reporting limits.

> ***Example Datasets Hint***
>
> While both the Single Reporting Limits and Multiple Reporting Limits files will work in Sed_SAT, the use of Multiple Reporting Limits is recommended.

Once the reporting limit dataset is imported, it can be subsetted and used with each imputation group without re-importing; therefore, it is recommended that the user import all reporting limits with the first imputation group. The reporting limits set will also be run through a *Data Testing Module* (see below) to check for proper formatting. See *appendix 10* for details on the Importing Reporting Limits screens.

Imputation Parameters

Single Imputation

The lrEM() function in the zCompositions R-package is used to perform single imputation, the process of estimating a single value for each nondetect. The lrEm() function uses a model-based ordinary expectation-maximization algorithm to impute values for nondetects, while incorporating the relative covariance structure of the data (Palarea-Albaladejo and Martin-Fernandez, 2014).

Multiple Imputation

Multiple imputation is the process by which *m* multiple datasets are created by imputing *m* values for each nondetect and then combining the resulting datasets (Palarea-Albaladejo and Martin-Fernandez, 2014). The value of *m* does not need to be large; a value between 3 and 5 is generally accepted (default m=3); with a higher percent missing, choosing a value for *m* on the upper side of the range is recommended (Little and Rubin, 1987; Lubin and others, 2004). The lrDA() function uses a simulation-based data augmentation algorithm incorporating the relative covariance structure of the data (Palarea-Albaladejo and Martin-Fernandez, 2014).

Both single and multiple imputation give valid estimates for nondetects; therefore, it is left up to the user to decide which method to use. Single Imputation has one advantage in that it is considerably faster to run.

Percent Missing Thresholds

Tracers with a greater percent missing than the user-defined threshold (specified by the user on the Imputations Parameters screen) will be removed from the analysis from this point forward as the nondetects will not be imputed or, in the case of "very small source groups" (<10 samples), the user can choose to remove the source group. Alternatively, the user can manage nondetects in tracers that break the threshold of percent missing through another method, and re-import the source dataset. Special care must be taken, however, if the user applies another method to manage nondetects, as tracers with any type of substituted or estimated values should never be used to impute values for other tracers. See *appendix 11* for details on the Imputation Parameters screen.

View Imputation Results

Sed_SAT displays imputation results as follows:

1. Imputed dataset with imputed values highlighted in yellow and flagged with "*"
2. A dataset of the reporting limits used for each nondetect. Reporting limits will be = 0 for values that were not flagged as nondetects.
3. A list of tracers and (or) source groups that were removed (if any) along with the reason for removing them during the imputation procedure.

At this time the user can choose to (1) adjust the imputation parameters and run imputation again, (2) accept the imputation results and continue to impute other groups, or (3) assuming all nondetects have been imputed, accept the imputation results and move on to the next step in the program. Once imputation results are accepted, the imputed values for those tracers become a permanent part of the source dataset. To impute these values again, the user must re-import the source dataset and start over.

Data Testing Module Loop

After each imputation run, the *Data Testing Module* is repeated to ensure that all nondetects have been imputed. If nondetects still exist because the user selected a group to impute that did not include all tracers, the imputation step will be repeated until all nondetects have been imputed or removed. Once imputation of all nondetects is complete (or at any point in the imputation process), the user can view the final (or current) source dataset by clicking on "DataforR" on the list of table names on the *Program MAP*.

Once all nondetects have been imputed in the source dataset, the user can export the final imputed dataset by clicking Export Data/Tables/Plots at the bottom of the main Program MAP and selecting "DataforR."

> ***Example Datasets Hint***
>
> Once all nondetects have been imputed, the "DataTestFAILED" screen will appear again with only "true negatives" and "true zeros." Click "accept and continue."

Table 1. Single imputation parameters.

Parameter	Single imputation	Default
n	Maximum number of iterations for the expectation-maximization algorithm	50
Percent missing small	For small source groups[1]	20
Percent missing large	For large source groups[2]	60

[1]Number of samples < number of tracers.

[2]Number of samples > number of tracers.

Table 2. Multiple imputation parameters.

Parameter	Description	Default
n	Maximum number of iterations for the data-augmentation algorithm	1,000
m	Number of imputed datasets to be combined	3
Percent missing small	For small source groups[1]	20
Percent missing large	For large source groups[2]	60

[1]Number of samples < number of tracers.

[2]Number of samples > number of tracers.

Target Dataset Data Test

The target dataset is run through the same *Data Testing Module* as the source dataset to identify any missing values, zeros, negatives, or text. The only difference being that the explanation of "nondetect" prompts the user to re-import the target dataset without nondetects as there is no built-in method for managing nondetects in the target dataset. True negatives and true zeros are permitted in the target dataset.

> ***Example Datasets Hint***
>
> The stable isotopes in the target dataset contain "true negatives."

For more details or explanations of potential problems identified refer to the *Data Testing Module* and *appendix 8*.

Negatives and True Zeros

Although "true negatives" and "true zeros" are accepted in the program, in some cases they must be transformed to all positive values prior to running certain steps in the program. To shift data into positive space the program will either (1) apply a constant of (-1) if all of values for a given tracer are negative, or (2), if a given tracer includes both positive and nonpositive values, the user will be prompted to provide a function to shift the data into positive space. The function will be applied only if all other transformations that can be applied to the raw data without shifting it into the set of complex numbers have failed to satisfy the operation being performed (i.e. transforming to achieve a normal distribution or a significant linear regression). User-defined functions must shift the data completely into positive space and must satisfy the inverse function test. If these conditions are not met, the user will be prompted to enter new functions.

> ***Example Datasets Hint***
>
> The example functions given on the Negative Functions input screen (*appendix 13*) can be used on the example datasets.

Start Step 1: Test for Univariate Normal Distributions

Introduction

Step 1 evaluates the univariate distributions of each tracer in each source group to prepare for *Start Step 2: Outlier Test*, where outliers are evaluated via a conservative outlier detection method which requires that tracer values follow a univariate normal distribution (Gellis, Noe, and others, 2015). For more information on the outlier test, refer to Start Step 2: Outlier Test.

Shapiro-Wilk Test

The Shapiro-Wilk W test for normality is used to determine if the raw tracer concentration values are normally distributed within each source group (Shapiro and Wilk, 1965). The user must specify a p-value for rejecting the null hypothesis that the data are normally distributed; the default is Shapiro-Wilk p-value >0.05.

Transformations

If tracer concentration values are normally distributed in their original state, the values are not transformed. However, if tracer concentration values are not normally distributed, the values are transformed via the Tukey Ladder of Powers transformations (see table 3) (Tukey, 1977) If a given tracer's values are all positive, all transformations are applied in a single step and the transformation that yields the maximum Shapiro-Wilk p-value is selected. Refer to section *Negatives and/or Zeros* for details on how transformations are applied to tracers with nonpositive values.

Negatives and/or Zeros

Only tracer concentration values in the set of real numbers are allowed by the program. Transformations that place the tracer concentration values in the set of complex numbers are not permitted. The program defaults to preserving the original data, so in this step if a normal distribution can be achieved by transformations that do not require the data to be shifted via a constant into positive space prior to transformation, those transformations are used (see Group B in table 4 and table 5).

Table 3. Transformations used in Sed_SAT.

Transformations						
x	x^2	$\sqrt[3]{x}$	$\frac{1}{x}$	$\sqrt{x}$	$\frac{1}{\sqrt{x}}$	$\log(x)$

[1]log(x) transformation is a base 10 log transformation.

To transform a tracer containing negatives and (or) zeros using all of the Tukey Ladder of Powers transformations (Tukey, 1977), a constant must be applied. The constant is defined by the user for all tracers with positive and nonpositive values at the end of *Data Testing Module* in *Negatives and True Zeros*. Transformations to achieve a normal distribution of tracer concentration values are applied in groups in the order shown in table 4 and table 5. The transformation with the highest Shapiro-Wilk p-value within each group will be selected. Once a normal distribution has been found within a transformation group, the program stops and does not continue to test the next group. For example, if a normal distribution can be achieved within Group B, Group C transformations will not be applied or evaluated for normality.

Step 1 Best Transforms

The user is presented with a table of transformations for each tracer in each source group along with corresponding Shapiro-Wilk p-values, skewness, and kurtosis. Tracers that cannot be transformed to a normal distribution are highlighted in yellow. Only program defaults (based on rules from table 3, table 4, and table 5) are shown in the initial Step 1 output; however, more output can be obtained by clicking individual tracer names. Clicking the teal tracer name opens a second screen that displays Shapiro-Wilk p-values, skewness, kurtosis, QQ-plots, and histograms for every possible transformation of the selected tracer in the selected source group. The user can then manually change the transformation applied to prepare for *Start Step 2: Outlier Test* by double clicking the plots. Once the user is satisfied with all transformations for all source groups, transformations are applied. See *appendix 14* for details on navigating the Step 1 output screens.

Table 4. Rules for transforming negatives for normality.

Transformation grouping order	Transformation	If ALL values < 0
A	x	No constant applied
B	x^2	No constant applied
B	$\sqrt[3]{x}$	No constant applied
B	$\frac{1}{x}$	No constant applied
C[1]	$\sqrt{x}$	All values multiplied by (-1) and then transformed
C[1]	$\frac{1}{\sqrt{x}}$	All values multiplied by (-1) and then transformed
C[1]	$\log(x)$	All values multiplied by (-1) and then transformed

[1]Indicates a constant will be applied prior to transforming data.

Start Step 2: Outlier Test

Definition of an Outlier

Outliers can strongly bias many statistical tests (Zimmerman, 1994); however, outliers can also contain valuable data (Helsel and Hirsch, 2002), and thus the decision to discard or retain outliers is left up to the user. Automatic removal of outliers without evaluation is not recommended (Helsel and Hirsch, 2002). Step 2 flags outliers in the source dataset so that the user can evaluate the outliers and decide whether to include each sample containing outliers. Outliers are defined by equation 1 below (Gellis, Noe, and others, 2015).

$$x_i > \mu_{X_i} \pm N * \sigma_{X_i} \quad (1)$$

where

x_i = tracer value for tracer (i) in a given source group;
X_i = all tracer values for tracer (i) in a given source group;
N = number, integer or decimal multiple of σ_{X_i};
μ_{X_i} = mean of X_i; and
σ_{X_i} = standard deviation of X_i.

This definition of an outlier assumes that the data are normally distributed (Gellis, Noe, and others, 2015). Outliers found in tracers that could not be transformed to achieve a normal distribution in *Start Step 1: Test for Univariate Normal Distributions* should be evaluated carefully by the user.

Table 5. Rules for transforming data that span both positive and nonpositive space for normality.

Transformation grouping order	Transformation	Some values ≤ 0
A	x	No constant applied
B	x^2	No constant applied
B	$\sqrt[3]{x}$	No constant applied
C[1]	$\frac{1}{x}$	User defined constant/ function applied
C[1]	$\sqrt{x}$	User defined constant/ function applied
C[1]	$\frac{1}{\sqrt{x}}$	User defined constant/ function applied
C[1]	$\log(x)$	User defined constant/ function applied

[1]Indicates a constant will be applied prior to transforming data.

Selecting Parameters

To run the outlier test, select a value for *N* from equation 1 as a multiple of the standard deviation from the mean that will define an outlier. *N*=3 is recommended as a conservative definition of an outlier (Gellis, Noe, and others, 2015). *N* can be either an integer or decimal value. To continue without evaluating outliers, click "Don't Remove Outliers." To select an appropriate N-value, consider the simulated normal distribution below (fig. 11).

Step 2 Output

Samples with values flagged as outliers, tracers containing outliers, and tracer concentration values found to be outliers according to equation 1 are highlighted in yellow in the Step 2 output. Samples with outliers by default do not have the "Include" checkbox checked. To include samples with outliers, the user must check the "Include" checkbox. Only samples marked to be included will be used from this point forward in the program.

To give the user a sense of the degree to which a value falls outside the user-selected *N* standard deviations from the mean, a table of the number of multiples of standard deviations from the mean in transformed space for each outlier is presented below the outlier selection table. Summary statistics in normal (not transformed) space for each tracer with outliers is also presented as a means of evaluating outliers.

> ***Example Datasets Hint***
>
> Outliers flagged in the example datasets can be assumed true outliers due to measurement error and should not be included in subsequent steps.

See *appendix 15* for details on the Step 2 output screen.

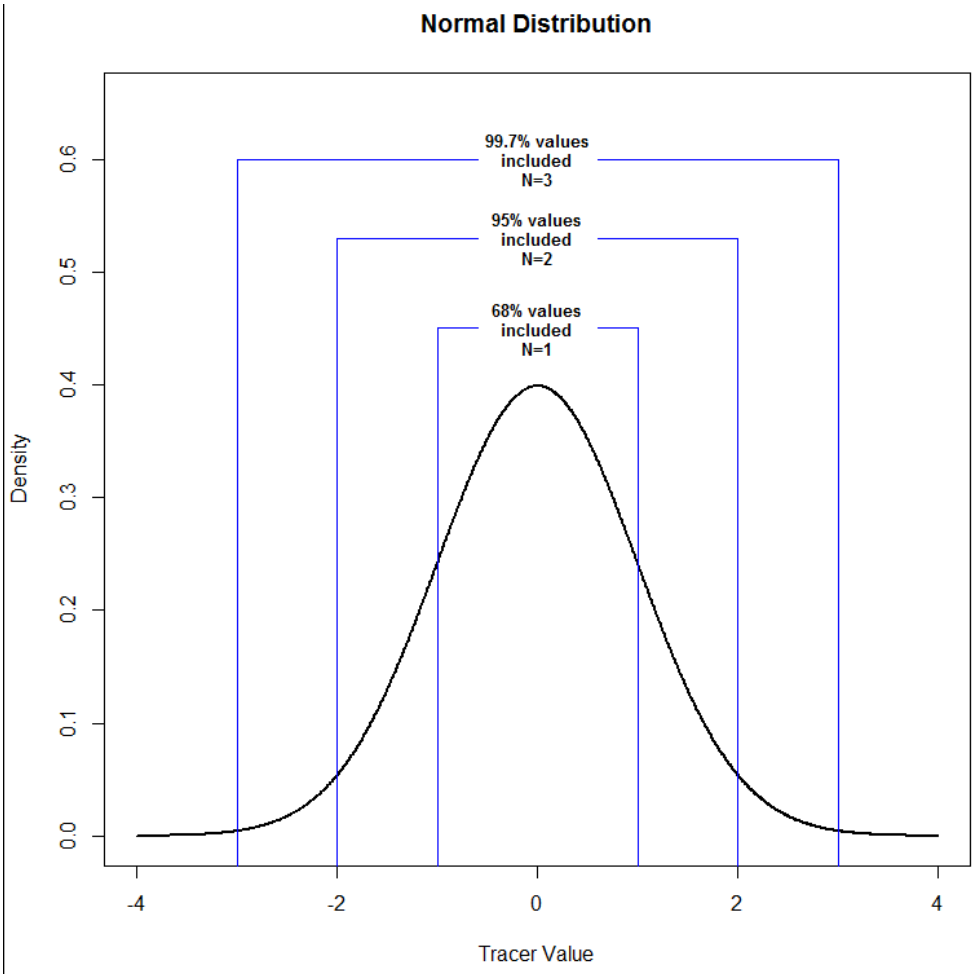


Figure 11. A simulated normal distribution.

Start Step 3: First Linear Regression

Steps 3 and 4 in Sed_SAT are used to adjust the source dataset with respect to size and organic content, thereby making it comparable to the target dataset. The premise behind applying correction factors to source samples for differences in size and organic content in comparison to target samples is outlined in detail in the EPA Manual section 6.4.2 (Gellis, Fitzpatrick, and others, 2016).

Tracer concentration is not solely a function of source material but is influenced by size and organic content (Horowitz and Elrick, 1987; Collins and others, 2010); therefore, before defining a fingerprint signature using source sample tracer concentrations, other factors affecting tracer concentration values must be addressed. Finer grains have greater surface area, which creates more sites to which tracers can attach, resulting in higher tracer concentration values. Similarly, organic matter attached to sediment also provides additional sites for sorption of tracers (orowitz and Elrick, 1987; Collins and others, 2010).

Correction Factors

The linear regression steps (Steps 3 and 4) remove the effect of differences in grain size and organic content between the source dataset and target dataset. This is achieved by testing for a relationship between tracer concentrations and size or organic content data and then applying a correction factor to the tracer concentration values using equation 2 (Gellis, Noe, and others, 2015).

$$C_n = \{Ti_{i(n)} - [(S_j - CF) * m]\}^{\wedge} \quad (2)$$

where

C_n = tracer after size correction (untransformed if transformation was applied);

$Ti_{i(n)}$ = original value of tracer (i) in Source group (n) (transformed if applicable);

S_j = size (or organic content in Step 4) value of sample (j) (transformed if applicable);

CF = mean size (or organic content in Step 4) in target samples (transformed if applicable);

m = slope of regression line; and

$^{\wedge}$ = if transform was applied, the tracer is then untransformed.

After the correction factor is applied, a bias correction factor is applied to adjust for bias resulting from transforming the data and then untransforming the corrected values (Gellis, Noe, and others, 2015). Table 6, bias correction factors, gives the bias corrections applied by transform used on the tracer value ($Ti_{i(n)}$ from eq. 2) (Gellis, Noe, and others, 2015). The corrected untransformed tracer values (C_n from eq. 2) are divided by the bias correction factor to get the final corrected tracer values used in subsequent steps.

Navigating Steps 3 and 4 in the Instruction Manual

If size data exists in the user's dataset (indicated in the *Basic Questions* section), the first linear regression is with size data; otherwise organic content data (TOC, LOI) will be used for the first linear regression. For more details on size and organic content data, see *appendix 3*. For simplicity, *Start Step 3: First Linear Regression* will be said to be a regression with size data from this point forward in the instruction manual; if the dataset being analyzed does not contain size data, assume that "organic content" can be substituted for "size" data in this section of the instruction manual. If neither size nor organic content data are present, no linear regression will take place and the program will proceed to *Start Step 5: Bracket Test*.

How to Skip Size Correction

User may choose to skip Step 3 and not correct the source dataset for size by clicking Skip at the beginning of Step 3. This action will remove the size data from the source dataset from this point forward and alter the user's path through the program. On the *Program MAP*, if organic content data exist, Step 3 will become a regression with organic content and Step 4 will be disabled (see fig. 12 below). If organic content data do not exist, both Steps 3 and 4 will be disabled. To go back and include a size data correction after clicking Skip by mistake, the user must navigate to the Program MAP and click Data Properties in the yellow box next to Step 3 Plots, as shown here in figure 12 and select "Yes" to the *Basic Questions* prompt "Does your dataset include Size Data?" Size data will then be included in the analysis again, and the source dataset can be corrected for size. A similar option to skip will be given in the beginning of *Start Step 4: Second Linear Regression for Organic Content*. Skipping steps when prompted within the normal progression of the program is the only acceptable method for skipping steps. The *Program MAP* should never be used to skip steps as it could contaminate results and (or) cause program failure.

Example Datasets Hint

The example datasets can be run with or without size and (or) organic content corrections, but it is recommended that the user run both corrections to gain familiarity with the output screens. If the example datasets are run without size and (or) organic content corrections, the user should expect the end results to change. Failing to correct for differences in the source dataset due to weathering of target samples results in a very different source dataset going into subsequent steps.

Table 6. Bias correction factors.

[$f(y)$, transformed value of the tracer; x_j, size value (or organic content value for Step 4) for sample (j); $\overline{x}_T$, mean value for size (or organic content value for Step 4) in target data selected for correction; $\hat{b}$, slope of regression line; $\hat{\sigma}\frac{2}{\hat{b}}$, standard error of the regression]

Transformation[1]	Bias Correction Factor
x	1
x^2	$1+\frac{1}{8}[f(y)-(x_j-\overline{x}_T)\hat{b}]^{-2}(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}$
$\sqrt[3]{x}$	$1+[f(y)-(x_j-\overline{x}_T)\hat{b}]^{-2}(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}$
$\frac{1}{x}$	$1+3[f(y)-(x_j-\overline{x}_T)\hat{b}]^{-2}(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}$
$\sqrt{x}$	$1+[f(y)-(x_j-\overline{x}_T)\hat{b}]^{-2}(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}$
$\frac{1}{\sqrt{x}}$	$1+3[f(y)-(x_j-\overline{x}_T)\hat{b}]^{-2}(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}$
$\log(x)$	$10\left[\frac{(x_j-\overline{x}_T)^2\hat{\sigma}\frac{2}{\hat{b}}}{2}\right]$

[1]Applied to $Ti_{i(n)}$ from Eq. (2).

Preparing for Step 3

Selecting Target Samples

Step 3 is a critical junction point in the model, where the user makes decisions that affect how the remainder of the analysis is run. The user can decide either to run Sed_SAT on each individual target sample one at a time or to run the program on the entire target dataset. If the program is run one target sample at a time, the source dataset will be corrected for size and organic content (if applicable) for each target sample being run as each target sample has a different grain size and organic content. If the program is run on the entire target dataset, the source dataset will be corrected for size and organic content to be comparable to the average size and organic content in the target dataset. Although running the model one sample at a time can be slightly more time consuming, the results may have stronger interpretive power (Gellis, Fitzpatrick, and others, 2016). If, however, there are missing values in the size or organic content data, the target dataset samples with missing data cannot be run individually and an average must be used or these target samples dropped from the analysis (Gellis, Noe, and others, 2015).

To begin Step 3, the user must first select the target sample or samples for the current program run. After completing a run on a target sample (through *Start Step 8: Mixing Model and Error Analysis*), go to the *Program MAP*, and click Start Step 3 to run the next target sample using the same source dataset used in the previous run, meaning the previously imputed source dataset with the same outliers removed.

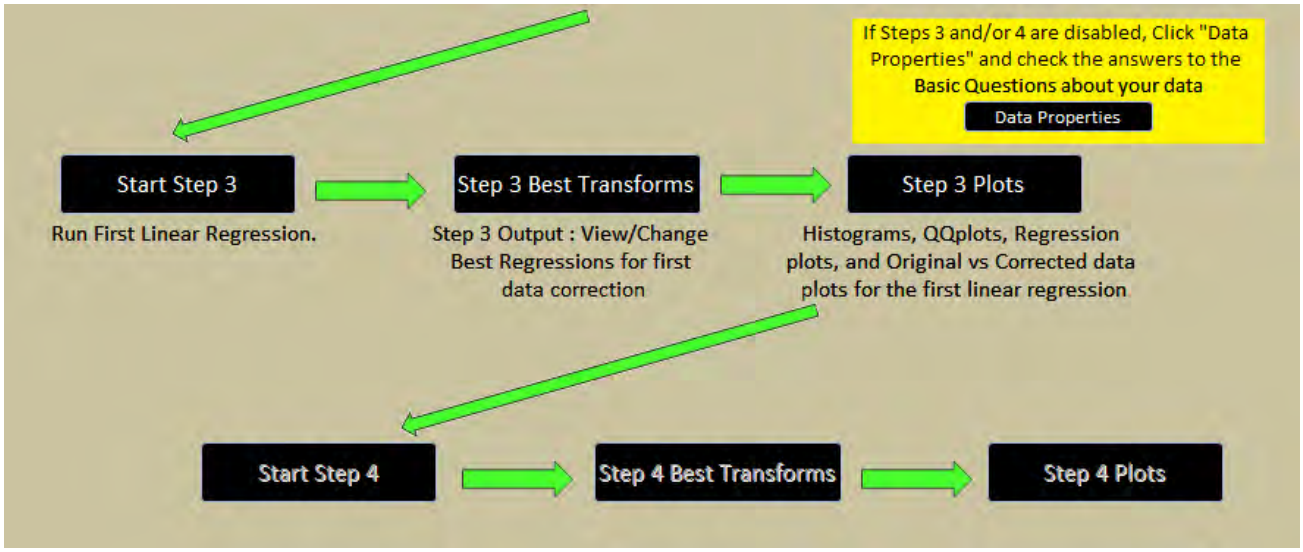


Figure 12. Changes to the Program MAP because of skipping the first linear regression with size data.

Example Datasets Hint

It is recommended that the target samples in the example datasets be run individually from this step forward. If samples are run all at once, the user will see differences in the results due to the corrections being based on all target samples rather than a single target sample's values. For example, sample "target4" has a D50 value of 28.04, but the average D50 value is 17.8; therefore, corrections based on the average would be significantly different from corrections based on the D50 value for "target4," resulting in a very different source dataset being passed into subsequent steps.

See *appendix 16* for details on the target sample selection screen.

Selecting Linear Regression Parameters

After selecting the target sample(s) for analysis, the user must specify parameters defining a "significant regression" in Step 3. A p-value threshold, or alpha value, for the slope of the regression line must be specified by the user. Typically, an alpha-value of 0.05 (default) is sufficient to reject the null hypothesis that there is no relation between the tracer concentration values and size (Helsel and Hirsch, 2002). An alpha value of 0.05 (default) for the Shapiro-Wilk p-value does not reject the null hypothesis that the residuals are normally distributed (Shapiro and Wilk, 1965). A "significant regression" is defined as a regression in which both conditions are met. Regressions that do not meet the user-defined criteria will not be corrected for size as there is no significant relation between the tracer concentration and size.

Linear Regression Procedure

Linear regression is performed for each tracer in each source group using the lm() function in the R-package stats (R Core Team, 2016b). If a significant regression exists according to the user-defined parameters, the regression statistics will be given in an output table. Tracers not found to have a significant relation to size will not be included in the regression output table and will not be corrected for size. Tracer concentrations are the response variables, whereas size is the explanatory variable. Both the explanatory and response variables are transformed via the transformations in table 3. Every possible combination of transformations is tested and the "best" regression is chosen according to the rules below.

1. The regression must be significant.
2. The corrected data (data adjusted for size relation according to eq. 2) must follow the same format as the original data and must not "over-correct" the data. For example, if the original data are all positive for a given tracer, then corrected data cannot be less than or equal to 0.
3. If 1 and 2 are true without transforming either the tracer concentration values or the size data, that regression is selected.
4. If 1 and 2 are true and 3 is not true, then the significant regression with the highest Shapiro-Wilk p-value is selected from the regressions that only transform the size data.
5. If 1 and 2 are true, but 3 and 4 are not true, then the regression with the highest Shapiro-Wilk p-value for the residuals is selected from all significant regressions.
6. If 1 and 2 are not true, no regression is selected.

If tracers with negative and (or) zero values exist in the dataset the rules are modified to choose regressions that do not involve applying a constant to the data in order to transform the data. The regression selection rules, like the *Start Step 1: Test for Univariate Normal Distributions* rules, always keeping the data in its original form if possible. See table 4 and table 5 for the order in which transformations are applied to tracers with negative and (or) zero values.

Step 3 Output

The initial Step 3 output screen is similar to the output for *Start Step 1: Test for Univariate Normal Distributions*. For all tracers found to have a *significant* regression with size, a table of regression statistics for the *selected* regressions is given for each source group. Regressions are chosen on the basis of a hierarchy that defaults toward utilizing raw rather than transformed data (see *Linear Regression Procedure*). However, the user has the option of choosing significant regressions other than the default. To view/change the regression used to correct tracer concentration values for a given tracer in a given source group, click the tracer name highlighted in teal. The user can then view regression plots, QQ-plots, histograms, corrected data plots, and statistics for all significant regressions found for the selected tracer in the selected source group and apply a preferred regression or continue with the default. The user can also choose to not correct a given tracer in a given source group by clicking the "No Relation" checkbox in the initial output table.

Once satisfied with all the regression results or to view plots for all selected significant regressions, click Click to Accept ALL Transforms for ALL Tracers for ALL Sources. The user will then be presented with all statistics, regression plots, QQ-plots, histograms, and corrected data plots for all tracers in all source groups found to have significant regressions according to user selections in the initial Step 3 output. It is strongly recommended that the user review all plots and statistics to ensure that the data are properly corrected for size. At this point the user can go back and select alternate regressions or indicate that there is "No Relation" for any given tracer in any given source group.

Once all regressions have been approved, the data are corrected for size according to equation 2 prior to running the next step in the program. This corrected source dataset will be the source data used from this point forward in the program. At the beginning of the next step in the program (Start *Step 4: Second Linear Regression for Organic Content* if organic content data are present or *Start Step 5: Bracket Test* if organic content data are not present), the user will be given the option to skip all corrections for size by clicking Skip Apply GRAINsize corrections. See *How to Skip Size Correction* for details on how skipping the step will alter the program path and how reverse the selection to skip the step.

See *appendix 17* for details on navigating the Step 3 outputs and selecting nondefault regressions.

Start Step 4: Second Linear Regression for Organic Content

Step 4 is an option only if the dataset contains both size and organic content data. If correcting for both, the size-corrected source dataset will be used in the second linear regression. The second linear regression procedure and output is identical to that in Step 3. For details on the regression procedure, refer to *Start Step 3: First Linear Regression*, and for output navigation see *appendix 17*. Refer to *How to Skip Size Correction* for instructions on how to skip Step 4.

Start Step 5: Bracket Test

A prerequisite of sediment fingerprinting is that the tracer concentration values in the target dataset must be conservative and not change during transport from the source to the sampling point (Gellis and Walling, 2011; Gellis, Noe, and others, 2015). Consequently, the next step in the analysis is determining that for any given tracer, the target samples are within the range or "bracket" of the tracer concentration values in the source dataset. Any tracers that fail to satisfy this constraint within measurement error (parameter "p" from eq. 3 input by the user) are considered nonconservative and are removed from all subsequent analyses. Conservative tracers conform to the following condition (Gellis, Noe, and others, 2015).

$$\min(Y_i) - p * \min(Y_i) < x_i < \max(Y_i) + p * \max(Y_i) \qquad (3)$$

where

x_i = target tracer concentration value for tracer (i);

Y_i = vector of all source tracer concentrations for tracer (i); and

p = increase in range of the source dataset defining the "bracket," input by the user.

Selecting Bracket Test Parameters

Before running the bracket test the user must choose a percent increase of the range of the source dataset (p from eq. 3 above) that will define the "bracket" within which the target samples should fall. The default p is set at 0.1 (10 percent) which is generally within the analytical error of laboratory analysis (Gellis, Noe, and others, 2015).

Step 5 Output

If all tracer concentrations for the target sample(s) pass the bracket test, the program will progress to *Start Step 6: Multivariate Normality Test* with all tracers. Any tracers that fail the bracket test are flagged as nonconservative (highlighted in yellow) and will be removed from the analysis from this point forward unless the user overrides the default. Target samples that result in a tracer(s) that fail the bracket test should be evaluated to discern if there is a legitimate reason for them to be outside the bracket. To include a tracer that fails the bracket test, check the "Include" checkbox in the list of "Discarded Tracers" on the left-hand side of the screen.

> ***Example Datasets Hint***
>
> Using the default value for p (0.1), only one target sample will have tracers that fail the bracket test.

For more details on the Step 5 output see *appendix 19*.

Start Step 6: Multivariate Normality Test

Step 6 evaluates assumptions behind the next step in the program, *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis* (DFA). DFA works under the assumptions that the data being analyzed follow a multivariate normal distribution and homoscedasticity or equal variance (Fisher, 1936). Step 6 tests whether or not these assumptions are met by the source dataset.

Power Transformations

The portion of the source dataset that passed the bracket test and was corrected for size and (or) organic content (if applicable) is tested without transformations and with Box-Cox transformations. The formula for the Box-Cox power transformation is shown in equation 4 (Box and Cox, 1964):

$$x'_\lambda = \frac{x^\lambda - 1}{\lambda}, \text{ if } \lambda \neq 0 \quad (4)$$

$$= \ln(x), \text{ if } \lambda = 0$$

where

x'_λ = transformed value;
x = original value; and
λ = power transform, value between -1 and 1.

Lambda values for the Box-Cox transform are generated using the powerTransform() function with family = "bcpower" in the car R-package (Fox and Weisberg, 2011). The Box-Cox transformations are applied using the boxCox() function in the USGS R-package smwrBase (Lorenz, 2015b), modified to preserve the original dimensional units (Draper and Smith, 1998).

In some datasets, the scale of the data may trigger an error when the powerTransform() and family = "bcpower." In this case transforms are chosen using family = "yjpower" on all positive data. This is the Yeo-Johnson power transform, which when applied to all positive data is the same as applying the Box-Cox transform after adding 1 to all values (Yeo and Johnson, 2000). Data are moved into positive space prior to transforming so that the boxCox() function can be used to apply transforms; therefore, only the first condition of the Yeo-Johnson transform is applied.

$$x'_\lambda = \frac{(x+1)^{\lambda_+} - 1}{\lambda_+}, \text{ if } \lambda_+ \neq 0, x \geq 0 \quad (5)$$

$$= \log(y+1), \text{ if } \lambda_+ = 0, x \geq 0$$

$$= -\frac{\left[(-y+1)^{2-\lambda_-} - 1\right]}{2-\lambda_-}, \text{ if } \lambda_- \neq 0, y < 0$$

$$= -\log(-y+1), \text{ if } \lambda_- = 0, y < 0$$

where

x'_λ = transformed value;
x = original value; and
λ = power transform, value between -1 and 1.

Multivariate Normality Tests

The five multivariate normality tests applied to the source dataset and the boxCox()-transformed source dataset are listed in table 7 below.

Homoscedasticity Test

The betadisper() function from the vegan R-package (Oksanen and others, 2016) applies a multivariate version of the Levene test for homogeneity of variances (Levene, 1960). Boxplots and ANOVA results are given to inform the user's choice of whether or not to transform the data prior to *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis*. Ideally, the p-value from the ANOVA table should be less than 0.05, and boxplots should show equal variance in all source groups.

Step 6 Output

A pass-fail value of 1 is assigned for each test with a resulting p-value of <0.05, and a value of 0 is assigned for each p-value of >0.05. The "MVN score" is the sum of the pass-fail values for all the tests. By default, the model will

Table 7. Multivariate normality tests applied with corresponding R-packages and functions.

Test	R-package	Function
Multivariate Shapiro-Wilk Test	mvnormtest (Jarek, 2012)	mshapiro.test()
Mardia Test for Skewness	MVN (Korkmaz and others, 2014)	mardiaTest()p.value.skew
Mardia Test for Kurtosis	MVN (Korkmaz and others, 2014)	mardiaTest()p.value.kurt
Royston Test	MVN (Korkmaz and others, 2014)	roystonTest()
Henze-Zirkler Test	MVN (Korkmaz and others, 2014)	hzTest()

choose the version of the source dataset (raw or transformed) that has the highest "MVN score," but the user can override the default and choose the alternate case by clicking the plot. If neither the source dataset nor the boxCox()-transformed source dataset passes any of the multivariate normality tests in table 7 (MVN score = 0), the user should note that the assumption behind *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis* has been violated, but can still proceed and complete the program run, as the DFA procedure is sufficiently robust that results can still be utilized if the assumption is not met. In the case in which the MVN score is 0 for both the raw and transformed source datasets, it is recommended that the source dataset not be transformed, since transforming the source dataset does not result in a multivariate normal distribution.

> ***Example Datasets Hint***
>
> The example datasets will not pass multivariate normality tests, which is typical for the type of data analyzed for sediment fingerprinting.

See *appendix 20* for details on navigating Step 6 output.

Start Step 7: Forward Stepwise Linear Discriminant Function Analysis

Forward Stepwise Linear Discriminant Function Analysis

Step 7 determines the best group of tracers on which to run *Start Step 8: Mixing Model and Error Analysis* through Forward Stepwise Linear Discriminant Function Analysis (DFA) (Collins and others, 2010; Gellis, Noe, and others, 2015). DFA looks for the linear combination of tracer coefficients that best separates or discriminates the source groups (Fisher, 1936). DFA is performed utilizing the greedy.wilks() function in the R-package klaR (Weihs and others, 2005). The greedy.wilks() function starts tracer selection with the tracer that yields the greatest separation between the groups, and adds tracers using the Wilk's lambda criterion (Mardia and others, 1980). The closer the Wilk's lambda statistic is to 0, the more significant a tracer's contribution to the linear discriminant function.

Parameter Selection

The user must specify a significance level or threshold for adding a tracer to the linear discriminant function in Forward Stepwise Linear Discriminant Function Analysis (DFA); this is the "niveau" argument in the greedy.wilks() function. The default significance level is 0.01.

If the default significance level results in fewer tracers selected for the linear discriminant function than there are source groups, the level must be increased as *Start Step 8: Mixing Model and Error Analysis* requires that there be more tracers (at least 1 more) than source groups for the mixing model to work properly (Mukundan and others, 2010). If fewer tracers are selected than the number of sources, the user will see a warning screen before starting Start Step 8: Mixing Model and Error Analysis.

The user must also set the tolerance ("tol") argument for the lda() function in the R-package MASS (Venables and Ripley, 2002) used in generating the *Weighting Factors*, *Confusion Matrix*, and *Biplot*. The default value for tol is 0.0001, which is the default value in the R-package MASS. If any tracer has a within source group variance less than the tol^2, the lda() function will fail, and the user will see an error message saying that some variables are constant. This can occur as a result of scaling. When setting the tol parameter, the user should consider the scale of the tracers going into DFA. If an error occurs, and the scale of the tracers is less than the tolerance, decrease the tol parameter and rerun Step 7.

Biplot

The biplot is generated using the ggord() function from the ggord R-package (Beck, 2016), using the tracers selected by the greedy.wilks() function. The biplot shows predicted values

from the first two linear discriminants. Keep in mind that if more than three sources are being analyzed, the first two linear discriminants do not show the full discriminatory power of the group of tracers selected. Refer to the *Confusion Matrix* to view how well the complete linear discriminant function classifies source groups.

Cross Validation

The leave-one-out, cross validation procedure (Lachenbruch and Mickey, 1968) tests how well the linear discriminant function classifies each source group given equal prior probabilities of each source sample belonging to any given source group using the lda() and predict() functions in the R-package MASS (Venables and Ripley, 2002). In other words, the procedure tests how well the group of tracers selected through the DFA procedure correctly identifies each source sample as belonging to the known correct source group. The "Cummulative_Percent_Classified" and "Individual_Percent_Classified" fields in the DFA results table, the *Confusion Matrix*, and the *Weighting Factors* table are all generated using leave-one-out cross validation.

Confusion Matrix

The confusion matrix shows the number of source samples predicted for each group versus the actual number of source samples in each group (Provost and Kohavi, 1998) (Kohavi and Provost, 1998). If 100 percent of the source samples are correctly identified, the confusion matrix will contain numbers only on the diagonal. While the confusion matrix gives some indication of how well the linear discriminant function works, that does not directly correlate to the number of source samples that will be correctly classified in the *Start Step 8: Mixing Model and Error Analysis* as the mechanisms for classifying the samples are different.

Weighting Factors

Weighting Factors used in *Start Step 8: Mixing Model and Error Analysis* are generated using *Cross Validation* in Step 7. The tracer with the lowest percent of source samples classified correctly will have a weighting factor equal to 1. Weighting factors are calculated using equation 6 (Collins and others, 2010; Gellis, Noe, and others, 2015).

$$W_i = \frac{P_i}{P_{opt}} \tag{6}$$

where

P_i = percent of source samples classified correctly using tracer (i) and

P_{opt} = percent of source samples classified correctly using tracer with the lowest P_i.

Details on navigating Step 7 output screens are given in *appendix 21*.

Start Step 8: Mixing Model and Error Analysis

Preparation and Parameters

To start Step 8, the user is first asked to determine which target samples to analyze in the mixing model. Only the samples selected in *Start Step 3: First Linear Regression* will appear as options. The only benefit to selecting a single sample at this time is if the user is working under a time constraint and cannot analyze all the samples at that time. The user is then prompted to input the number of iterations in the *Monte-Carlo Simulation* (N, default 1,000) as part of the *Error Analysis*. To run the mixing model without a Monte-Carlo simulation set N=0. Since every iteration of the Monte-Carlo Simulation uses means based on a source dataset with one randomly selected sample removed from each source group, the results from N=1 in the Monte-Carlo Simulation are not the same as the mixing model results based on the entire source dataset.

At this point the user must indicate whether the Weighting Factors generated in *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis* are to be applied to the mixing model. If the user indicates that uniform weighting is to be applied (equal weighting for all tracers selected), a second option will appear that allows the user to either apply uniform weighting to the tracers found to be significant in Start Step 7: Forward Stepwise Linear Discriminant Function Analysis or to apply uniform weighting to all tracers. If uniform weighting is not selected, using all tracers is not an option because only those weighting factors for tracers found to be significant in Step 7 are included. See *Weighting Factors* for details on how weighting factors are applied.

Mixing Model

The final step in sediment fingerprinting is the determination of sediment source contributions for Target samples using a Mixing Model (Collins and others, 2010; Gellis, Noe, and others, 2015). The remaining set of tracers selected in *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis* is used in the mixing model with the size and organic content correction factors applied (if applicable). The mixing model does not use data transformed for multivariate normality in *Start Step 6: Multivariate Normality Test*, as MVN is not an assumption behind the mixing model procedure.

Equation 7 below gives the error associated with the mixing model result (Collins and others, 2010; Gellis, Noe, and others, 2015).

This error term is optimized using the optim() function in base R (2015b; R Core Team, 2016b) to get the minimum error. The mixing model can be run with or without the *Weighting Factors* from *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis*, if the weighting factors are not applied $W_i = 1$ in Eq. (7).

$$\sum_{i=1}^{n}\{[C_i-(\sum_{i=1}^{m}P_sS_{si})]/C_i\}^2W_i \quad (7)$$

$$\text{with } \sum_{s=1}^{n}P_s=1$$

where

C_i = concentration of tracer (i) in the target sample;
P_s = optimized percentage of contribution of source type (s);
S_{si} = mean concentration of tracer (i) in source (s) (after size and/or organic content corrections if applicable);
W_i = weighting factor for tracer (i);
n = number of tracers comprising the optimum composite fingerprint (tracers selected in *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis*); and
m = number of sediment source types.

The mixing model output is a table of percent source contributions for each target sample along with the error term from equation 7 associated with the result. Graphical displays of the results are given as pie charts and stacked bar plots that can be accessed by clicking View Summary Plots on the initial Step 8 output screen.

Error Analysis

The program performs a variety of different analyses to evaluate error, including (1) Monte-Carlo simulation, (2) source dataset analysis, and (3) graphical display in tracer-by-tracer plots.

Monte-Carlo Simulation

A Monte-Carlo simulation approach (Metropolis and Ulam, 1949) is used to quantify the uncertainty in the sediment fingerprinting results produced in the *Mixing Model* (Collins and others, 2010; Gellis, Noe, and others, 2015). The Monte-Carlo simulation is run by randomly selecting a single source sample from each source group N times (parameter selected by user, default 1,000) and running the mixing model utilizing the remaining source dataset (Gellis, Noe, and others, 2015). Ideally, the mixing model results from the Monte-Carlo simulation should be the same or similar to the primary results of the Mixing Model run on the entire source dataset. If significant variation is present, further investigation into the source data may be needed. Boxplots and summary statistics from the Monte-Carlo simulation are output as well as a complete table of Monte-Carlo simulation results. Running a Monte-Carlo simulation is optional but recommended; to run the Mixing Model without a Monte-Carlo simulation, simply set N=0 when starting Step 8.

Source Dataset Analysis

In order to evaluate the effectiveness of the current Sed_SAT run in distinguishing sources, source samples are run through the mixing model as though they are target samples. Ideally each source sample would be classified as having 100 percent the known source type for that sample. In other words, if a bank sample is run through the mixing model it should classify as 100 percent bank; however, some variation within the source dataset is expected. For example, variation could be due to source types deposited on another source type, such as might occur when the tops of the banks have some agricultural sediment on the surface. A table of percent source contributions for each source sample is output as well as a summary table of the results by source type and pie charts and stacked bar plots for each source type. If source types do not consistently show a majority of that source type, the user may question whether the sources should be combined. For example, if cropland and pasture show mixed results, perhaps they should be combined.

Tracer-by-Tracer Plots

Lastly, the user can run an additional script to obtain tracer-by-tracer plots by clicking View Tracer-by-Tracer Plots on the initial Step 8 output screen. The tracer-by-tracer plots show the mean source sample tracer concentrations by source type. The expectation is that the target samples will plot in the region expected based on the *Mixing Model* results. For example, if the mixing model determined that the sediment sources for a given target sample were 80 percent BANK and 20 percent CROP, the target sample should plot somewhere between the mean concentration for BANK samples and CROP samples, but closer to BANK than CROP. This test tends to be more effective when applied to tracers with high *Weighting Factors*. This portion of the *Error Analysis* is not automatically run as part of the program; the user must click View Tracer-by-Tracer Plots to run this portion of the error analysis.

Details on navigating Step 8 output screens are given in *appendix 22*.

Export Data/Tables/Plots

At the end of each run, the user can document the current Sed_SAT run by exporting all data, output tables, plots, and parameters. The export screen can be accessed either from initial Step 8 output or from the *Program MAP*. All exports will be organized in folders by program step within the output path selected by the user. The user must select which data, tables, plots, and parameters to export. Data, tables, and parameters are output first; then the user will have the opportunity to export plots. All plots are saved temporarily during each run of the program within the folders in the SedimentFingerprinting_R folder; however, it is recommended that the user export plots using the export screen rather than copying plots from the temp folders as accidentally deleted or moved temp folders will cause program failure. See *appendix 23* for details on navigating the export screens.

Repeating Analysis for the Next Target Sample

If the user opted to run a single target sample through the program at the beginning of *Start Step 3: First Linear Regression* (or *Start Step 5: Bracket Test* if no size or organic content data are present), the user can run the next target sample by returning to the *Program MAP* and clicking Start Step 3 or Start Step 5 respectively, after exporting all necessary data, tables, parameters, and plots (these will be overwritten when the next sample is run). When selecting the next target sample for analysis, ensure that all currently selected samples are cleared. The next sample will then be run using the same sample source dataset as for the previously run sample, meaning the imputed source dataset (if applicable) with any selected outliers removed but not corrected for size or organic content. All tracers are present in the source dataset at this point, not only the set used in the most recent *Mixing Model* run as that set was only relevant to the previously run sample. See *Selecting Target Samples* for more details on how and why to run target samples individually, starting at Step 3 (or Step 5 if applicable.)

References

Adler, Daniel, Murdoch, Duncan, and others, 2016, rgl: 3D visualization using OpenGL (R package, ver. 0.95.1441).

Aitchison, John, 1986, The statistical analysis of compositional data.: London, U.K., Chapman and Hall, Ltd., 416 p.

Aitchison, John, 1997, The one-hour course in compositional data analysis or compositional data—The approximation of one matrix by another of lower rank, *in* Pawlowsky-Glahn, V., ed., Proceedings of IAMG'97—The Third Annual Conference of the International Association for Mathematical Geology, International Center for Numberical Methods in Engineering (CIMNE), p. 3–35.

Alfons, Andreas, 2012, cvTools: Cross-validation tools for regression models (R package, ver. 0.3.2 ed.).

Allaire, J.J., Horner, Jeffery, Marti, Vincent, and Porte, Natacha, 2015, markdown: 'Markdown' Rendering for R (R package, ver. 0.7.7).

Azzalini, Adelchi, and Genz, Alan, 2016, The R package 'mnormt': The multivariate normal and 't' distributions (ver. 1.5-4).

Bache, S.M., and Wickham, Hadley, 2014, magrittr: A Forward-Pipe Operator for R (R package, ver. 1.5).

Bates, Douglas, and Maechler, Martin, 2015, MatrixModels: Modelling with sparse and dense matrices (R package, ver. 0.4-1).

Bates, Douglas, and Maechler, Martin, 2016, Matrix: Sparse and dense matrix classes and methods (R package, ver. 1.2-6).

Bates, Douglas, Maechler, Martin, and Bolker, Ben, 2014a, MEMSS: Data sets from mixed-effects models in S (R package, ver. 0.9-2).

Bates, Douglas, Maechler, Martin, and Bolker, Ben, 2014b, mlmRev: Examples from multilevel modelling software review (R package, ver. 1.0-6).

Bates, Douglas, Maechler, Martin, Bolker, Ben, and Walker, Steve, 2015, Fitting Linear Mixed-Effects Models Using {lme4}: Journal of Statistical Software, v. 67, no. 1, p. 1–48, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v067.i01.

Bates, Douglas, Mullen, K.M., Nash, J.C., and Varadhan, Ravi, 2014, minqa: Derivative-free optimization algorithms by quadratic approximation (R package, ver. 1.2.4).

Beck, M.W., 2016, ggord: Ordination plots with ggplot2 (R package, ver. 0.11.9000).

Bernaards, C.A., and Jennrich, R.I., 2005, Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis: Educational and Psychological Measurement, v. 65, no. 5, p. 676–696, accessed July 17, 2017, at https://doi.org/10.1177/0013164404272507.

Box, G.E.P., and Cox, D.R., 1964, An analysis of transformations: Journal of the Royal Statistical Society, Series B (Methodological), v. 26, no. 2, p. 211–252.

Braun, W.J., 2015, MPV: Data sets from Montgomery, Peck and Vining's Book (R package, ver. 1.38).

Bravo, H.C., 2016, Rcsdp: R Interface to the CSDP semidefinite programming library (R package, ver. 0.1.55).

Canty, Angelo, and Ripley, Brian, 2016, boot: Bootstrap R (S-Plus) functions (R package, ver. 1.3-18).

Chang, Winston, 2016, R6: Classes with reference semantics (R package, ver. 2.1.2).

Chasalow, Scott, 2012, combinat: combinatorics utilities (R package, ver. 0.0-8).

Collins, A.L., Walling, D.E., Webb, Louise, and King, Paul, 2010, Apportioning catchment scale sediment sources using a modified composite fingerprinting technique incorporating property weightings and prior information: Geoderma, v. 155, nos. 3–4, p. 249–261, accessed July 17, 2017, at https://doi.org/10.1016/j.geoderma.2009.12.008.

Conceicao, E.L.T., and Maechler, M., 2015, DEoptimR: Differential evolution optimization in pure R (R package, ver. 1.0-4).

Conway, Joe, Eddelbuettel, Dirk, Nishiyama, Tomoaki, Prayaga, S.K., and Tiffin, Neil, 2016, RPostgreSQL: R interface to the PostgreSQL database system (R package, ver. 0.4-1).

Csardi, Gabor, 2015, crayon: Colored terminal output (R package, ver. 1.3.1).

Csardi, Gabor, and Sorhus, Sindre, 2015, praise: Praise users (R package, ver. 1.0.0).

Dahl, D.B., 2016, xtable: Export tables to LaTeX or HTML (R package, ver. 1.8-2).

Davison, A.C., and Hinkley, D.V., 1997, Bootstrap methods and their applications: Cambridge, U.K., Cambridge University Press, 596 p.

Dowle, Matt, Srinivasan, Arun, Short, Tom, and Lianoglou, Steve, 2015, data.table: Extension of Data.frame (R package, ver. 1.9.6).

Draper, N.R., and Smith, H., 1998, Applied regression analysis: New York, John Wiley & Sons, Inc., 736 p.

Dubois, Anne, Bertand, Julie, Mentre, France, and Bates, Douglas, 2012, PKPDmodels: Pharmacokinetic/pharmacodynamic models (R package, ver. 0.3.2).

Eddelbuettel, Dirk, 2013, Seamless R and C++ integration with Rcpp (1st ed.): New York, Springer, 220 p.

Eddelbuettel, Dirk, Emerson, J.W., and Kane, M.J., 2016, BH: Boost C++ Header Files (R package, ver. 1.60.0-2 ed.).

Eddelbuettel, D., and Francois, R., 2011, Rcpp: Seamless R and C++ Integration: Journal of Statistical Software, v. 40, no. 8, p. 1–18, R package version 1.60.0-2, accessed July 17, 2017, at https://www.jstatsoft.org/article/view/v040i08/v40i08.pdf.

Eddelbuettel, Dirk, Lucas, Antoine, Tuszynski, Jarek, Bengtsson, Henrik, Urbanek, Simon, Frasca, Mario, Lewis, Bryan, Stokely, Murray, Muehleisen, Han, Murdock, Duncan, Hester, Jim, Wu, Wush, and Onkelinx, Thierry, 2016, digest: Create compact hash digests of R objects (R package, ver. 0.6.9).

Everiit, B.S., and Hothorn, Torsten, 2015, HSAUR2: A handbook of statistical analyses using R (R package, ver. 1.1-14).

Filzmoser, Peter, Fritz, Heinrich, and Kalcher, Klaudius, 2014, pcaPP: Robust PCA by projection pursuit (R package, ver. 1.9-60).

Filzmoser, Peter, and Gschwandtner, Moritz, 2015, mvoutlier: Multivariate outlier detection based on robust methods (R package, ver. 2.0.6).

Fisher, R.A., 1936, The use of multiple measurements in taxonomic problems: Annals of Eugenics, v. 7, no. 2, p. 179–188, doi: 10.1111/j.1469-1809.1936.tb02137.x, accessed July 17, 2017, at http://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x/abstract.

Fortran code by Genz, A., and R code by Kenkel, B., 2015, pbivnorm: Vectorized Bivariate Normal CDF (R package, ver. 0.6.0).

Fox, John, 2003, Effect Displays in R for Generalised Linear Models: Journal of Statistical Software, v. 8, no. 15, p. 1–27, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v008.i15.

Fox, John, Nie, Z.N., and Byrnes, Jarrett, 2016, sem: Structural equation models (R package, ver. 3.1-7).

Fox, John, and Weisberg, Sanford, 2011, An R companion to applied regression (2d ed.): Thousand Oaks, California, SAGE Publications, Inc., 472 p.

Friendly, Michael, 2015, Lahman: Sean Lahman's Baseball Database (R package, ver. 4.0-1).

Gagolewski, Marek, and Tartanus, Bartek, 2015, R package stringi: Character string processing facilities (R package, ver. 1.1.1).

Gellis, A.C., Fitzpatrick, Faith, and Schubauer-Berigan, J., 2016, A manual to identify sources of fluvial sediment: U.S. Environmental Protection Agency, EPA/600/R-16/210, 106 p., accessed July 17, 2017, at https://nepis.epa.gov/Exe/ZyPDF.cgi/P100QVM1.PDF?Dockey=P100QVM1.PDF

Gellis, A.C., Noe, G.B., Clune, J.W., Myers, M.K., Hupp, C.R., Schenk, E.R., and Schwarz, G.E., 2016, Sources of fine-grained sediment in the Linganore Creek watershed, Frederick and Carroll Counties, Maryland, 2008–10: U.S. Geological Survey Scientific Investigations Report 2014–5147, 56 p., accessed July 17, 2017, at http://dx.doi.org/10.3133/sir20145147.

Gellis, A.C., and Walling, D.E., 2011, Sediment-source fingerprinting (tracing) and sediment budgets as tools in targeting river and watershed restoration programs, *in* Simon, A., Bennett, S.J., and Castro, J.M., eds., Stream restoration in dynamic fluvial systems: American Geophysical Union Monograph Series 194, p. 263–291., DOI: 10.1029/2010GM000960, accessed July 17, 2017, at http://onlinelibrary.wiley.com/doi/10.1029/2010GM000960/summary.

Gelman, Andrew, and Su, Yu-Sung, 2015, arm: Data analysis using regression and multilevel/hierarchical models (R package, ver. 1.8-6).

Genz, Alan, and Bretz, Frank, 2009, Computation of multivariate normal and t probabilities: Heidelberg, Springer-Verlag, v. 195, 126 p.

Genz, Alan, Bretz, Frank, Miwa, Tetsuhisa, Mi, Xuefei, Leisch, Friedrich, Scheipl, Fabian, and Hothorn, Torsten, 2016, mvtnorm: Multivariate normal and t distributions (R package, ver. 1.0-5 ed.).

Gilbert, Paul, 2014, setRNG: Set (normal) random number generator and seed (R package, ver. 2013.9-1).

Gilbert, Paul, and Varadhan, Ravi, 2015, numDeriv: Accurate numerical derivatives (R package, ver. 2014.2-1).

Gorman Sanisaca, L., Gellis, Allen, Lorenz, David, 2017, Sediment Source Assessment Tool (Sed_SAT): U.S. Geological Survey software release, accessed at https://my.usgs.gov/bitbucket/projects/SED/repos/sed_sat/.

Goulet, Vincent, Dutang, Christophe, Maechler, Martin, Firth, David, Shapira, Marina, Stadelmann, Michael, and expm-developers@lists.R-forge.R-project.org, 2015, expm: Matrix Exponential (R package, ver. 0.999-0).

Grolemund, Garrett, and Wickham, Hadley, 2011, Dates and times made easy with lubridate: Journal of Statistical Software, v. 40, no. 3, p. 1–25. accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v040.i03.

Grosjean, Phillipe, 2016, SciViews-R: A GUI API for R (R package, ver. 0.7-12).

Gross, Juergen, and Ligges, Uwe, 2015, nortest: Tests for normality (R package, ver. 1.0-4).

Grothendieck, Gabor, 2014a, gsubfn: Utilities for strings and function arguments (R package, ver. 0.6-6).

Grothendieck, Gabor, 2014b, sqldf: Perform SQL selects on R data frames (R package, ver. 0.4-10 ed.).

Grothendieck, Gabor, and Mueller, Thomas, 2014, RH2: DBI/RJDBC interface to h2 database (R package, ver. 0.2.3).

Halekoh, Ulrich, and Højsgaard , Søren, 2014, A Kenward-Roger approximation and parametric bootstrap methods for tests in linear mixed models - The R package pbkrtest: Journal of Statistical Software, v. 59, no. 9, p. 1–30, accessed July 17, 2017 at http://dx.doi.org/10.18637/jss.v059.i09.

Helsel, D.R., 2012, Statistics for censored environmental data using Minitab and R (2d ed.): New York, John Wiley & Sons, Inc., 324 p.

Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water Resources Investigations, book 4, chap. A3, 523 p., accessed July 17, 2017, at https://pubs.usgs.gov/twri/twri4a3/.

Hester, Jim, 2016, covr: Test coverage for packages (R package, ver. 2.0.1).

Hester, Jim, Müller, K.M., Wickham, Hadley, and Chang, Winston, 2016, withr: Run code 'With' temporarily modified global state (R package, ver. 1.0.2).

Hirsch, R.M., and De Cicco, L.A., 2015, User guide to Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval: R packages for hydrologic data U.S. Geological Survey Techniques and Methods (ver. 2.0), book 4, chap. A10, 93 p., accessed July 17, 2017, at https://pubs.usgs.gov/tm/04/a10/.

Højsgaard, Søren, and Halekoh, Ulrich, 2016, doBy: Groupwise statistics, LSmeans, linear contrasts, utilities (R package, ver. 4.5-15).

Hornik, K., 2016, R FAQ: Frequently Asked Questions on R, accessed August 9, 2016, at https://cran.r-project.org/doc/FAQ/R-FAQ.html.

Horowitz, A.J., and Elrick, K.A., 1987, The relation of stream sediment surface area, grain size, and composition to trace element chemistry: Applied Geochemistry, v. 2, no. 4, p. 437–451, doi: 10.1016/0883-2927(87)90027-8.

Ihaka, Ross, Murrel, Paul, Hornik, Kurt, Fisher, J.C., and Zeileis, Achim, 2009, colorspace: Color space manipulation (R package, ver. 1.2-6).

James, D., and Hornick, K., 2015, chron: Chronological Objects which Can Handle Dates and Times (R package, ver. 2.3-47).

Jarek, Slawomir, 2012, mvnormtest: Normality test for multivariate variables (R package, ver. 0.1-9).

Johnson, S.G., 2014, The NLopt nonlinear-optimization package (R package, ver. 1.0.4).

Karatzoglou, Alexandros, Smola, Alex, Hornik, Kurt, and Zeileis, Achim, 2004, kernlab—An 'S4' package for kernel methods in 'R': Journal of Statistical Software, v. 11, no. 9, p. 1–20, accessed July 17, 2017, at http://www.jstatsoft.org/v11/i09/.

Kates, Louis, and Petzoldt, Thomas, 2012, proto: Prototype object-based programming (R package, ver. 0.3-10).

Koenker, Roger, 2016, quantreg: Quantile regression (R package, ver. 5.24).

Koenker, Roger, and Ng, Pin, 2015, SparseM: Sparse linear algebra (R package, ver. 1.7).

Kohavi, Ron, and Provost, Foster, 1998, Glossary of terms: Journal of Machine Learning, v. 30, nos. 2–3, p. 271–274.

Komsta, Lukasz, and Novomestky, Frederick, 2015, moments: Moments, cumulants, skewness, kurtosis and related tests (R package, ver. 0.14).

Konis, Kjell, 2013, fit.models: fit.models (R package, ver. 0.5-10).

Korkmaz, Selcuk, Goksuluk, Dincer, and Zararsiz, Gokmen, 2014, MVN: An R package for assessing multivariate normality.: The R Journal, v. 6, no. 2, p. 151–162, accessed July 17, 2017, at https://journal.r-project.org/archive/2014-2/korkmaz-goksuluk-zararsiz.pdf.

Lachenbruch, P.A., and Mickey, M.R., 1968, Estimation of error rates in discriminant analysis: Technometrics, v. 10, no. 1, p. 1–11, doi: 10.1080/00401706.1968.10490530, accessed July 17, 2017, at http://www.tandfonline.com/doi/abs/10.1080/00401706.1968.10490530.

Lang , D.T., and the CRAN Team, 2016, XML: Tools for parsing and generating XML within R and S-Plus (R package, ver. 3.98-1.4).

Larsen, M.C., Gellis, A.C., Glysson, G.D., Gray, J.R., and Horowitz, A.J., 2010, Fluvial sediment in the environment: A national problem in Proceedings, in 2nd Joint Federal Interagency Conference, Las Vegas, Nevada, June 7–July 1, 2010, p. 15.

Lee, Lopaka, 2013, NADA: Nondetects And Data Analysis for environmental data (R package, ver. 1.5-6).

Leisch, Friedrich, and Dimitriadou, Evgenia, 2010, mlbench: Machine learning benchmark problems (R package, ver. 2.1-1).

Levene, H., 1960, Robust tests for equality of variances—Contributions to probability and statistics Palo Alto, California, Stanford University Press, p. 278–292.

Ligges, Uwe, and Mächler, Martin, 2003, Scatterplot3d—an R package for visualizing multivariate data. R package, ver. 0.3-37: Journal of Statistical Software, v. 8, no. 11, p. 1–20.

Little, R.J.A., and Rubin, D.B., 1987, Statistical analysis with missing data, New York, John Wiley & Sons, Inc., 278 p.

Lorenz, D.L., 2015a, smwrBase—An R package for managing hydrologic data, ver. 1.1.1: U.S. Geological Survey Open-File Report 2015–1202, 7 p.

Lorenz, D.L., 2015c, smwrData—An R Package of Example Hydrologic Data, ver. 1.1.1: U.S. Geological Survey Open-File report 2015–1103, 5 p.

Lubin, J.H., Colt, J.S., Camann, David, Davis, Scott, Cerhan, J.R., Severson, R.K., Bernstein, Leslie, and Hartge, Patricia, 2004, Epidemiologic evaluation of measurement data in the presence of detection limits: Environmental Health Perspectives, v. 112, no. 17, p. 1,691–1,696., doi:10.1289/ehp.7199, accessed July 17, 2017, at https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1253661/.

Lumley, Thomas, 2013, dichromat: Color schemes for dichromats (R package, ver. 2.0-0).

Lumley, Thomas, 2014, survey: Analysis of complex survey samples. R package version 3.30-3: Journal of Statistical Software., v. 9, no. 8, p. 1–19, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v009.i08.

Lumley, Thomas, and using Fortran code by Miller, Andy, 2009, leaps: regression subset selection (R package, ver. 2.9).

Maechler, Martin, and others, 2016, sfsmisc: Utilities from "Seminar fuer Statistik" ETH Zurich, Switzerland (R package, ver. 1.1-0).

Maechler, Martin, Rousseeuw, Peter, Struyf, Anja, Hubert, Mia, and Hornick, Kurt, 2016, cluster: Cluster analysis basics and extensions (R package, ver. 2.0.4).

Majure, J.J., and Gebhardt, Albrecht, 2016, sgeostat: An object-oriented framework for geostatistical modeling in S+ (R package, ver. 1.0-27).

Mardia, K.V., Kent, J.T., and Bibby, J.M., 1980, Multivariate Analysis (1st ed.): New York, Academic Press, 521 p.

Mersmann, Olaf, 2015, microbenchmark: Accurate timing functions (R package, ver. 1.4-2.1).

Metropolis, Nicholas, and Ulam, S.M., 1949, The Monte Carlo method: Journal of American Statistical Association, v. 44, no. 247, p. 335–341.

Mevik, Bjørn-Helge, Wehrnes, Ron, and Liland, K.H., 2015, pls: Partial least squares and principal component regression (R package, ver. 2.5-0).

Meyer, David, Dimitriadou, Evgenia, Hornik, Kurt, Weingessel, Andreas, and Leisch, Friedrich, 2015, e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. (R package, ver. 1.6-7).

Meyer, David, Zeileis, Achim, and Hornik, Kurt, 2015, vcd: Visualizing categorical data. R package, ver. 1.4-1: Journal of Statistical Software, v. 3, no. 17, p. 1–48.

Microsoft®, 2007, Decide whether to trust a database, Office Support, accessed July 19, 2017 at https://support.office.com/en-us/article/Decide-whether-to-trust-a-database-4fee685e-72dd-47f9-925d-57c6e9a86e02.

Miller, Jerry, Mackin, Gail, and Orbock Miller, S.M., eds., 2015, Sediment geochemisty, in application of geochemical tracers to fluvial sediment (1st ed.); Springer International Publishing, doi: 10.1007/978-3-319-13221-1.

Mueller, Kirill, 2014, kimisc: Kirill's miscellaneous functions (R package, ver. 0.2-1).

Mukundan, Rajith, Radcliff, D.E., Richie, J.C., Risse, L.M., and McKinley, R.A., 2010, Sediment fingerprinting to determine the source of suspended sediment in a southern Piedmont stream: Journal of Environmental Quality, v. 39, p. 1,328–1,337.

Nash, J.C., 2014a, On best practice optimization methods in R: Journal ofStatistical Software, v. 60, no. 2, p. 1–14, accessed July 17, 2017, at https://www.jstatsoft.org/article/view/v060i02.

Nash, J.C., 2014b, optextras: Tools to support optimization methods (R package, ver. 2013-10.28).

Nash, J.C., 2014c, Rcgmin: Conjugate gradient minimization of nonlinear functions (R package, ver. 2013-2.21 ed.).

Nash, J.C., 2016, Rvmmin: Variable metric nonlinear function minimization (R package, ver. 2013-11.12).

Nash, J.C., and Varadhan, Ravi, 2011, Unifying optimization algorithms to aid software system users: optimx for R: Journal of Statistical Software, v. 43, no. 9, p. 1–14, accessed July 17, 2017, at https://jhu.pure.elsevier.com/en/publications/unifying-optimization-algorithms-to-aid-software-system-users-opt-3.

Neuwirth, Erich, 2014, RColorBrewer: ColorBrewer palettes (R package, ver. 1.1-2).

Nielsen, H.B., and Mortensen, S.B., 2012, ucminf: General-purpose unconstrained non-linear optimization (R package, ver. 1.1-3).

Novomestky, Frederick, 2012, matrixcalc: Collection of functions for matrix calculations (R package, ver. 1.0-3).

Oksanen, Jari, Blanchet, F.G., Friendly, Michael, Kindt, Roeland, Legendre, Pierre, McGlinn, Dan, Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, Peter, Stevens, M.H.H., Szoecs, Eduard, and Wagner, Helene, 2016, vegan: Community ecology package (R package, ver. 2.4-1).

Ooms, Jeroen, 2014, The jsonlite Package: A practical and consistent mapping between JSON data and R objects: arXiv:1403.2805 [stat.CO], v. 1, p. 1–29, accessed July 17, 2017, at http://www.stat.wvu.edu/~jharner/courses/stat623/docs/json-mapping.pdf.

Ooms, Jeroen, 2016a, curl: A modern and flexible web client for R (R package, ver. 0.9.7).

Ooms, Jeroen, 2016b, openssl: Toolkit for encryption, signatures and certificates based on OpenSSL (R package, ver. 0.9.3).

Ooms, Jeroen, James, David, DebRoy, Saikat, Wickham, Hadley, and Horner, Jeffrey., 2016, RMySQL: Database interface and 'MySQL' driver for R (R package, ver. 0.10.9).

Palarea-Albaladejo, Javier, and Martin-Fernandez, J.A. 2015, zCompositions: R package for multivariate imputation of left-censored data under a compositional approach. R package, ver. 1.0.3-1: Chemometrics and Intelligent Laboratory Systems, v. 143, p. 85–96.

Palarea-Albaladejo, Javier, and Martin-Fernandez, J.A., 2014, zCompositions: Imputation of zeros and nondetects in compositional data sets. R package, ver. 1.0.3-1, accessed July 17, 2017, at https://rdrr.io/cran/zCompositions/man/zCompositions-package.html#heading-2.

Pebesma, E.J., and Bivand, R.S., 2005, Classes and methods for spatial data in R (R package, ver. 1.2-3).

Pinheiro, Jose, Bates, Douglas, DebRoy, Saikat, Deepayan, Sarkar, EISPACK authors, Heisterkamp, Siem, Van Willigen, Bert, and R Core Team, 2016, nlme: Linear and nonlinear mixed effects models (R package, ver. 3.1-128).

Plate, Tony, and Heiberger, Richard, 2015, abind: Combine multidimensional arrays (R package, version 1.4-3).

Plummer, Martyn, Best, Nicky, Cowles, Kate, and Vines, Karen, 2006, CODA: Convergence diagnosis and output analysis for MCMC: R News, v. 6, no. 1, p. 7-11.

Provost, Foster, and Kohavi, Ron, 1998, On applied research in machine learning: Machine Learning, v. 30, no. 2, p. 127–132.

Qiu, Yixuan, and Xie, Yihui, 2016, highr: Syntax highlighting for R source code. (R package, ver. 0.6).

R Core Team, 2016a, R: A language and environment for statistical computing: Vienna, Austria, R Foundation for Statistical Computing.

R Core Team, 2016b, R: A Language and Environment for Statistical Computing (ver. 3.3.0): Vienna, Austria, R Foundation for Statistical Computing.

R Special Interest Group on Databases (R-SIG-DB), Wickham, Hadley, and Muller, Kirill, 2016, DBI: R database interface (R package, ver. 0.4-1).

Revelle, William, 2016, psych: Procedures for psychological, psychometric, and personality research (R package, ver. 1.6.4).

Revolution Analytics, and Weston, Steve, 2015a, doParallel: Foreach parallel adaptor for the 'parallel' package (R package, ver. 1.0.10).

Revolution Analytics, and Weston, Steve, 2015b, foreach: Provides foreach looping construct for R (R package, ver. 1.4.3).

Revolution Analytics, and Weston, Steve, 2015c, iterators: Provides iterator construct for R (R package, ver. 1.0.8).

Ripley, Brian, and Lapsley, Michael, 2015, RODBC: ODBC database access (R package, ver. 1.3-11).

Ripley, Brian, and Lapsley, Michael, 2016, RODBC: ODBC database access (R package, ver. 1.3-13).

Rosseel, Yves, 2012, lavaan: An 'R' package for structural equation modeling: Journal of Statistical Software, v. 48, no. 2, p. 1–3, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v048.i02.

Rousseeuw, Peter, Croux, Christophe, Todorov, Valentin, Ruckstuhl, Andreas, Salibian-Barrera, Matias, Verbeke, Tobias, Koller, Manuel, and Maechler, Martin, 2015, robustbase: Basic robust statistics (R package, ver. 0.92-5).

S original by Turlach, B.A., and R port by Weingessel, Andreas, 2013, quadprog: Functions to solve quadratic programming problems (R package, ver. 1.5-5).

Sarkar, D., 2008, Lattice: Multivariate Data Visualization with R (1st ed.): New York, Springer, 268 p.

Schauberger, Gunther, and Tutz, Gerhard, 2014, catdata: Categorical data (R package, ver. 1.2.1).

Schloerke, Barret, Crowley, Jason, Cook, Di, Briatte, Francois, Marbach, Moritz, Thoen, Edwin, and Elberg, Amos, 2016, GGally: Extension to ggplot2 (R package, ver. 1.0.1).

Shapiro, S.S., and Wilk, M.B., 1965, An analysis of variance test for normality (complete samples): Biometrika, v. 52, nos. 3–4, p. 591–611.

Simpson, G.L., 2016, permute: Functions for generating restricted permutations of data (R package, ver. 0.9-4).

Stephens, Jeremy, 2014, yaml: Methods to convert R data to YAML and back (R package, ver. 2.1.13).

Su, Yu-Sung, Gelman, Andrew, Hill, Jennifer, and Yajima, Masanao, 2011, Multiple imputation with diagnostics (mi) in R: Opening windows into the black box: Journal of Statistical Software, v. 45, no. 2, p. 1–31, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v045.i02.

Talbot, Justin, 2014, labeling: Axis labeling (R package, ver. 0.3).

Templ, Matthias, Alfons, Andreas, Kowarik, Alfons, and Prantner, Bernd, 2015, VIM: Visualization and imputation of missing values (R package, ver. 4.4.1).

Templ, Matthias, Hron, Karel, and Filzmoser, Peter, 2011, robCompositions: an R-package for robust statistical analysis of compositional data (R package, ver. 2.0.0).

Therneau, Terry, 2014, bdsmatrix: Routines for block diagonal symmetric matrices (R package, ver. 1.3-2).

Therneau, Terry, Atkinson, Beth, and Ripley, Brian, 2015, rpart: Recursive partitioning and regression trees (R package, ver. 4.1-10).

Therneau, T.M., 2015, coxme: Mixed effects cox models (R package, ver. 2.2-5).

Therneau, T.M., 2016, Survival analysis (R package, ver. 2.39-5).

Tierney, Luke, 2015, codetools: Code analysis tools for R (R package, ver. 0.2-14).

Todorov, Valentin, and Filzmoser, Peter, 2009, An object-oriented framework for robust multivariate analysis. R package, ver. 1.3-11: Journal of Statistical Software, v. 32, no. 3, p. 1–47, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v032.i03.

Trautmann, Heike, Steuer, Detlef, and Mersman, Olaf, 2014, truncnorm: Truncated normal distribution (R package, ver. 1.0-7).

Tukey, J.W., 1977, Exploratory data analysis (18th ed.): Reading, Massachusetts, Addison-Wesley Publishing Company, 688 p.

Urbanek, Simon, 2014, RJDBC: Provides access to databases through the JDBC interface (R package, ver. 0.2-5).

Urbanek, Simon, 2016a, rJava: Low-level R to Java interface (R package, ver. 0.9-8).

Urbanek, Simon, 2016b, sqldf: Perform SQL selects on R data frames. (R package, ver. 0.9-8).

Ushey, Kevin, Hester, Jim, and Krzyzanowski, Robert, 2016, rex: Friendly regular expressions. (R package, ver. 1.1.1).

Varadhan, Ravi, and Gilbert, Paul, 2009, BB: An R Package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function.: Journal of Statistical Software, v. 32, no. 4, p. 1–26, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v032.i04.

Varadhan, Ravi, Johns Hopkins University, Borchers, H.W., and ABB Corporate Research, 2011, dfoptim: Derivative-free optimization (R package, ver. 2011.8-1).

Venables, W.N., and Ripley, B.D., 2002, Modern Applied Statistics with S: New York, Springer, 498 p.

Walker, Alexander, 2015, openxlsx: Read, write and edit XLSX Files (R package, ver. 3.0.0.).

Wand, Matt, 2015, KernSmooth: Functions for kernel smoothing supporting Wand & Jones (1995) (R package, ver. 2.23-15).

Wang, Jiahui, Zamar, Ruben, Marazzi, Alfio, Yohai, Victor, Salibian-Barrera, Matias, Maronna, Ricardo, Zivot, Eric, Rocke, David, Martin, Doug, Maechler, Martin, and Konis, Kjell, 2014, robust: Robust library (R package, ver. 0.4-16).

Wang, Xiao-Feng, 2012, sROC: Nonparametric smooth ROC curves for continuous data (R package, ver. 0.1-2).

Weihs, Claus, Ligges, Uwe, Luebke, Karsten, and Raabe, Nils, 2005, klaR Analyzing german business cycles, Data Analysis and Decision Support: Berlin, Springer-Verlag, p. 335–343.

Weisberg, Sanford, 2014, Applied Linear Regression (4th ed.): Hoboken, NJ, Wiley, p. 384

Weston, Steve, and Wickham, Hadley, 2014, itertools: Iterator tools (R package, ver. 0.1-3).

Wickham, C., 2016a, munsell: Utilities for Using Munsell Colours. (R package, ver. 0.4.3 ed.).

Wickham, Hadley, 2007, Reshaping Data with the reshape package: Journal of Statistical Software, v. 21, no. 12, p. 1–20, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v021.i12.

Wickham, Hadley, 2009, ggplot2: Elegant graphics for data analysis (1st ed.): New York, Springe, 213 p.

Wickham, Hadley, 2011, testthat: Get started with testing: The R Journal, v. 3, no. 1, p. 5-10, accessed July 17, 2017, at https://journal.r-project.org/archive/2011-1/RJournal_2011-1.pdf.

Wickham, Hadley, 2011, The split-apply-combine strategy for data analysis: Journal of Statistical Software, v. 40, no. 1, p. 1–29, accessed July 17, 2017, at http://dx.doi.org/10.18637/jss.v040.i01.

Wickham, Hadley, 2013, assertthat: Easy pre and post assertions. (R package, ver. 0.1).

Wickham, Hadley, 2015a, lazyeval: Lazy (non-standard) evaluation. (R package, ver. 0.1.10).

Wickham, Hadley, 2015b, pryr: Tools for computing on the language (R package, ver. 0.1.2).

Wickham, Hadley, 2015c, stringr: Simple, consistent wrappers for common string operations (R package, ver. 1.0.0).

Wickham, Hadley, 2016b, evaluate: Parsing and evaluation tools that provide more details than the defualt (R package, ver. 0.9).

Wickham, Hadley, 2016c, gtable: Arrange ‘Grobs’ in tables (R package, ver. 0.2.0).

Wickham, Hadley, 2016d, httr: Tools for working with URLs and HTTP (R package, ver. 1.1.0).

Wickham, Hadley, 2016e, nycflights13: Flights that departed NYC in 2013 (R package, ver. 0.2.0).

Wickham, Hadley, 2016f, scales: Scale functions for visualization. (R package, ver. 0.4.0).

Wickham, Hadley, and Francois, Romain, 2015, readr: Read tabular data (R package, ver. 0.2.2).

Wickham, Hadley, Hester, Jim, and Muller, Kirill, 2016, memoise: Memoisation of functions (R package, ver. 1.0.0).

Wickham, Hadley, James, D.A., and Falcon, Seth, 2014, RSQLite: SQLite interface for R (R package, ver. 1.0.0).

Wood, Simon, and Scheipl, Fabian, 2014, gamm4: Generalized additive mixed models using mgcv and lme4 (R package, ver. 0.2-3).

Wood, S.N., 2000, Modelling and smoothing parameter estimation with multiple quadratic penalties: Journal of the Royal Statistical Society: Series B, v. 62, no. 2, p. 413–428, accessed July 17, 2017, at http://onlinelibrary.wiley.com/doi/10.1111/1467-9868.00240/abstract.

Wood, S.N., 2004, Stable and efficient multiple smoothing parameter estimation for generalized additive models: Journal of the American Statistical Association, v. 99, no. 467, p. 673–686, doi: 10.1198/016214504000000980, accessed July 17, 2017, at http://www.tandfonline.com/doi/abs/10.1198/016214504000000980.

Wood, S.N., 2006, Generalized additive models: An introduction with R., Chapman and Hall/CRC, Boca Raton, Florida, 410 p.

Wood, S.N., 2011, Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models: Journal of the Royal Statistical Society: Series B, v. 73, no. 1, p. 3–36, accessed July 17, 2017, at http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2010.00749.x/abstract.

Xie, Yihui, 2014, knitr: A comprehensive tool for reproducible research in R., implementing reproducible computational research. Chapman and Hall/CRC. Boca Raton, Florida, IBSN 978-1466561595

Xie, Yihui, 2015a, Dynamic documents with R and knitr (1st ed.); Chapman and Hall/CRC, 216 p.

Xie, Yihui, 2015b, mime: Map filenames to MIME types. (R package, ver. 0.4).

Xie, Yihui, 2016a, formatR: Format R code automatically. (R package, ver. 1.4).

Xie, Yihui, 2016b, knitr: A General-Purpose Package for Dynamic Report Generation in R. (R package, ver. 1.13).

Yan, Jun, 2010, som: Self-organizing map (R package, ver. 0.3-5).

Yeo, In-Kwon, and Johnson, R.A., 2000, A family of power transformation to improve normality or symmetry: Biometrika, v. 87, p. 954–959, accessed July 17, 2017, at https://www.jstor.org/stable/2673623?seq=1#page_scan_tab_contents.

Zeileis, Achim, 2004, Econometric computing with HC and HAC covariance matrix estimators. R package, ver. 2.3-4: Journal of Statistical Software, v. 11, no. 10, p. 1–17.

Zeileis , Achim, and Grothendieck, Gabor, 2005, zoo: S3 Infrastructure for regular and irregular time series. R package, ver. 1.7-13: Journal of Statistical Software, v. 14, no. 6, p. 1–27.

Zeileis, Achim, and Hothorn, Torsten, 2002, Diagnostic checking in regression relationships: R News, v. 2, no. 3, p. 7–10.

Zimmerman, D.W., 1994, A note on the influence of outliers on parametric and nonparametric tests: Journal of General Psychology, v. 121, no. 4, p. 391–401.

Appendix 1. Sed_SAT File Structure

Altering the program's file structure will result in program failure. Refer to fig. 1–1 for the required file structure supported and supplied by Sed_SAT.

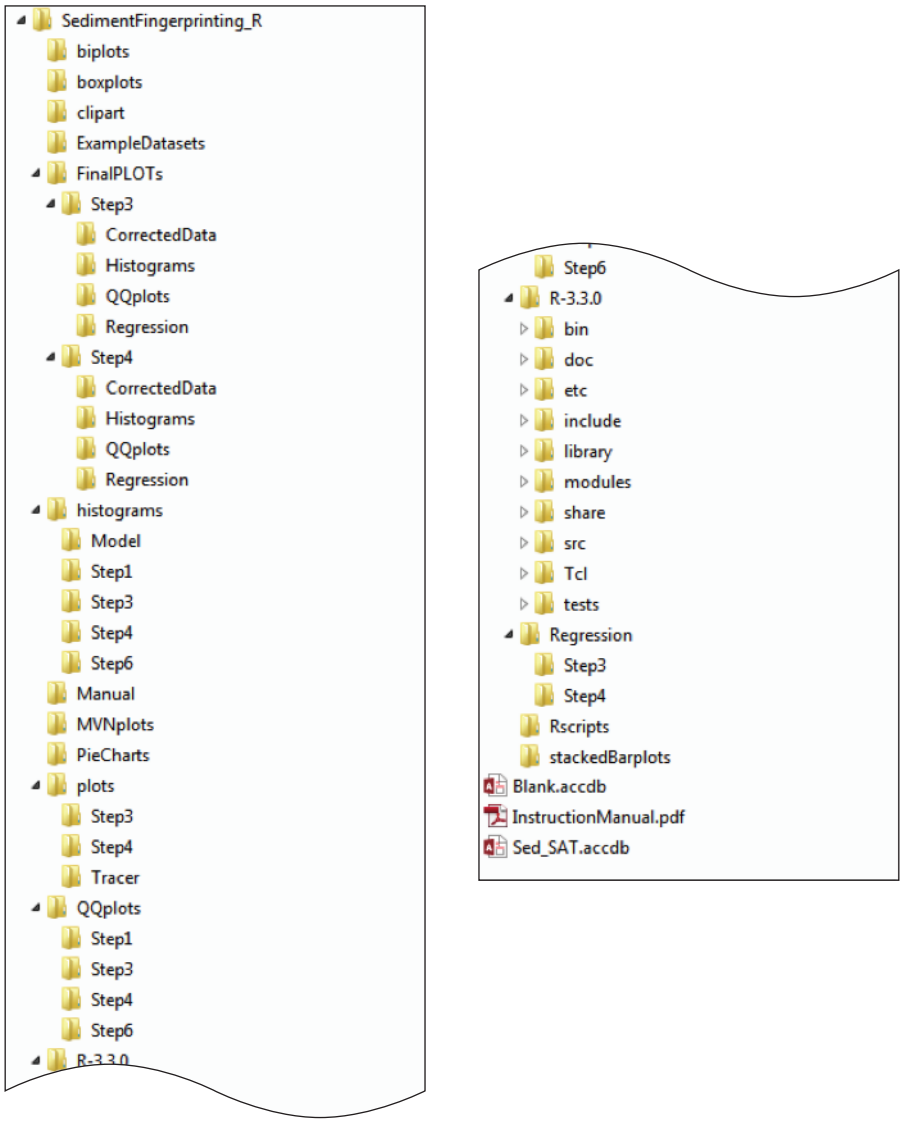


Figure 1–1. Sed_SAT required file structure.

Appendix 2. Example Datasets

Table 2–1 gives column information on the example data included in Sed_SAT.

Table 2–1. Example Dataset column details.

Column type	Tracer type	Source data column name	Target data column name	Reporting limit column name
SampleID		SampleName	SampleName	SampleID
Type		SourceType	SampleType	SourceType
Size data		Size	D50	
Organic content data/tracer		TOC	C_Organic_Mass_Fraction	
Tracer	elemental analysis (metals)	Aluminum	Aluminum	Aluminum_RL
Tracer	elemental analysis (metals)	Arsenic	Arsenic	Arsenic_RL
Tracer	elemental analysis (metals)	Barium	Barium	Barium_RL
Tracer	elemental analysis (metals)	Beryllium	Beryllium	Beryllium_RL
Tracer	elemental analysis (metals)	Cadmium	Cadmium	Cadmium_RL
Tracer	elemental analysis (metals)	Chromium	Chromium	Chromium_RL
Tracer	elemental analysis (metals)	Cobalt	Cobalt	Cobalt_RL
Tracer	elemental analysis (metals)	Copper	Copper	Copper_RL
Tracer	elemental analysis (metals)	Iron	Iron	Iron_RL
Tracer	elemental analysis (metals)	Lead	Lead	Lead_RL
Tracer	elemental analysis (metals)	Magnesium	Magnesium	Magnesium_RL
Tracer	elemental analysis (metals)	Manganese	Manganese	Manganese_RL
Tracer	elemental analysis (metals)	Nickel	Nickel	Nickel_RL
Tracer	elemental analysis (metals)	Potassium	Potassium	Potassium_RL
Tracer	elemental analysis (metals)	Strontium	Strontium	Strontium_RL
Tracer	elemental analysis (metals)	Uranium	Uranium	Uranium_RL
Tracer	elemental analysis (metals)	Vanadium	Vanadium	Vanadium_RL
Tracer	elemental analysis (metals)	Zinc	Zinc	Zinc_RL
Tracer	stable isotope data	Delta_13C_Organic	Delta_13C_Organic	
Tracer	stable isotope data	Delta_15N	Delta_15N	
Tracer	radionuclide data	Cs137	Cs137	
Tracer	radionuclide data	210PbXs	210PbXs	

Appendix 3. Size and Organic Content Data

If your dataset contains size and (or) organic content data, *Start Step 3: First Linear Regression* and (or) *Start Step 4: Second Linear Regression for Organic Content* are performed to adjust the source sample tracer concentration values in order to make them comparable to the target sample values. See Start Step 3: First Linear Regression for details on the linear regression procedure. Applying correction factors for size and organic content is not required, but is recommended (Collins and others, 2010; Gellis, Noe, and others, 2016).

Table 3–1. How Sed_SAT utilizes size and organic content data.

Data type	Example	Purpose	Rules
Size data	Grain size Surface area	To correct source tracer concentration values that are shown to have a significant relation with size making the source dataset comparable to the target dataset	If present, it MUST be in the 3rd column in BOTH the source and target datasets. Missing values are permitted Must be all positive, nonzero numbers
Organic content data	Total organic carbon (TOC) Loss on ignition (LOI)	To correct source tracer concentration values that are shown to have a significant relation with organic content making the source dataset comparable to the target dataset.	If present and size data are also present, it MUST be in the 4th column in BOTH the source and target datasets. If present and size data are NOT present, it MUST be in the 3rd column in BOTH the source and target datasets. Must be all positive, nonzero numbers Missing values are ONLY permitted for LOI not TOC. While both TOC and LOI can be used to correct *source* data for bias associated with organic content, TOC is also a tracer that can be used in the mixing model and therefore cannot contain missing values. LOI on the other hand is a mass loss of combustion residue rather than a concentration and cannot be considered a tracer. Since LOI is only used to correct for bias associated with organic content, missing values are permitted.

Appendix 4. SetPATHs Screen

On opening, this screen will search the computer for the relevant programs and autofill paths if possible. See SetPATHs Instructions for more information on how to set paths for Sed_SAT.

Figure 4–1. SetPATHs screen.

Appendix 5. Information on R-Packages Used in Sed_SAT

The following R-packages are included with R version 3.3.0 (R Core Team, 2016b) in the directory SedimentFingerprinting_R/ R-3.3.0/library. The "core packages" are directly used in the Sed_SAT scripts, while the dependent packages are required by the core packages. Click the package name to view the package documentation on The Comprehensive R Archive Network (2015a; R Core Team, 2016b). Core packages are initialized at the beginning of each Rscript through the .Rprofile file located in the "Rscripts" folder. The .Rprofile is internal to the program and will not alter a user's personal .Rprofile.

Table 5–1. Core packages used by Sed_SAT.

Package	Version	Use	Reference
RODBC	1.3-13	connect to .accdb files	(Ripley and Lapsley, 2015)
klaR	0.6-12	forward stepwise linear discriminant function analysis using Wilk's lambda criterion	(Weihs and others, 2005):
MASS	7.3-45	linear discriminant function analysis	(Venables and Ripley, 2002)
moments	0.14	calculate skewness and kurtosis	(Komsta and Novomestky, 2015)
zCompositions	1.0.3-1	single and multiple imputation	(Palarea-Albaladejo and Martin-Fernandez, 2014)
openxlsx	3.0.0	reading Microsoft Excel files	(Walker, 2015)
kimisc	0.3	locating files	(Mueller, 2014)
dplyr	0.4.3	counting records	(Wickham and Francois, 2015)
sqldf	0.4-10	query functions	(Grothendieck, 2014; Urbanek, 2016b)
car	2.1-2	find Box Cox power transformations	(Fox and Weisberg, 2011)
smwrBase	1.1.2	Apply Box Cox power transformations maintaining original scale of data	(Lorenz, 2015b)
mvnormtest	0.1-9	Shapiro-Wilk multivariate normality test	(Jarek, 2012)
MVN	4.0	multivariate normality tests	(Korkmaz and others, 2014)
colorspace	1.2-6	generate color pallets for plots	(Ihaka and others, 2009)
ggplot2	2.1.0	create biplot in Step 7	(Wickham, 2009)
ggord	0.11.9000	create biplot in Step 7	(Beck, 2016)
vegan	2.4-1	homoscedasticity test	(Oksanen and others, 2016)

Table 5-2. Dependent packages required by the core packages within Sed_SAT.

Package	Version	Reference
abind	1.4-3	(Plate and Heiberger, 2015)
alr4	1.0.5	(Weisberg, 2014)
arm	1.8-6	(Gelman and Su, 2015)
assertthat	0.1	(Wickham, 2013)
base	3.3.0	(R Core Team, 2016a)
BB	2014.10-1	(Varadhan and Gilbert, 2009)
bdsmatrix	1.3-2	(Therneau, 2014)
BH	1.60.0-2	(Eddelbuettel, Emerson, and others, 2016)
boot	1.3-18	(Davison and Hinkley, 1997; Canty and Ripley, 2016)
catdata	1.2.1	(Schauberger and Tutz, 2014)
chron	2.3-47	(James and Hornick, 2015)
class	7.3-14	(Venables and Ripley, 2002)
cluster	2.0.4	(Maechler, Rousseeuw, and others, 2016)
coda	0.18-1	(Plummer and others, 2006)
codetools	0.2-14	(Tierney, 2015)
combinat	0.0-8	(Chasalow, 2012)
compiler	3.1.3	(R Core Team, 2016a)
covr	2.0.1	(Hester, 2016)
coxme	2.2-5	(Therneau, 2015)
crayon	1.3.1	(Csardi, 2015)
curl	0.9.7	(Ooms, 2016a)
cvTools	0.3.2	(Alfons, 2012)
data.table	1.9.6	(Dowle and others, 2015)
dataRetrieval	2.5.5	(Hirsch and De Cicco, 2015)
datasets	3.3.0	(R Core Team, 2016a)
DBI	0.4-1	(R Special Interest Group on Databases (R-SIG-DB) and others, 2016)
DEoptimR	1.0-4	(Conceicao and Maechler, 2015)
dfoptim	2011.8-1	(Varadhan and others, 2011)
dichromat	2.0-0	(Lumley, 2013)
digest	0.6.9	(Eddelbuettel, Lucas, and others, 2016)
doBy	4.5-15	(Højsgaard and Halekoh, 2016)
doParallel	1.0.10	(Revolution Analytics and Weston, 2015a)
e1071	1.6-7	(Meyer, Dimitriadou, and others, 2015)
effects	3.1-1	(Fox, 2003)
evaluate	0.9	(Wickham, 2016b)
expm	0.999-0	(Goulet and others, 2015)

Table 5-2. Dependent packages required by the core packages within Sed_SAT.—Continued

Package	Version	Reference
fit.models	0.5-10	(Konis, 2013)
foreach	1.4.3	(Revolution Analytics and Weston, 2015b)
formatR	1.4	(Xie, 2016a)
gamm4	0.2-3	(Wood and Scheipl, 2014)
GGally	1.0.1	(Schloerke and others, 2016)
GPArotation	2014.11-1	(Bernaards and Jennrich, 2005)
graphics	3.3.0	(R Core Team, 2016a,)
grDevices	3.3.0	(R Core Team, 2016a, b)
grid	3.1.3	(R Core Team, 2016a, b)
gsubfn	0.6-6	(Grothendieck, 2014a)
gtable	0.2.0	(Wickham, 2016c)
highr	0.6	(Qiu and Xie, 2016)
HSAUR2	1.1-14	(Everiit and Hothorn, 2015)
httr	1.1.0	(Wickham, 2016d)
iterators	1.0.8	(Revolution Analytics and Weston, 2015c)
itertools	0.1-3	(Weston and Wickham, 2014)
jsonlite	0.9.20	(Ooms, 2014)
kernlab	0.9-24	(Karatzoglou and others, 2004)
KernSmooth	2.23-15	(Wand, 2015)
knitr	1.13	(Xie, 2014, 2015a, 2016b)
labeling	0.3	(Talbot, 2014)
Lahman	4.0-1	(Friendly, 2015)
lattice	0.20-33	(Sarkar, 2008)
lavaan	0.5-20	(Rosseel, 2012)
lazyeval	0.1.10	(Wickham, 2015a)
leaps	2.9	(Lumley and using Fortran code by Miller, 2009)
lme4	1.1-12	(Bates and others, 2015)
lmtest	0.9-34	(Zeileis and Hothorn, 2002)
lubridate	1.5.6	(Grolemund and Wickham, 2011)
magrittr	1.5	(Bache and Wickham, 2014)
markdown	0.7.7	(Allaire and others, 2015)
Matrix	1.2-6	(Bates and Maechler, 2016)
matrixcalc	1.0-3	(Novomestky, 2012)
MatrixModels	0.4-1	(Bates and Maechler, 2015)
memoise	1.0.0	(Wickham and others, 2016)
MEMSS	0.9-2	(Bates, Maechler, and others, 2014a)

Table 5-2. Dependent packages required by the core packages within Sed_SAT.—Continued

Package	Version	Reference
methods	3.3.0	(R Core Team, 2016a, b)
mgcv	1.8-12	(Wood, 2000, 2004, 2006, 2011)
mi	1.0	(Su and others, 2011)
microbenchmark	1.4-2.1	(Mersmann, 2015)
mime	0.4	(Xie, 2015b)
minqa	1.2.4	(Bates, Mullen, and others, 2014)
mlbench	2.1-1	(Leisch and Dimitriadou, 2010)
mlmRev	1.0-6	(Bates, Maechler, and others, 2014b)
mnormt	1.5-4	(Azzalini and Genz, 2016)
MPV	1.38	(Braun, 2015)
munsell	0.4.3	(Wickham, 2016a)
mvoutlier	2.0.6	(Filzmoser and Gschwandtner, 2015)
mvtnorm	1.0-5	(Genz and Bretz, 2009; Genz and others, 2016)
NADA	1.5-6	(Lee, 2013)
nlme	3.1-128	(Pinheiro and others, 2016)
nloptr	1.0.4	(Johnson, 2014)
nnet	7.3-12	(Venables and Ripley, 2002)
nortest	1.0-4	(Gross and Ligges, 2015)
numDeriv	2014.2-1	(Gilbert and Varadhan, 2015)
nycflights13	0.2.0	(Wickham, 2016e)
openssl	0.9.3	(Ooms, 2016b)
optextras	2013-10.28	(Nash, 2014b)
optimx	2013.8.7	(Nash and Varadhan, 2011; Nash, 2014a)
permute	3.1.3	(Simpson, 2016)
parallel	3.1.3	(R Core Team, 2016a, b)
pbivnorm	0.6.0	(Fortran code by Genz and R code by Kenkel, 2015)
pbkrtest	0.4-6	(Halekoh and Højsgaard 2014)
pcaPP	1.9-60	(Filzmoser and others, 2014)
PKPDmodels	0.3.2	(Dubois and others, 2012)
pls	2.5-0	(Mevik and others, 2015)
plyr	1.8.3	(Wickham, 2011)
praise	1.0.0	(Csardi and Sorhus, 2015)
proto	0.3-10	(Kates and Petzoldt, 2012)
pryr	0.1.2	(Wickham, 2015b)
psych	1.6.4	(Revelle and Northwestern University, 2016)
quadprog	1.5-5	(S original by Berwin and Turlach R port by Weingessel, 2013)

Table 5-2. Dependent packages required by the core packages within Sed_SAT.—Continued

Package	Version	Reference
quantreg	5.24	(Koenker, 2016)
R6	2.1.2	(Chang, 2016)
Rcgmin	2013-2.21	(Nash, 2014c)
RColorBrewer	1.1-2	(Neuwirth, 2014)
Rcpp	0.12.5	(Eddelbuettel and Francois, 2011; Eddelbuettel, 2013)
Rcsdp	0.1.55	(Bravo, 2016)
readr	0.2.2	(Wickham and Francois, 2015)
reshape	0.8.5	(Wickham, 2007)
reshape2	1.4.1	(Wickham, 2007)
rex	1.1.1	(Ushey and others, 2016)
rgl	0.95.1441	(Adler and others, 2016)
RH2	0.2.3	(Grothendieck and Mueller, 2014)
rJava	0.9-8	(Urbanek, 2016a; 2016b)
RJDBC	0.2-5	(Urbanek, 2014)
RMySQL	0.10.9	(Ooms and others, 2016)
robCompositions	2.0.0	(Templ and others, 2011)
robust	0.4-16	(Wang and others, 2014)
robustbase	0.92-5	(Rousseeuw and others, 2015)
RODBC	1.3-13	(Ripley and Lapsley, 2016)
rpart	4.1-10	(Therneau and others, 2015)
RPostgreSQL	0.4-1	(Conway and others, 2016)
rrcov	1.3-11	(Todorov and Filzmoser, 2009)
RSQLite	1.0.0	(Wickham and others, 2014)
Rvmmin	2013-11.12	(Nash, 2016)
sandwich	2.3-4	(Zeileis, 2004)
scales	0.4.0	(Wickham, 2016f)
scatterplot3d	0.3-37	(Ligges and Mächler, 2003)
sem	3.1-7	(Fox and others, 2016)
setRNG	2013.9-1	(Gilbert, 2014)
sfsmisc	1.1-0	(Maechler and others, 2016)
sgeostat	1.0-27	(Majure and Gebhardt, 2016)
smwrBase	1.1.1	(Lorenz, 2015a)
smwrData	1.1.1	(Lorenz, 2015c)
som	0.3-5	(Yan, 2010)
sp	1.2-3	(Pebesma and Bivand, 2005)
SparseM	1.7	(Koenker and Ng, 2015)

Table 5-2. Dependent packages required by the core packages within Sed_SAT.—Continued

Package	Version	Reference
splines	3.1.3	(R Core Team, 2016b)
sqldf	0.4-10	(Grothendieck, 2014b)
sROC	0.1-2	(Wang, 2012)
stats	3.3.0	(R Core Team, 2016b)
Stats4	3.1.3	(R Core Team, 2016b)
stringi	1.1.1	(Gagolewski and Tartanus, 2015)
stringr	1.0.0	(Wickham, 2015c)
survey	3.30-3	(Lumley, 2014)
survival	2.39-5	(Therneau, 2016)
svUnit	0.7-12	(Grosjean, 2016)
tcltk	1.2-11	(R Core Team, 2016b)
testthat	1.0.2	(Hadley, 2011)
tools	3.1.3	(R Core Team, 2016a, b)
truncnorm	1.0-7	(Trautmann and others, 2014)
ucminf	1.1-3	(Nielsen and Mortensen, 2012)
utils	2.3.0	(R Core Team, 2016b)
vcd	1.4-1	(Meyer, Zeileis, and others, 2015)
VIM	4.4.1	(Templ and others, 2015)
withr	1.0.2	(Hester and others, 2016)
XML	3.98-1.4	(Lang and the CRAN Team, 2016)
xtable	1.8-2	(Dahl, 2016)
yaml	2.1.13	(Stephens, 2014)
zCompositions	1.0.3-1	(Palarea-Albaladejo and Martin-Fernandez, 2015)
zoo	1.7-13	(Zeileis and Grothendieck, 2005)

Appendix 6. Stable Isotope Selection Screen

This screen will appear only if the user answers "Yes" to the question "Does your data contain stable isotope data?" in the *Basic Questions* section prior to importing data. See *Organic Content and Stable Isotope Data* for more information on how stable isotopes are handled in the program.

> ***Example Datasets Hint***
>
> The example datasets contain stable isotopes. See *appendix 1* for column details.

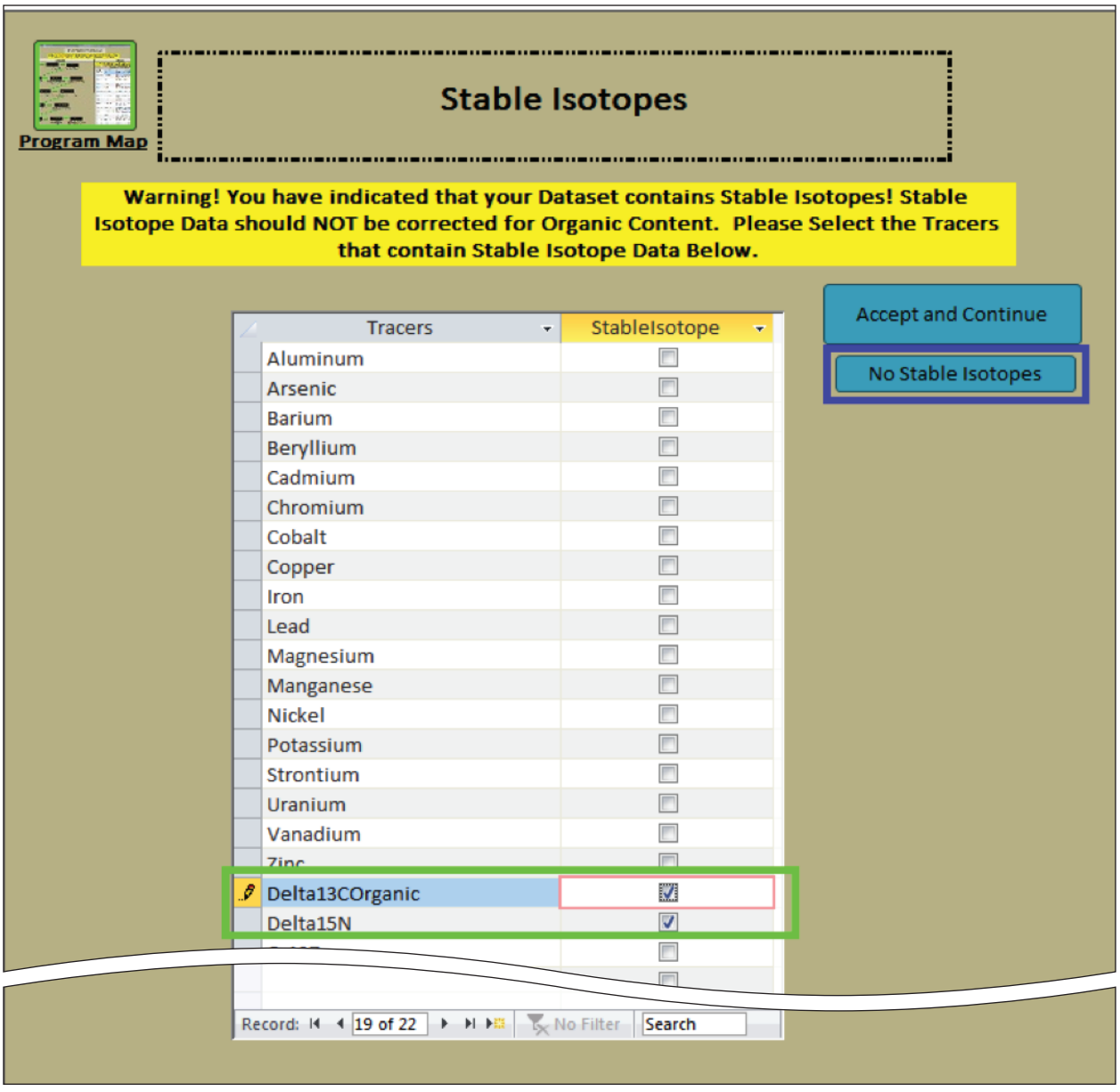


Figure 6–1. Stable Isotopes screen.

Appendix 7. Import Data Screens

See *Import Data* for more details on the import procedure.

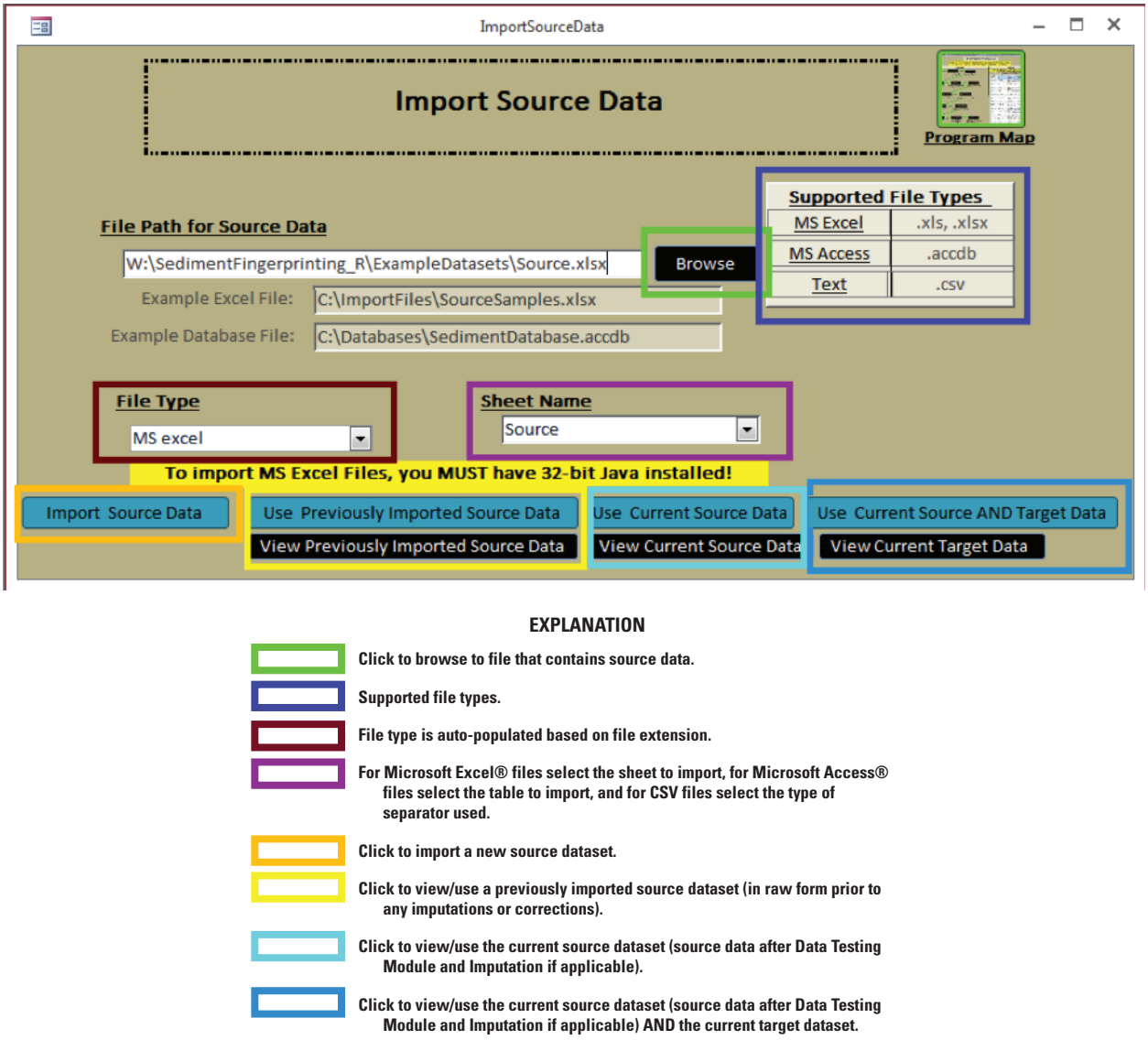
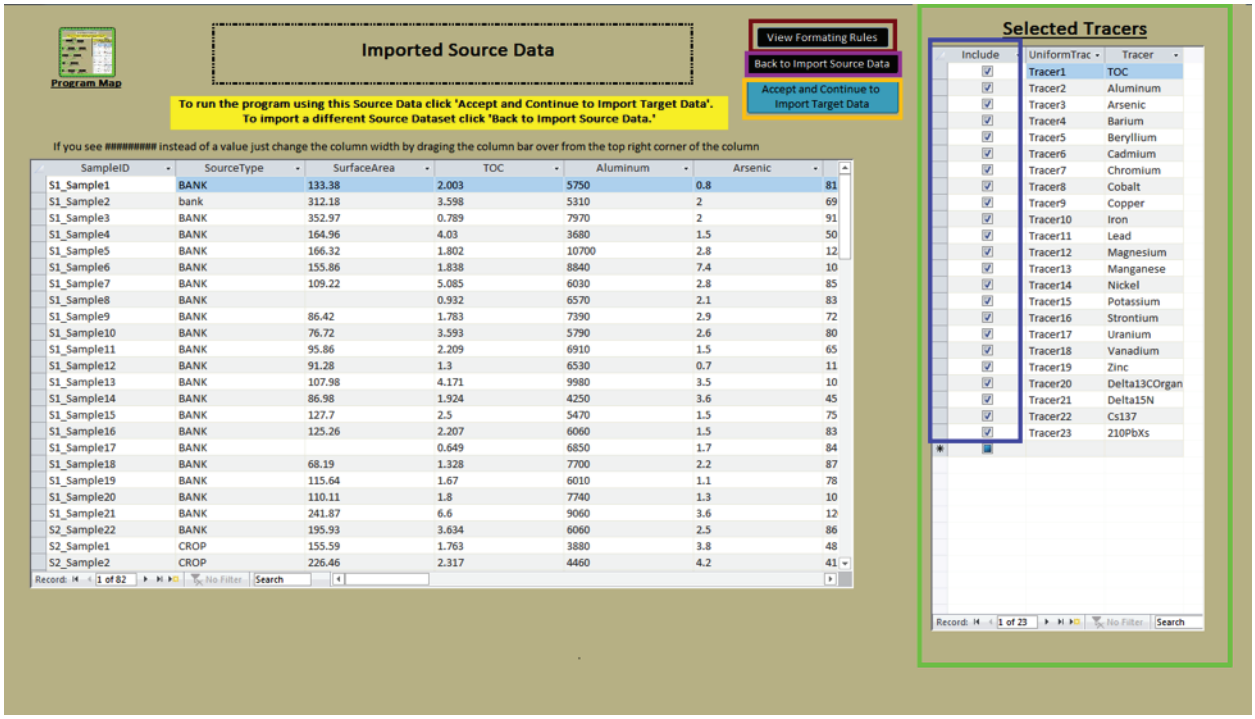


Figure 7–1. Import Source Data Screen.

> ***Example Datasets Hint***
>
> The example datasets come preloaded in the program. If running Sed_SAT for the first time, to use the example datasets either click "Use Previously Imported Source Data" or "Import Source Data" and import the example datasets from ~/SedimentFingerprinting_R/ExampleDatasets/.



EXPLANATION

Tracers imported in the source dataset.

Uncheck the "Include" checkbox to remove a tracer from the analysis. The tracer in the target dataset in the same location (will have same "UniformTracerName") will automatically be removed since the source dataset and target datasets must have the same tracers. Do not use this to subset out tracers that appear in the source dataset and not in the target dataset. Prior to import, both datasets MUST have the same number of tracers.

Click to view data formatting rules.

Click to re-import source dataset, if mistakes are found.

Click to accept the imported source dataset as the final source dataset and continue to the next step, importing the target dataset.

Figure 7–2. View and accept source data screen.

Appendix 8. Problems Found in the Data Testing Module

> ***Example Datasets Hint***
>
> Although these screens appear to have the same data as are present in the example datasets, they are for demonstration purposes only. Many of the problems shown in this section do not exist in the example datasets, with the exception of problems 4, 7, and 8, which can all be handled within the program with no need to alter the raw files.

For more details on the Data Testing Module procedure, see *Data Testing Module*.

Problem 1: Number of Fields in the Source Dataset ≠ Number of Fields in the Target Dataset

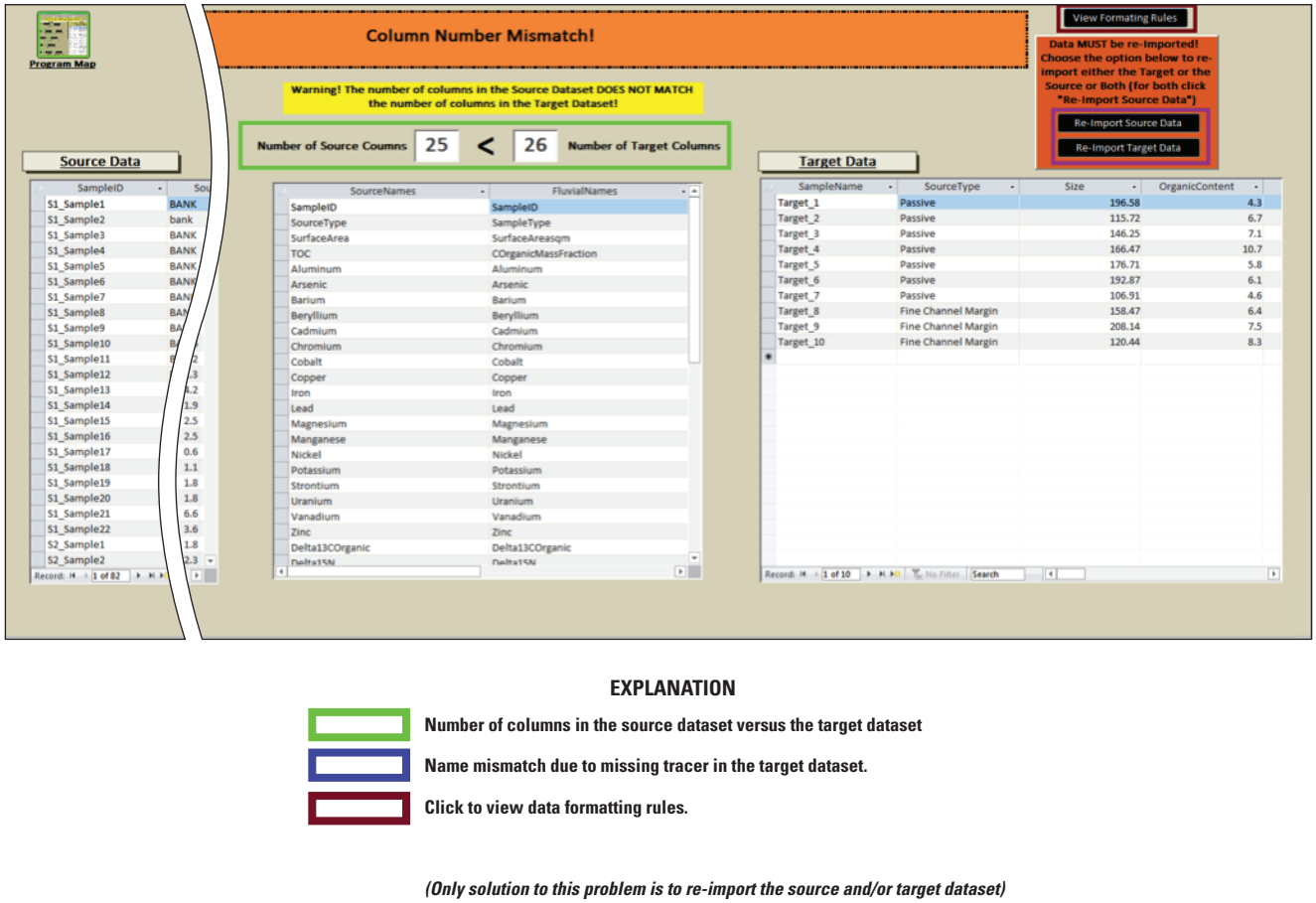
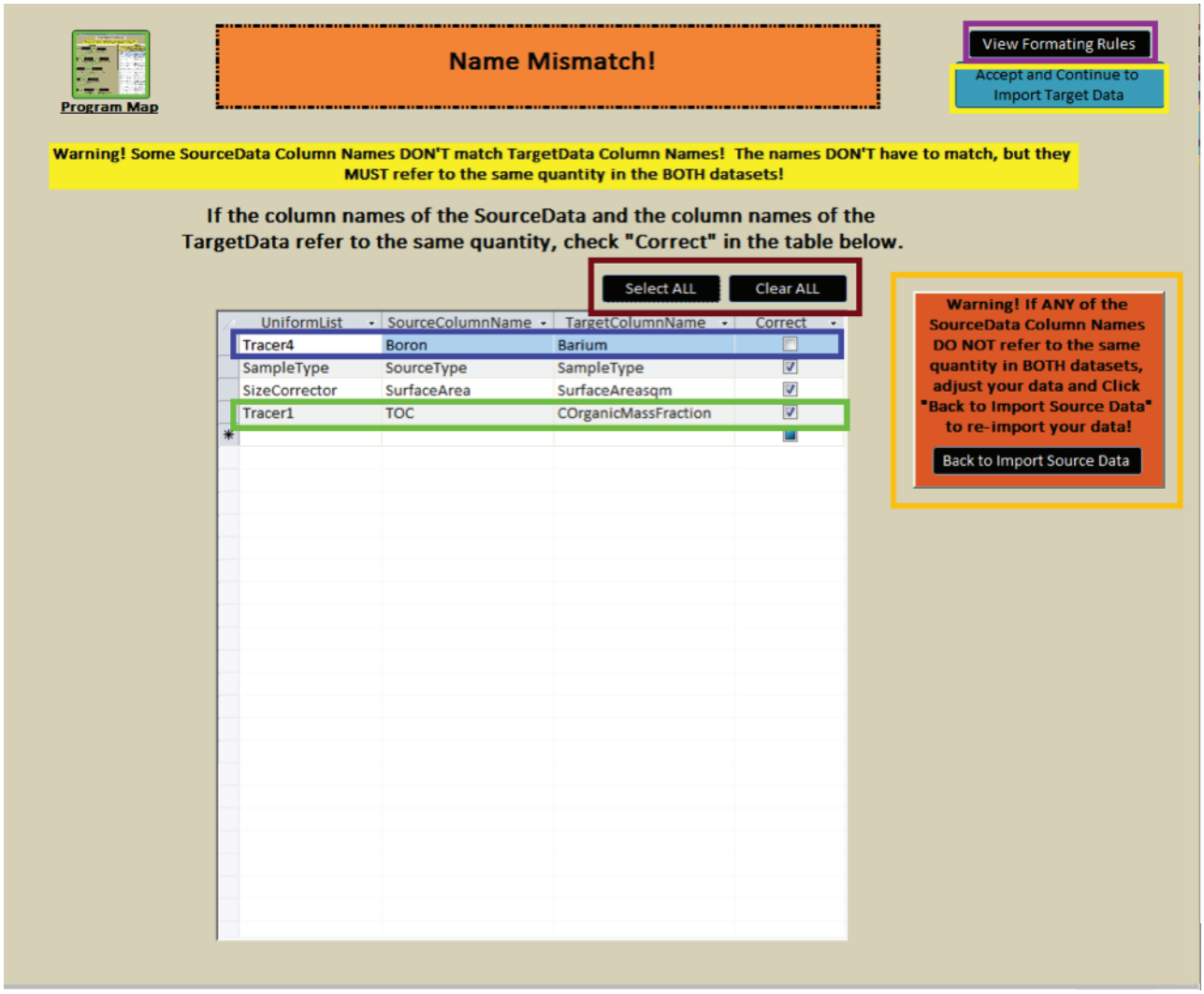


Figure 8–1. Column number mismatch screen will appear if Problem 1 has been found.

Problem 2: Tracers Not in the Same Relative Location (or Columns?) in Both the Source and Target Datasets

Example: Aluminum is in the fourth column in the source dataset and fifth column in the target dataset.



EXPLANATION

Text in tracer names differs, but refers to the same quantity. If all tracer names refer to the same quantity in both the source and the target datasets, check all boxes and click Accept and Continue to Import Target Data.

Tracer names refer to different quantities. If any tracer name refers to a different quantity in the source dataset versus the target dataset, the program must be restarted and the source dataset re-imported. Click Back to Import Source Data.

To check all "Correct" checkboxes, click Select ALL. To clear all checkboxes, click Clear ALL.

Click to view data formatting rules. See Data Formatting Rules for more details on formatting rules.

Figure 8–2. Name Mismatch screen will appear if Problem 2 has been found.

Problem 3: Non-Unique Sample Names

Sample names in both the source and target dataset must be unique

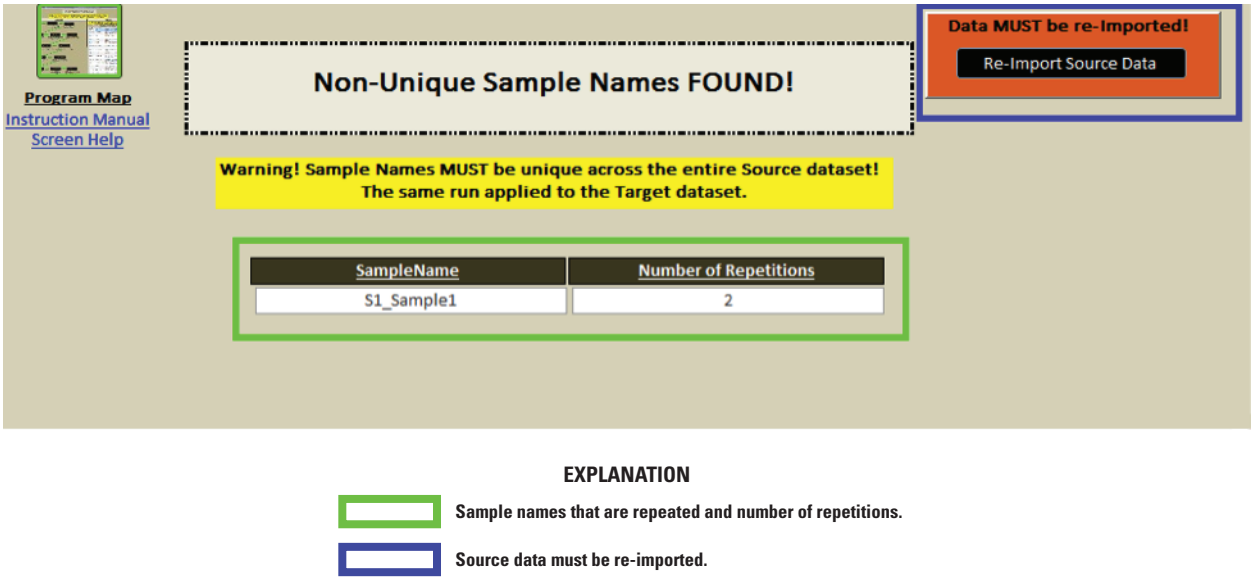


Figure 8–3. Non-unique sample names found.

Problem 4: Non-Unique Source Types

Source types must be labeled using exactly the same text throughout the source dataset (i.e. bank and BANK will be seen as different).

> ***Example Datasets Hint***
>
> The source "bank" MUST be marked as NOT unique. Select "BANK" as the unique source type from the dropdown list as shown in figure 8–4.

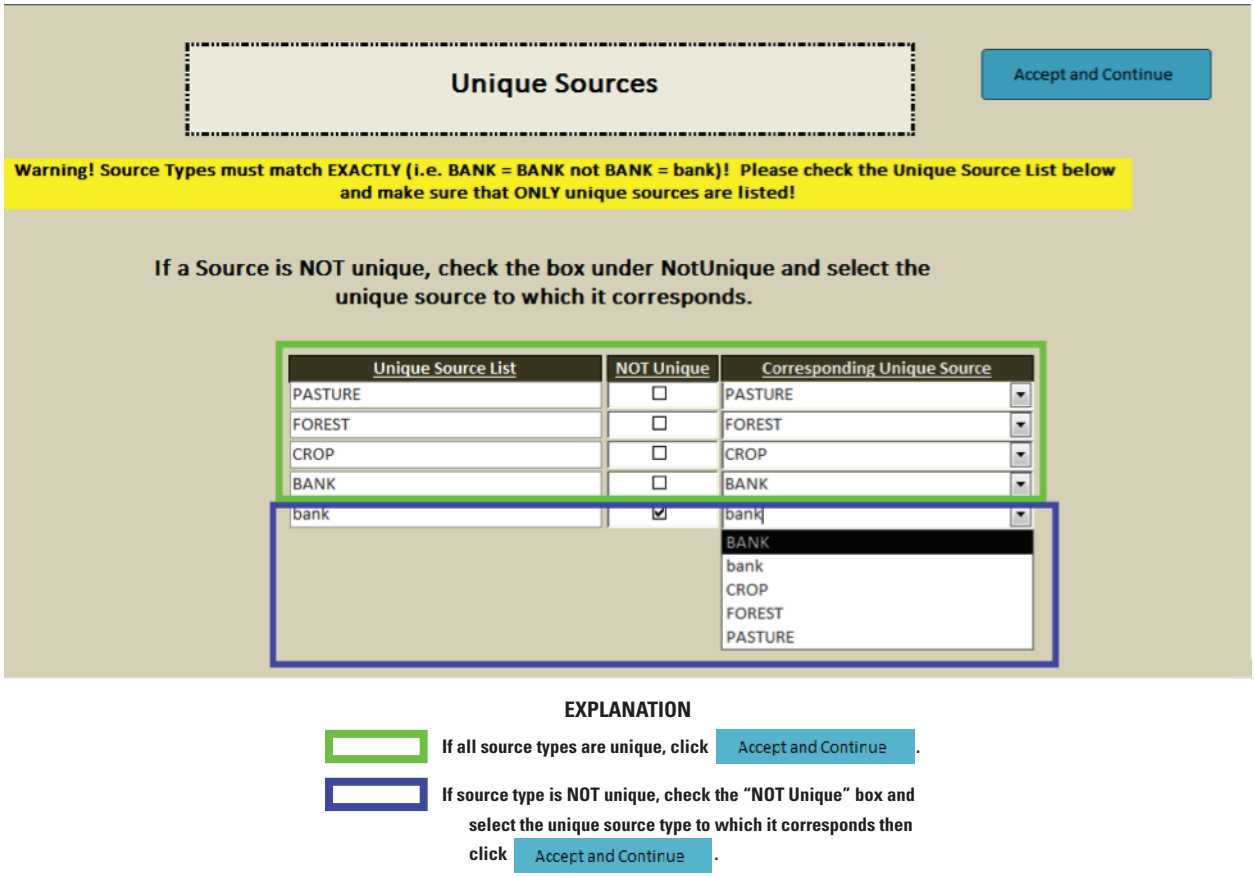


Figure 8–4. Unique Sources screen will appear on every run of the model.

Problem 5: Less Than Three Samples Per Source Group

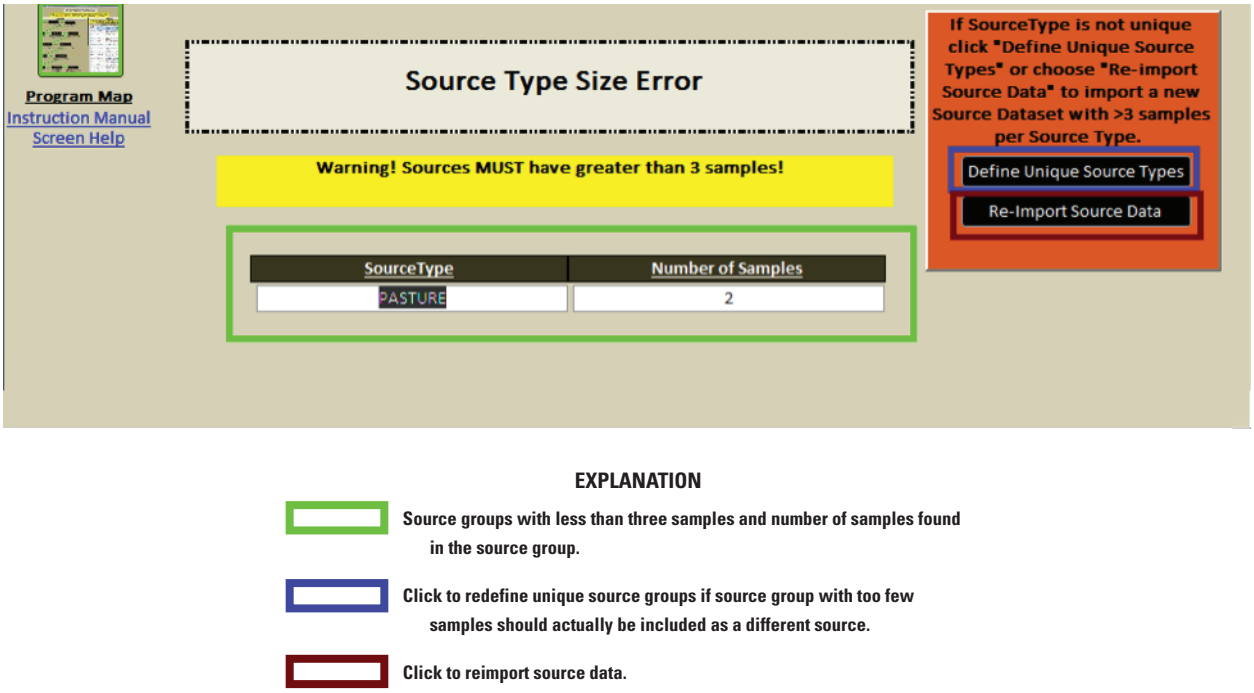


Figure 8–5. Sources with less than three samples found.

Problem 6: Number of Tracers < Number of Sources

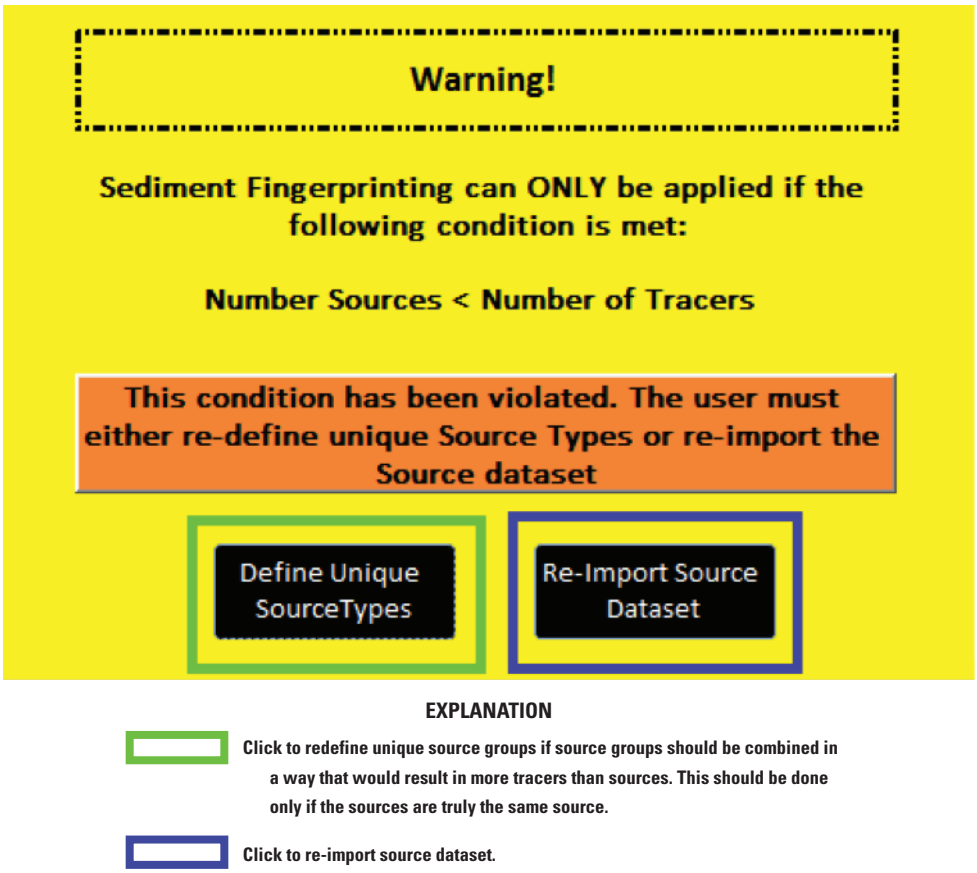


Figure 8–6 Fewer tracers than sources warning.

Problem 7: Missing */Null/* Blank Values

Nondetects in the source dataset are permitted.

Problem 8: Zeros and Negatives

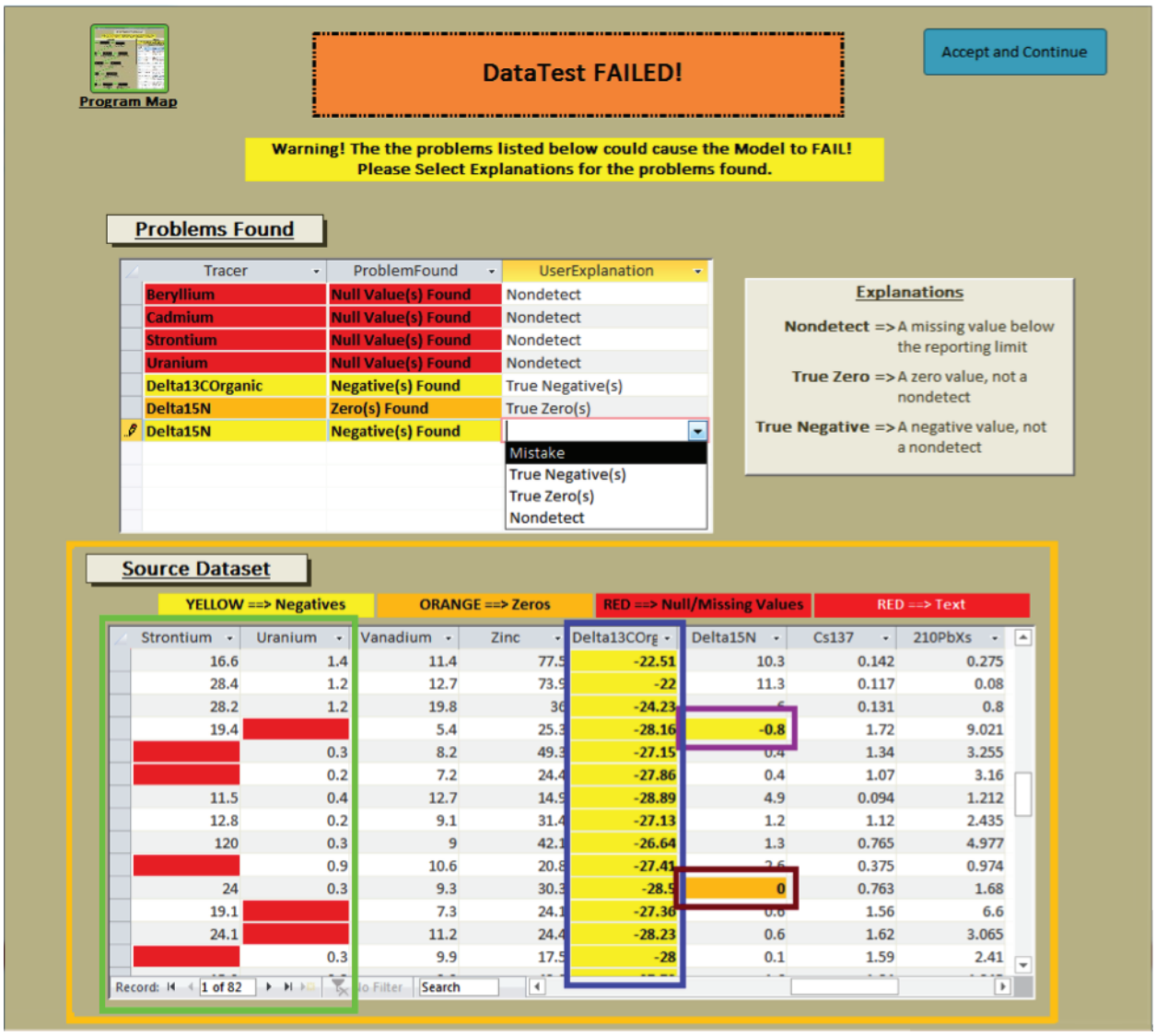
True negatives/zeros and nondetects are permitted.

Problem 9: Text in Tracer Data

Text that indicates nondetects is permitted (i.e. "<")

If any or all of Problems 7, 8, and 9 are found, the DataTest FAILED screen will appear. This does not indicate overall program failure, but the user must select appropriate explanations for any problems that were flagged in the *Data Testing Module*.

> ***Example Datasets Hint***
>
> Problems 7 and 8 appear in the example datasets. See the *Data Testing Module* for details on "User Explanation" selection.



EXPLANATION

Problem 7: Missing values found indicating nondetects. A nondetect is defined as a concentration value below the reporting limit (Helsel, 2012). Missing values that are NOT nondetects are not permitted. If missing values exist that are not nondetects, flag as a mistake and re-import the data.

Problem 8: Negatives. "True negatives" are values that are negative and are NOT nondetects.

Problem 8: Zeros. "True zeros" are values that are equal to zero and are NOT nondetects.

Problem 8: Negative values found NOT indicating a nondetect or a "True negative." If a mistake exists in the dataset, a new form will appear that will prompt the user to re-import the source dataset (See Mistakes in the Dataset and begin the Data Testing Module again.

The source dataset is shown highlighting all problems found during the Data Testing Module. If scrolling through the table, colors may not appear until the mouse is released.

Figure 8–7. Data Test FAILED screen will appear if Problems 7–9 have been found.

> ***Example Datasets Hint***
>
> Delta15N in the example dataset has both "true negatives" and "true zeros" and no mistakes.

Mistakes in the Dataset

If the user indicates that a mistake has been found in the dataset by selecting "Mistake" as the explanation for a problem found on the DataTest FAILED screen, the following screen will appear.

> ***Example Datasets Hint***
>
> No mistakes exist in the example datasets.

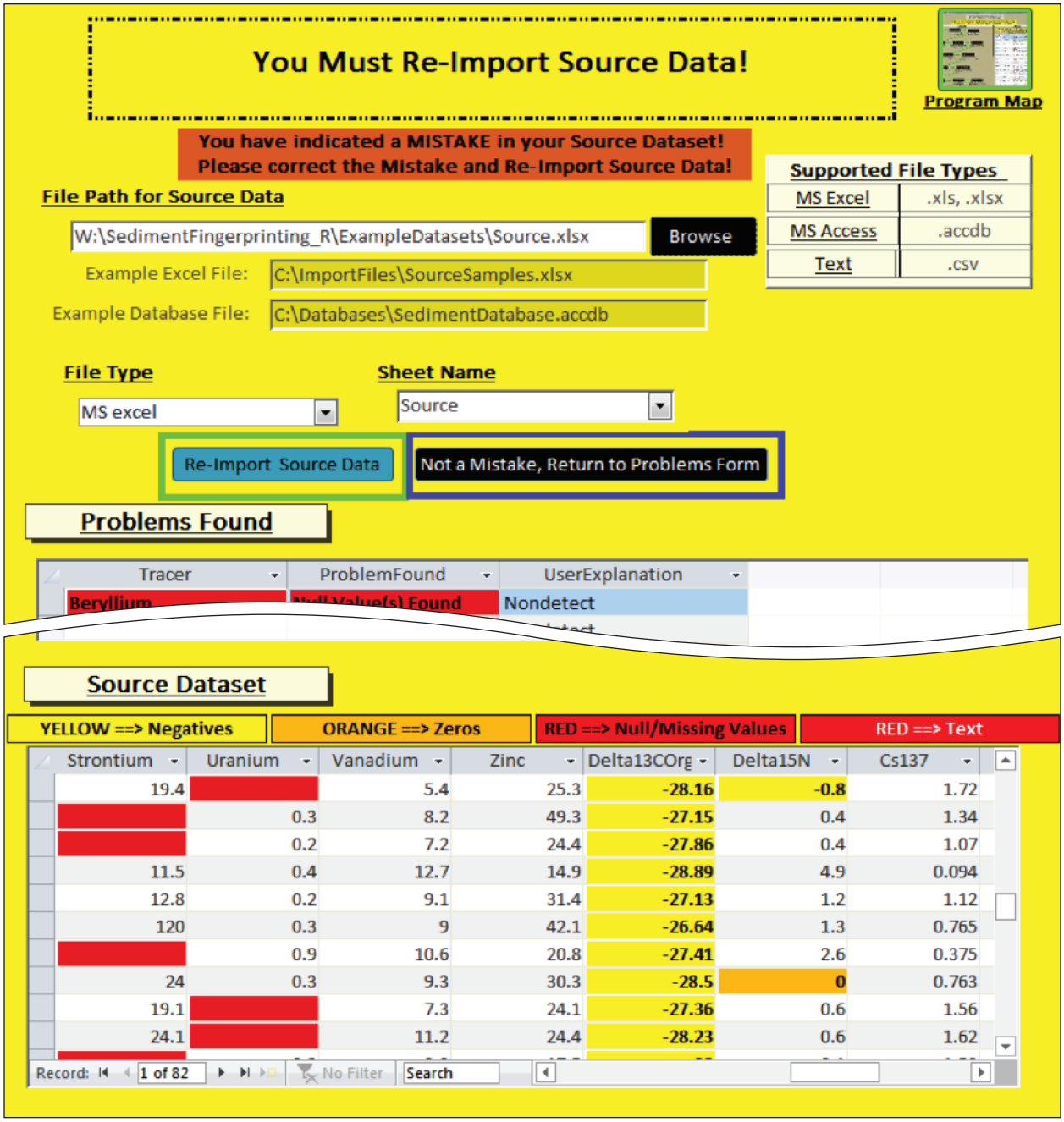
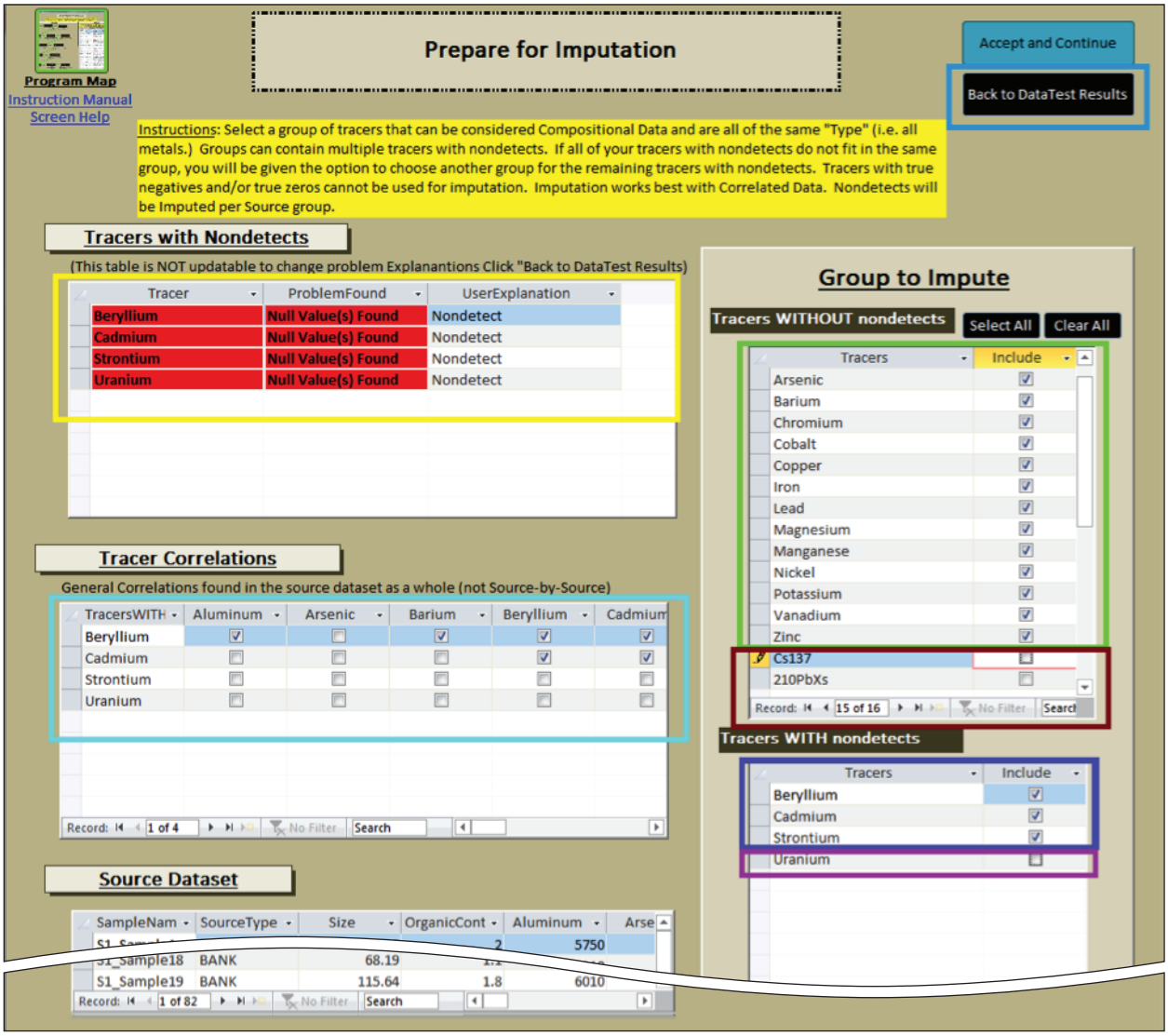


Figure 8–8. Re-import Data screen will appear if the user indicates a mistake in the imported dataset.

Appendix 9. Preparing for Imputation and Imputation Group Selection Screen

For more information on the imputation procedure, see *Imputation*.



EXPLANATION

Select tracers without nondetects to use in the current imputation group. Tracers without nondetects can be used in multiple imputation groups.

Select tracers WITH nondetects to impute. Tracers with nondetects can be used only in one imputation group, though a different group can be selected for each tracer with nondetects.

Do NOT select tracers that contain noncompositional data in any imputation group.

Not all tracers with nondetects have to be imputed at the same time. Leave the box unchecked to impute a tracer with a different imputation group.

Click to select/clear all tracers with AND without nondetects.

Data Testing Module results with user-selected explanation of "nondetect."

Table of correlations between tracers with nondetects and all tracers in the source dataset.

Click to return to Data Testing Module results (fig. 8–7), if errors have been made in selecting explanations for problems discovered.

Example Datasets Hint

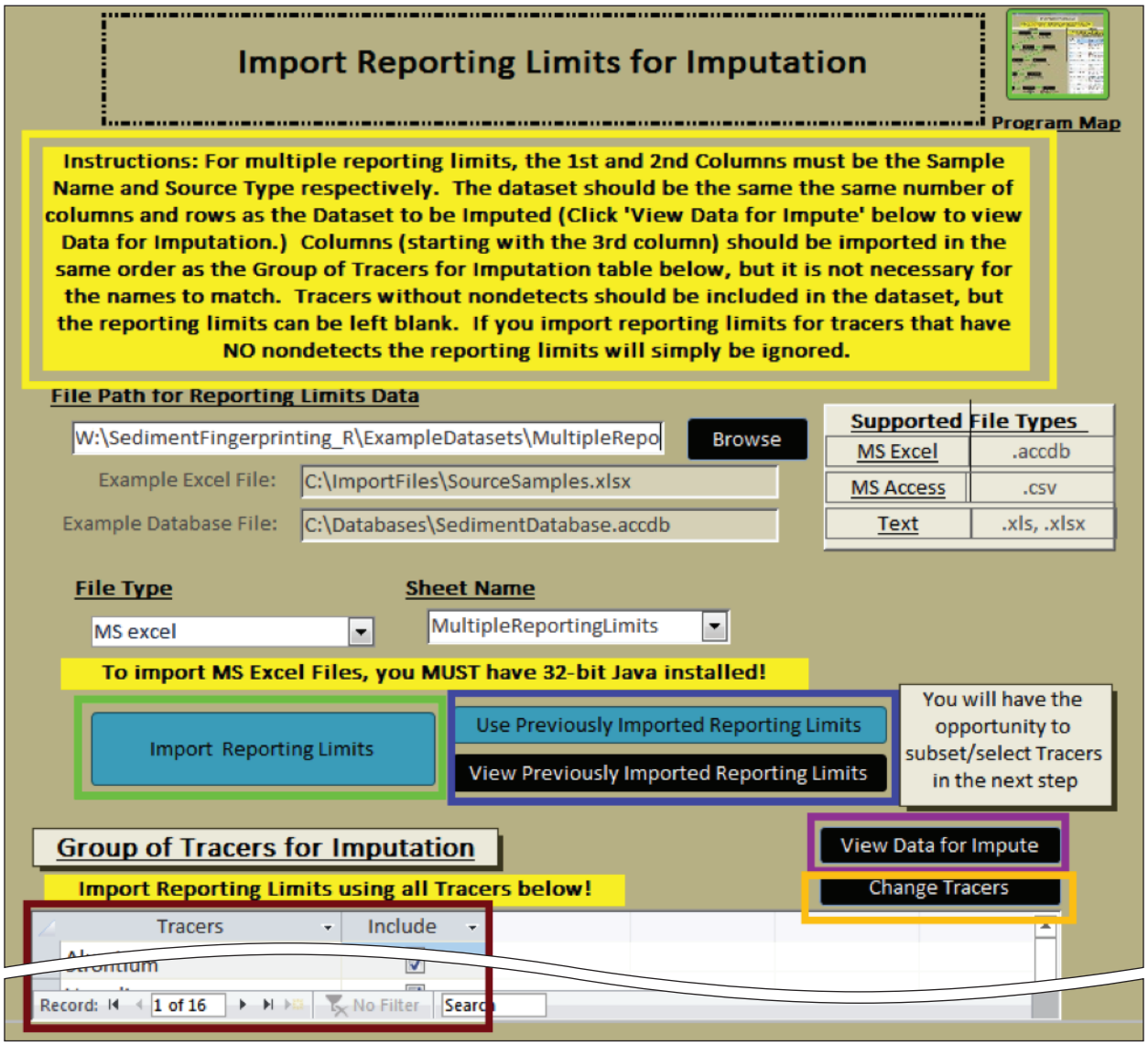
Select all tracers except radionuclides (Cs137 and Pb210.)

Figure 9–1. Prepare for Imputation screen.

Appendix 10. Reporting Limits Import Screens

For more details on reporting limits, see *What is a reporting limit?* and *Importing Reporting Limits*.

Import Reporting Limits



EXPLANATION

- Click to import new reporting limits.
- If a set of reporting limits for the current source dataset has previously been imported, click to view/use the current reporting limits set.
- Current imputation group.
- View the imputation group (the tracer data for selected tracers from the source dataset).
- Click to change the imputation group.
- Instructions (shown in yellow) for formatting multiple reporting limits. Instructions will change according to the type of reporting limits being imported (i.e. single or multiple).

Example Datasets Hint

See Single Reporting Limits and Multiple Reporting Limits for examples of how single and multiple reporting limit sets should be formatted.

Figure 10–1. Import Reporting Limits screen.

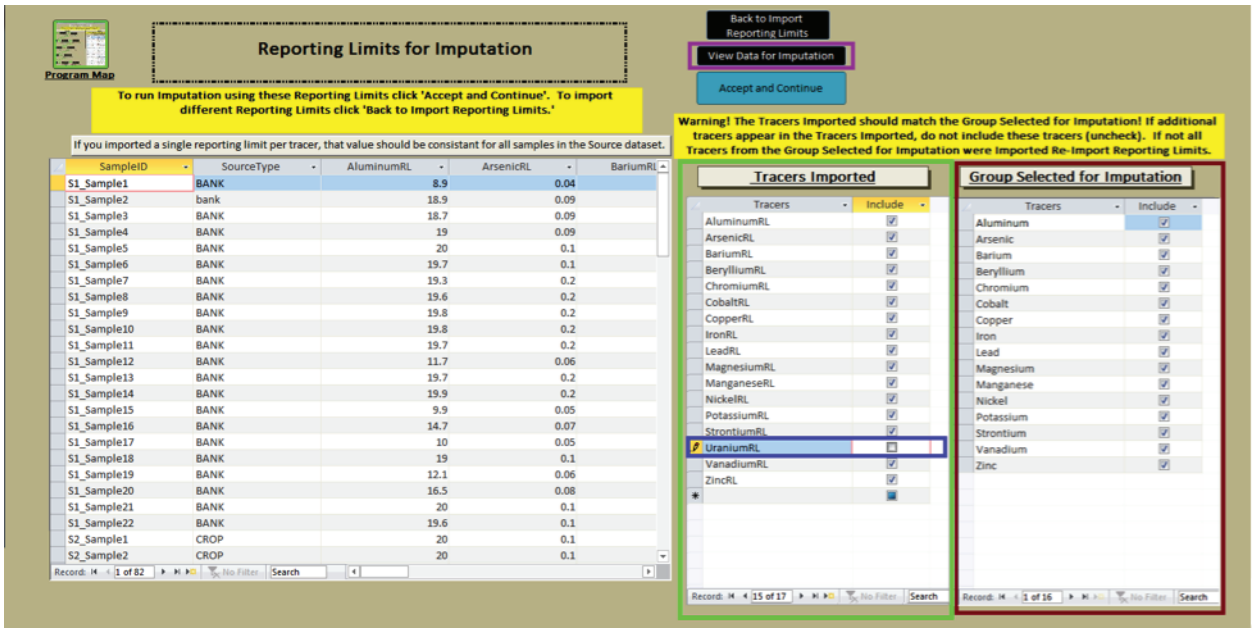


Figure 10–2. View and accept imported reporting limits.

Data Test of Reporting Limits

The reporting limits will go through a *Data Testing Module* to check for proper formatting. Below is an example of a problem found in the Data Testing Module for reporting limits, column mismatch between the source and reporting limit datasets.

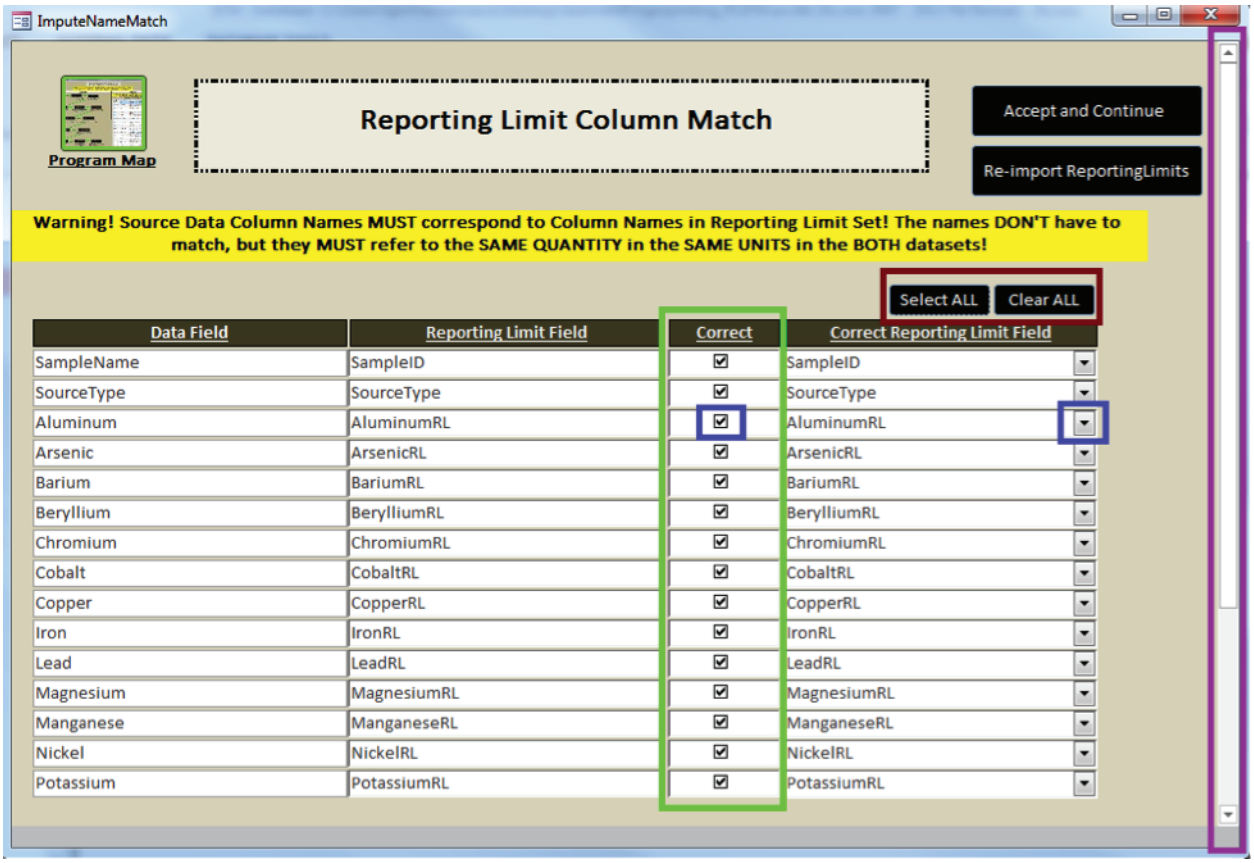


Figure 10–3. Reporting Limit Column Match screen will appear if column names in source data and reporting limits data are not identical.

> ***Example Datasets Hint***
>
> All columns are in the correct location in the reporting limits files in the example datasets.

Appendix 11. Choosing Imputation Parameters Screen

See *Imputation Parameters* for more details on the imputation procedure and parameter selection.

Single Imputation

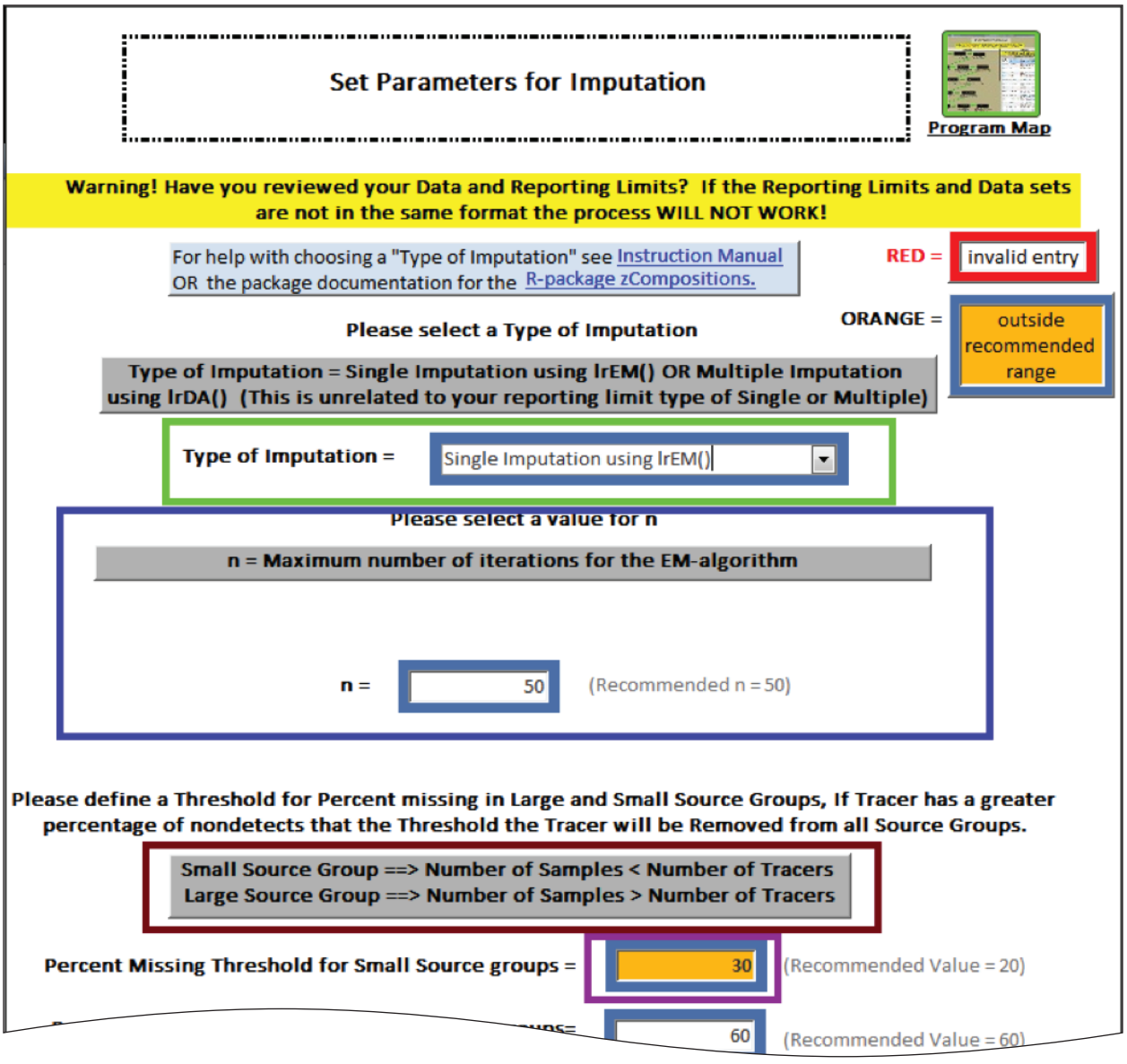


Figure 11–1. Imputation Parameters screen once the user has selected "Single Imputation."

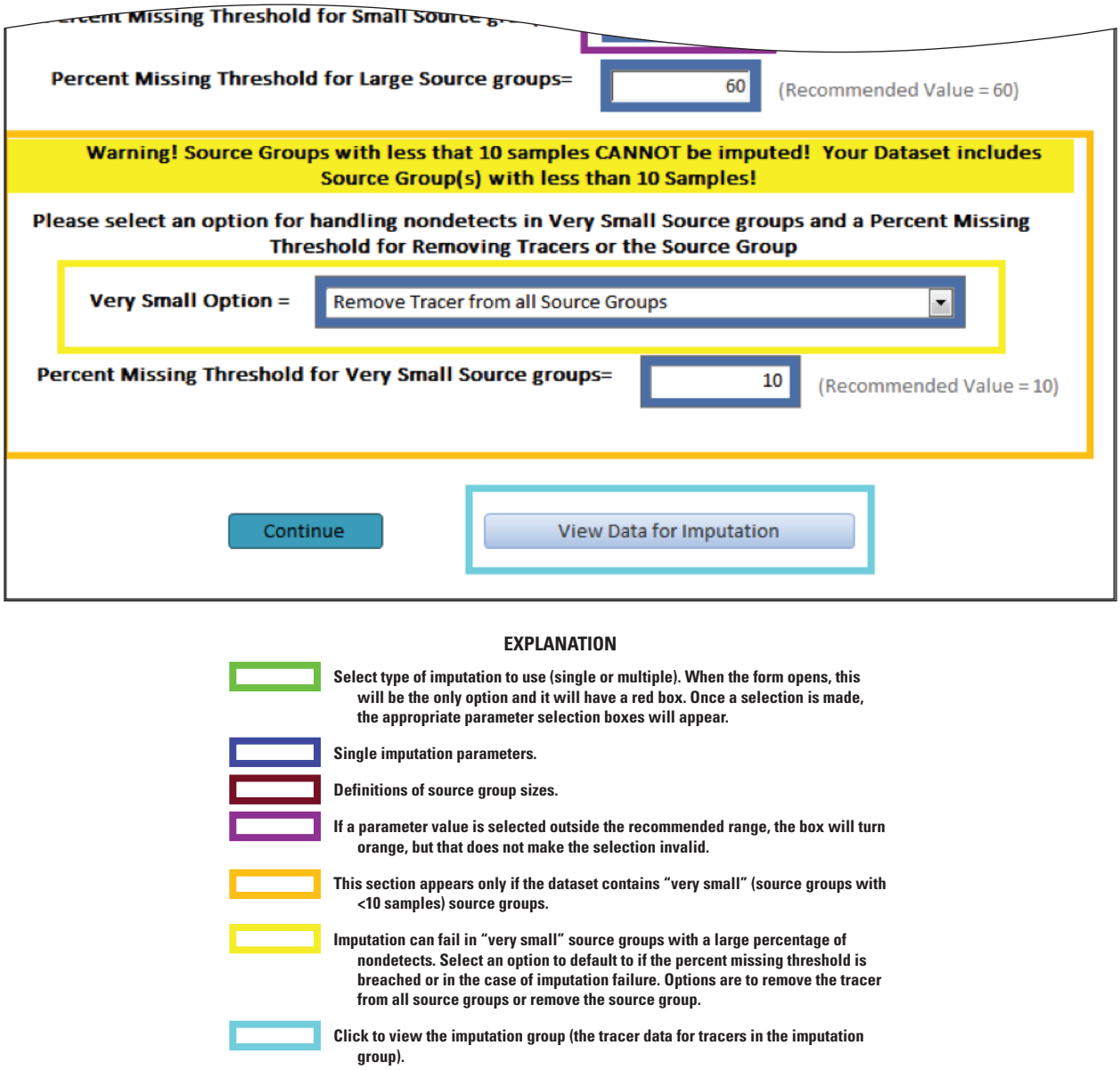


Figure 11–1. Imputation Parameters screen once the user has selected "Single Imputation."—Continued

Multiple Imputation

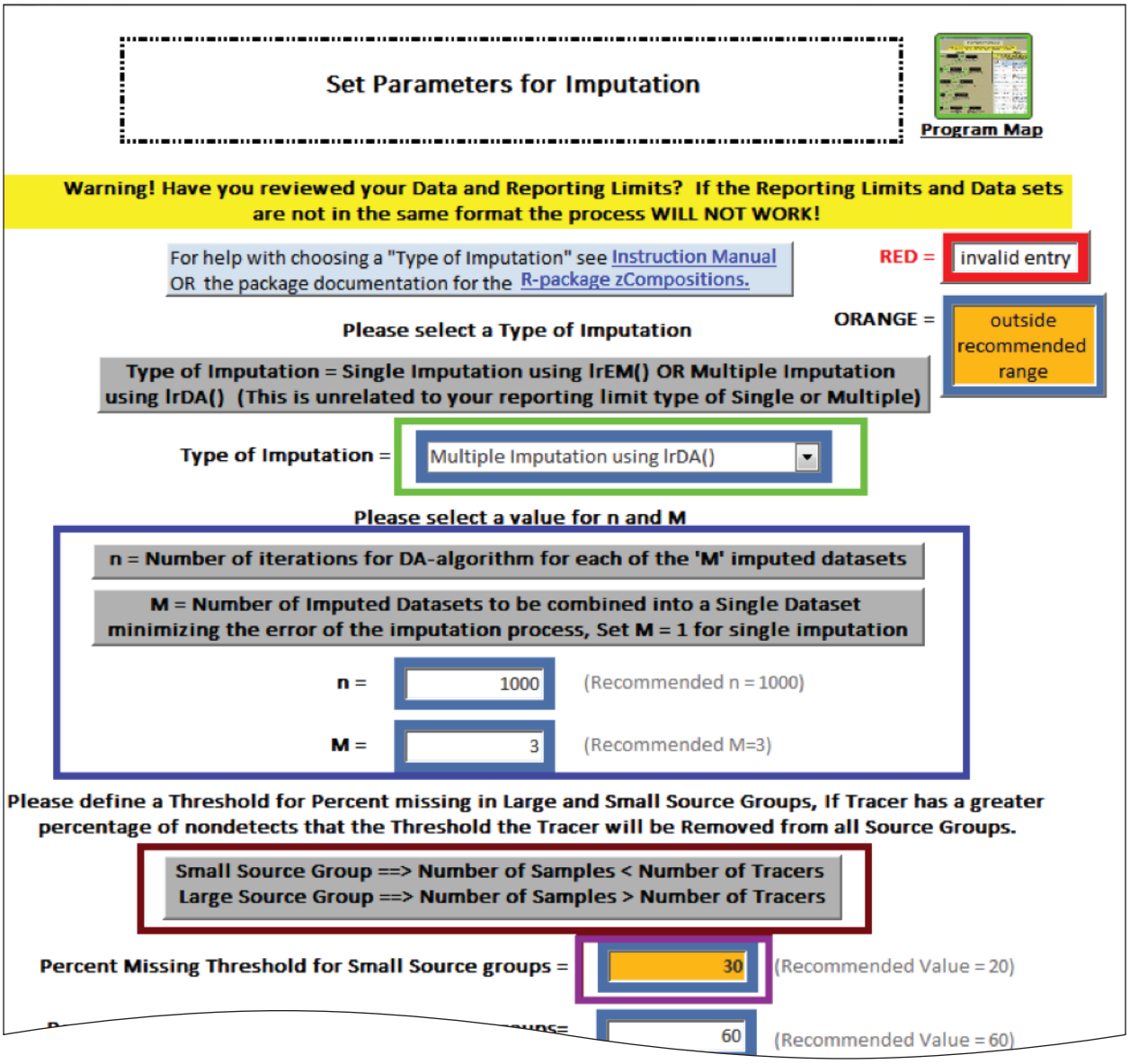
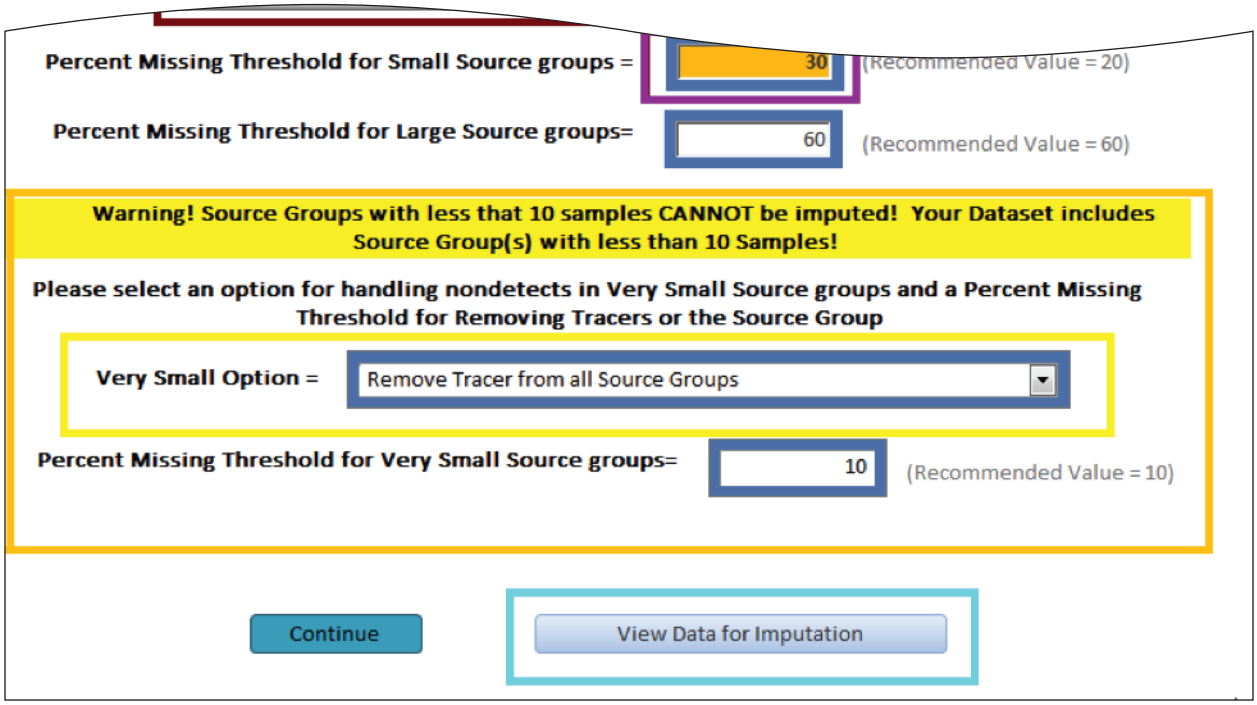


Figure 11–2. Imputation Parameters screen once the user has selected "Multiple Imputation."



EXPLANATION

- Select type of Imputation to use (single or multiple). When the form opens, this will be the only option and it will have a red box. Once a selection is made, the appropriate parameter selection boxes will appear.
- Multiple imputation parameters.
- Definitions of source group sizes.
- If a parameter value is selected outside the recommended range, the box will turn orange, but that does not make the selection invalid.
- This section appears only if the dataset contains "very small" (source groups with <10 samples) source groups.
- Imputation can fail in "very small" source groups with a large percentage of nondetects. Select an option to default to if the percent missing threshold is breached or in the case of imputation failure. Options are to remove the tracer from all source groups or remove the source group.
- Click to view the imputation group (the tracer data for tracers in the imputation group).

Figure 11–2. Imputation Parameters screen once the user has selected "Multiple Imputation."—Continued

Appendix 12. Imputation Results

See *Imputation* for details on the imputation procedure.

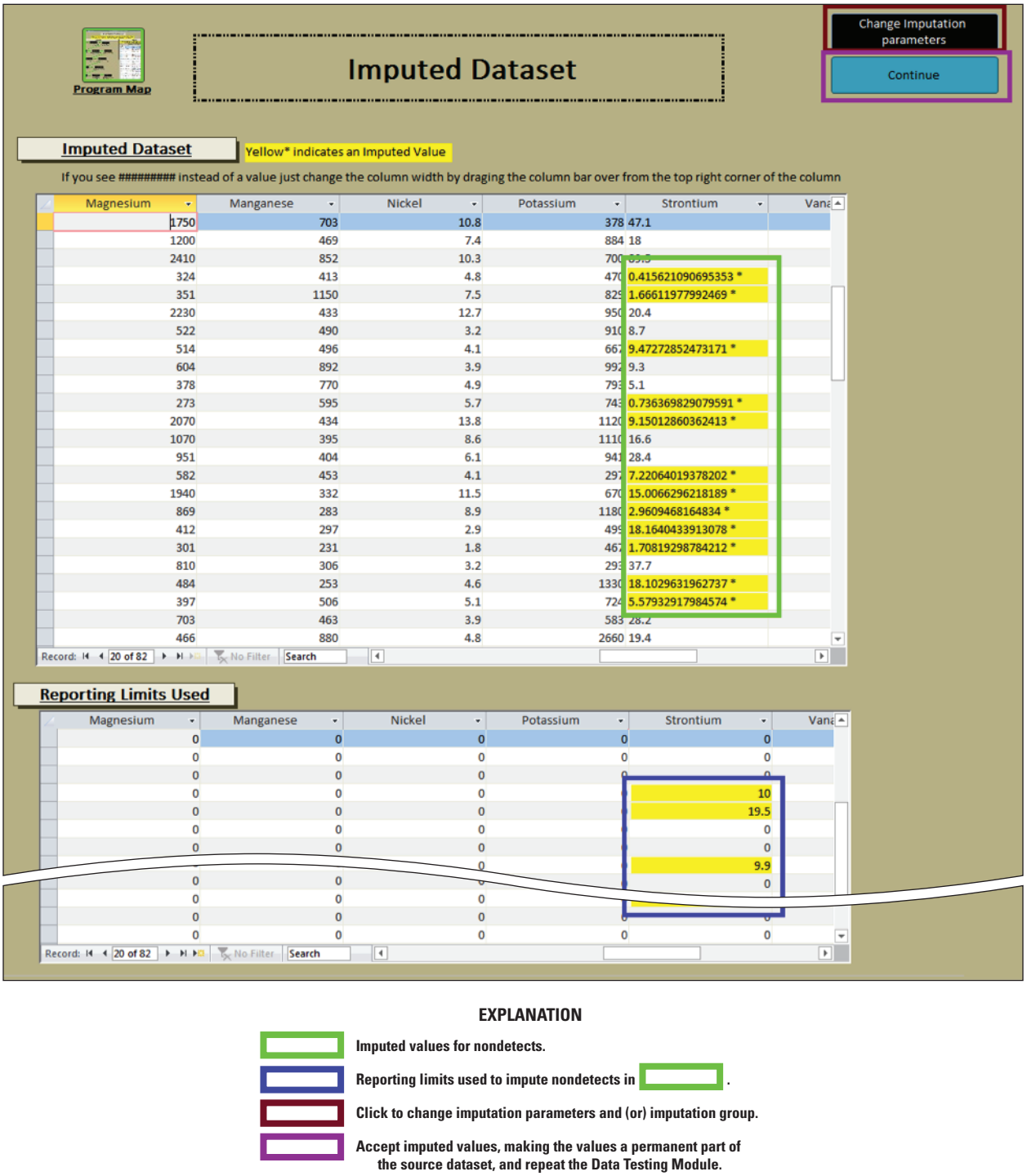


Figure 12–1. Imputation Results screen.

Appendix 13. Defining Functions to Shift True Negatives and/or True Zeros Into Positive Space

For more details on how negatives and zeros are managed throughout Sed_SAT, see *Negatives and True Zeros*.

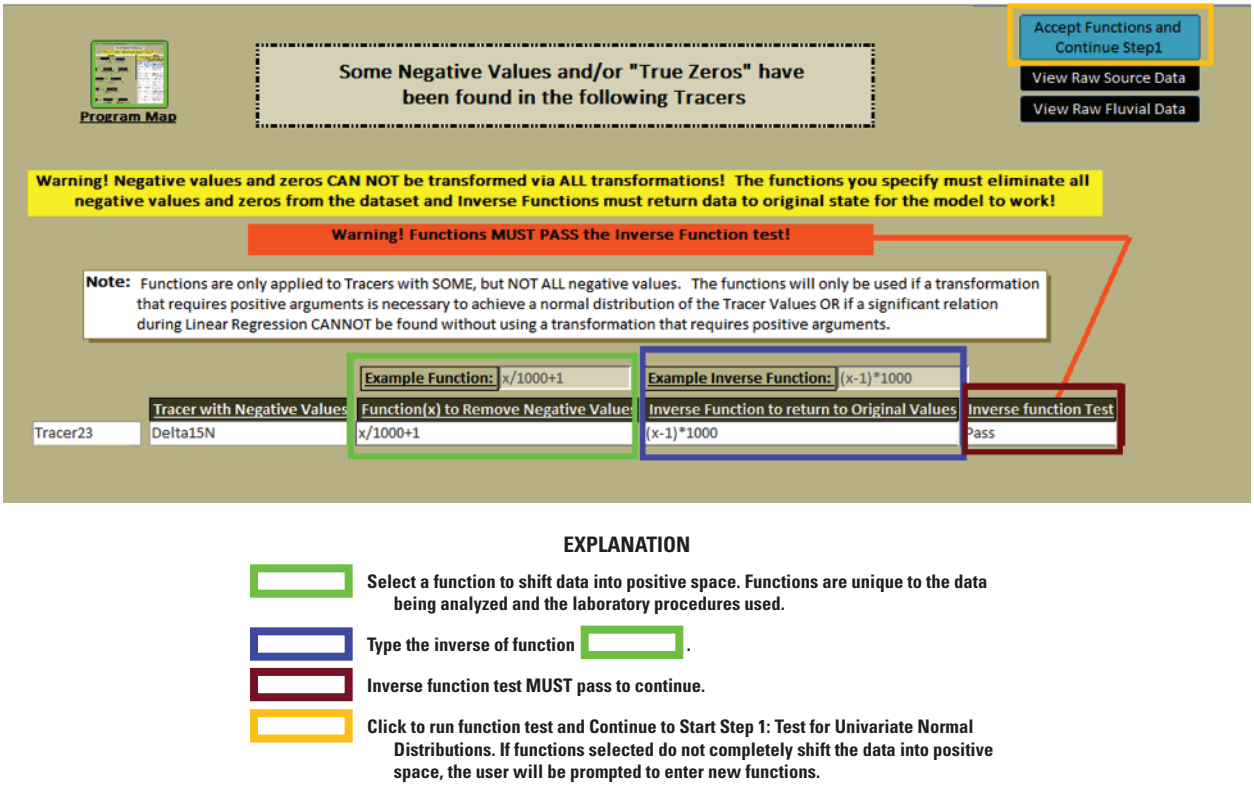


Figure 13–1. Negative Functions screen will appear if any tracer has data that is both positive and nonpositive.

> ***Example Datasets Hint***
>
> The “Example Functions” above the function input boxes can be used for the example datasets.

Appendix 14. Step 1 Outputs

See *Start Step 1: Test for Univariate Normal Distributions* for details on the Step 1 procedure.

Initial Step 1 Output

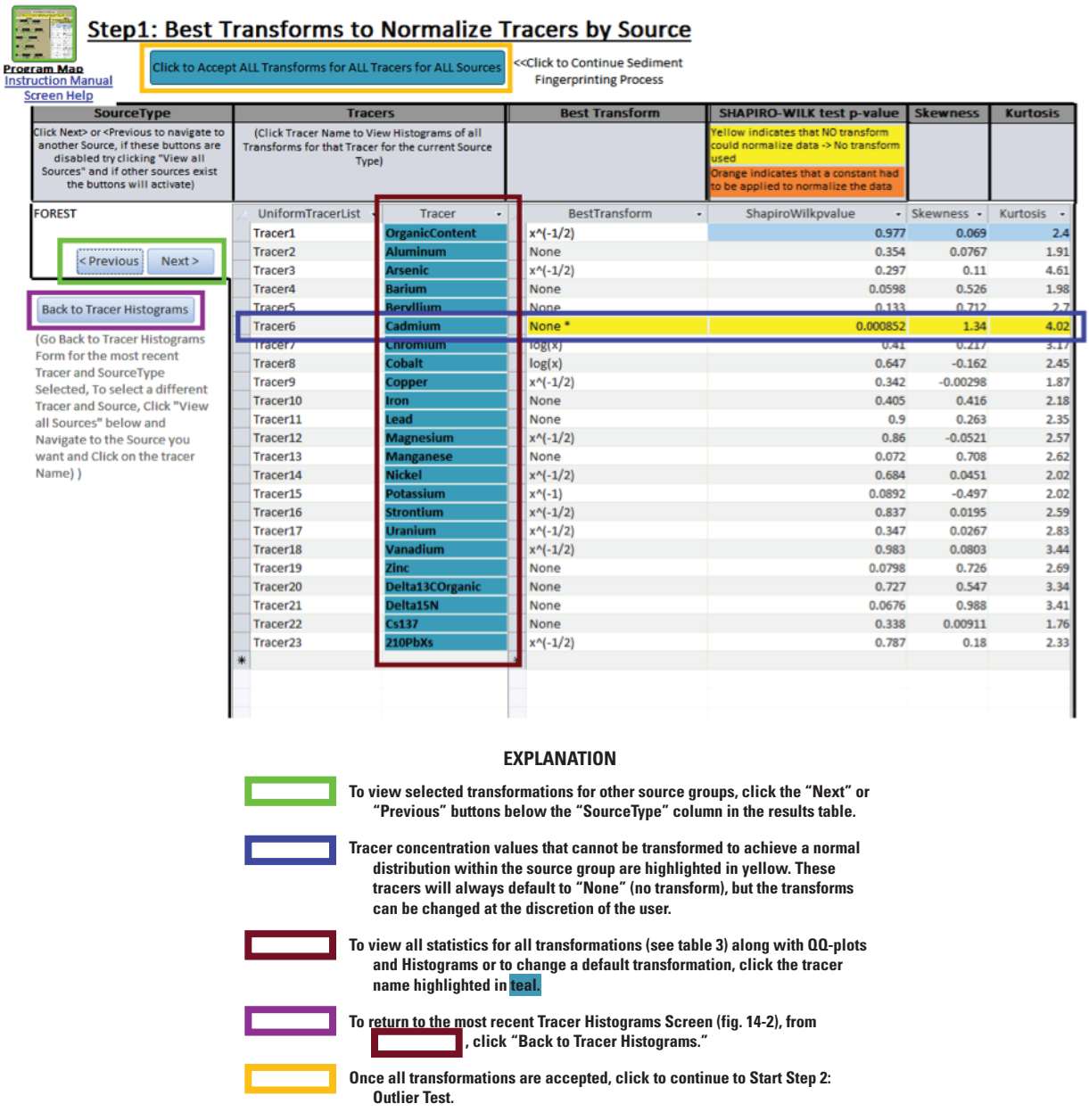
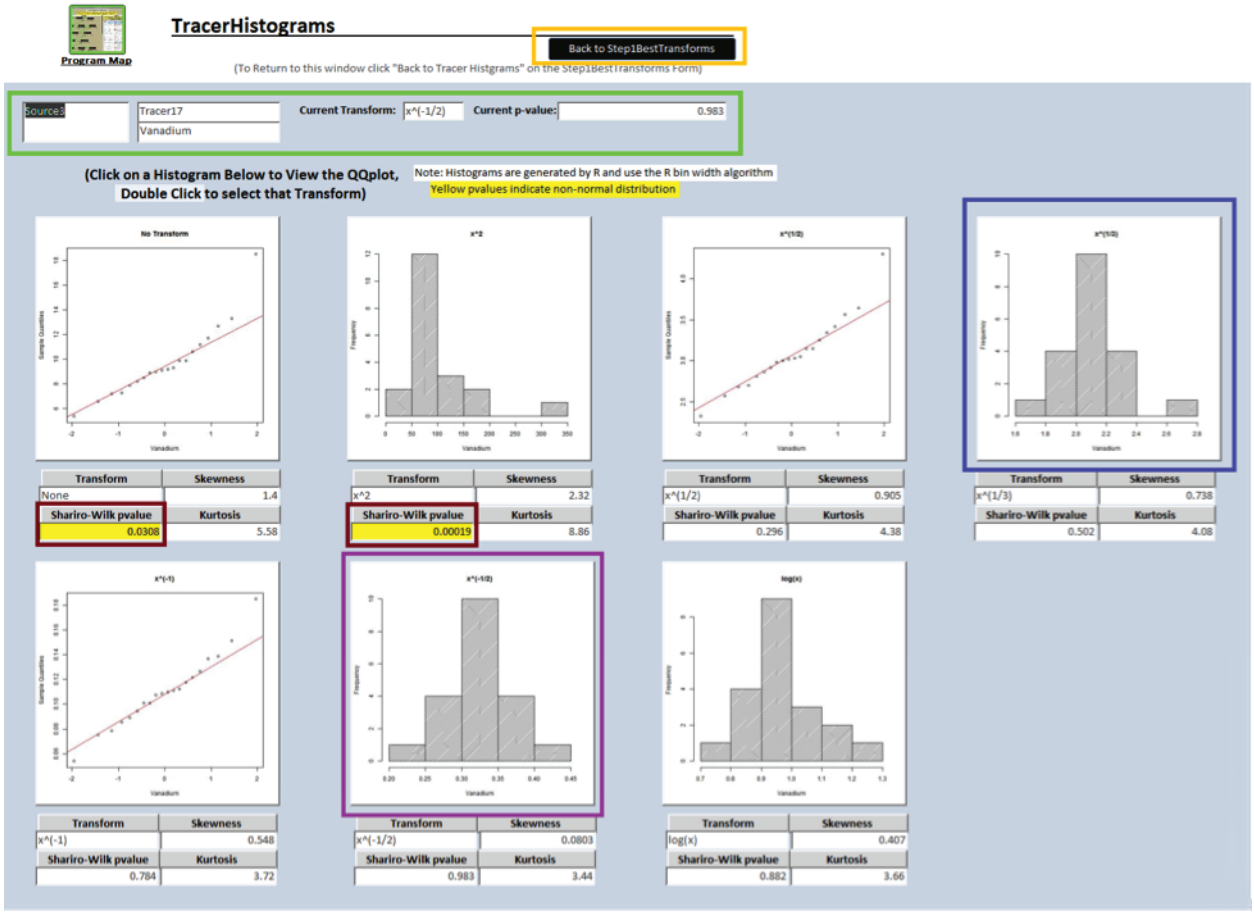


Figure 14–1. Step 1 Initial Output screen.

Changing the Default Transformation

If the user is unsatisfied with the transformation output for any tracer in any source group, the default transformation can be changed manually by the user. To view outputs for all possible transformations (table 3) for a given tracer in a given source group, click the tracer name highlighted in teal. This opens the screen below where the user is given the Shapiro-Wilk p-value, skewness, kurtosis, QQ-plot, and histogram for all possible transformations.



EXPLANATION

The current selected transformation is given at the top of the screen.

To view the histogram or QQ-plot if not displayed (flip to alternate plot), click the displayed plot once.

Transformations that have a Shapiro-Wilk p-value < 0.05 and are highlighted in yellow.

To select a transformation, double-click the plot of the desired transformation.

Click to return to the previous screen, Initial Step 1 Output.

Figure 14–2. Step 1 Change Transform screen.

Appendix 15. Step 2 Output

For details on the outlier test procedure, see *Start Step 2: Outlier Test*.

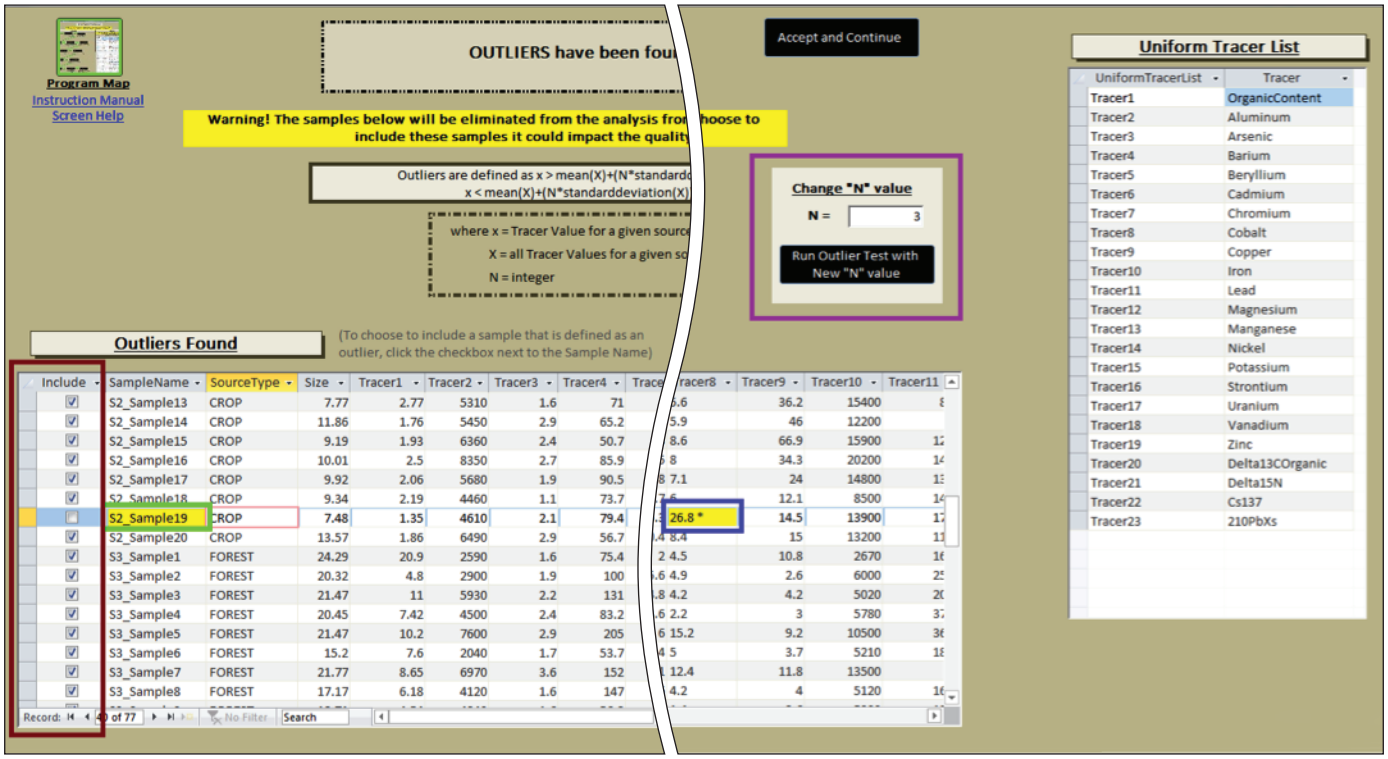


Figure 15–1. Step 2 Outliers Found (top half) screen will appear if any outliers are found in the source dataset.

> ***Example Datasets Hint***
>
> Assume that all outliers in the example datasets are due to measurement error and should NOT be included in the analysis.

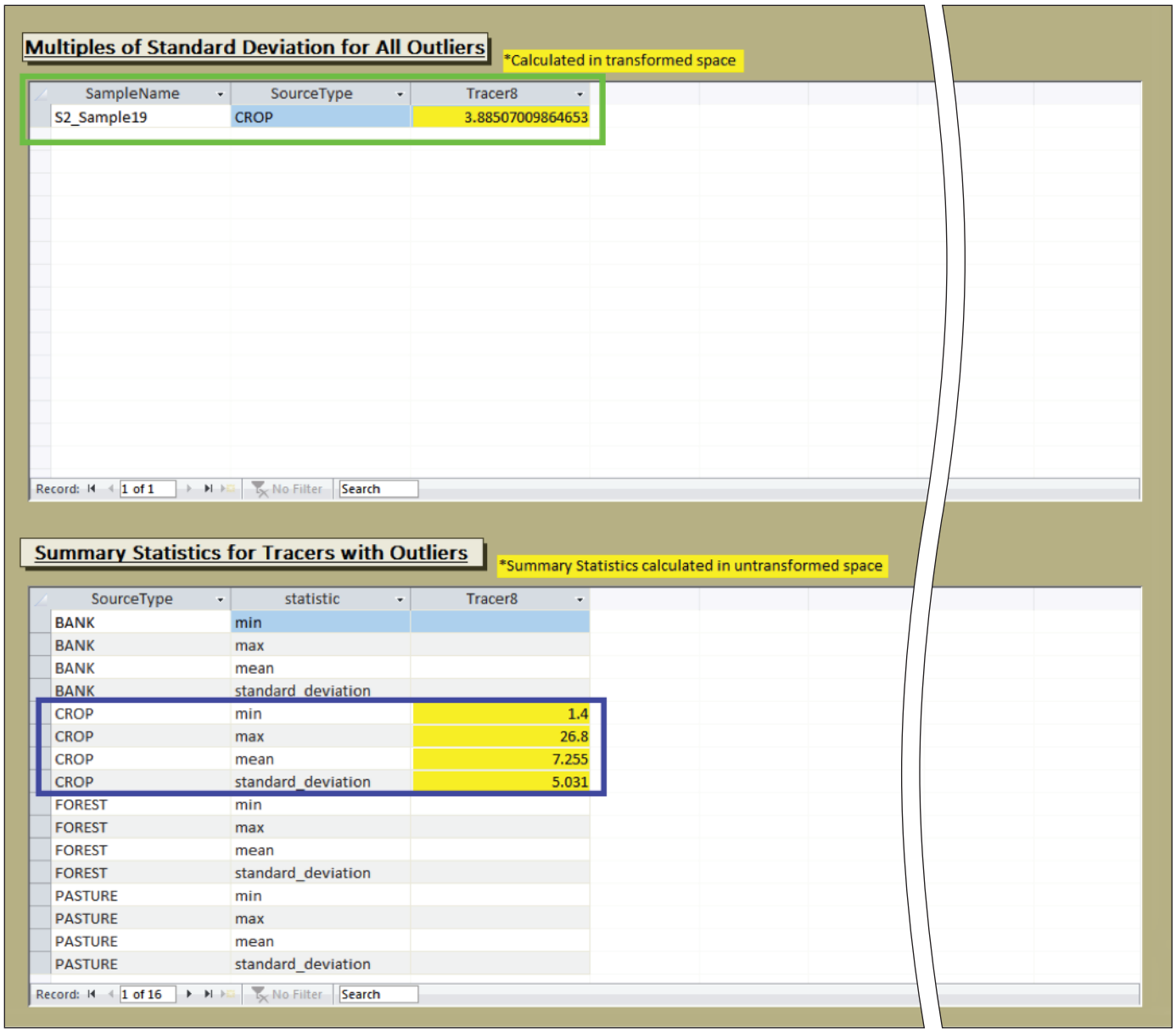
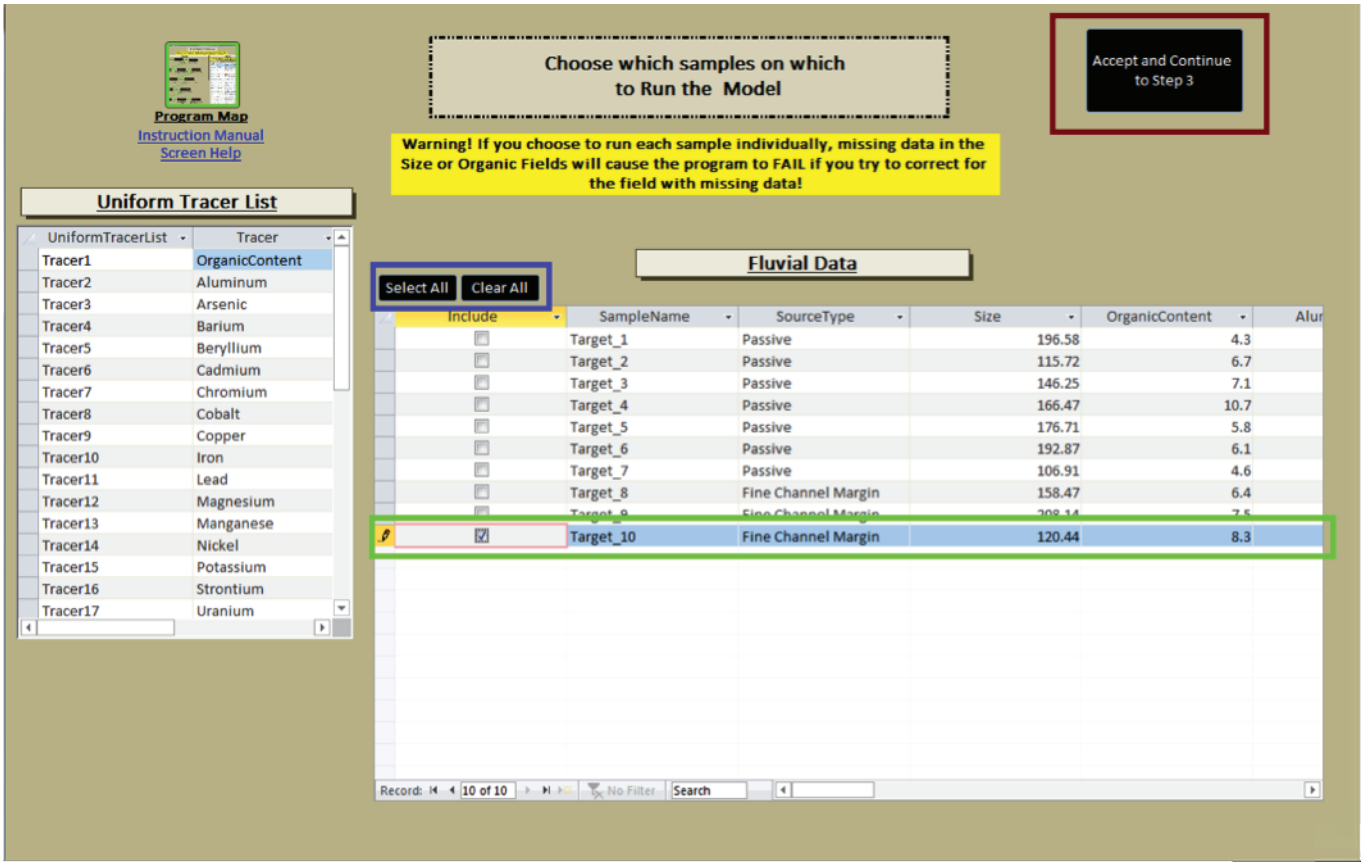


Figure 15–2. Step 2 Outliers Found (bottom half) screen will appear if any outliers are found in the source dataset.

Appendix 16. Selecting Target Samples to Analyze

See *Selecting Target Samples* for details on target sample selection.



EXPLANATION

Only target samples with the "Include" checkbox checked will be used from this point forward. To run a different target sample(s), return to this screen by clicking Start Step 3 on the Program MAP (fig. 7). Samples with missing size data cannot be run individually in Start Step 3: First Linear Regression since there are no size data present with which to calculate the correction factor in equation 2.

To select all or clear all target samples, click the "Select All" and "Clear All" buttons respectively.

Click to proceed with Start Step 3: First Linear Regression for the target samples selected

Figure 16–1. Choose Target Samples screen will appear after Step 2.

Appendix 17. Step 3 Output

For details on the linear regression procedure and application of correction factors, see *Start Step 3: First Linear Regression*.

Initial Step 3 Output

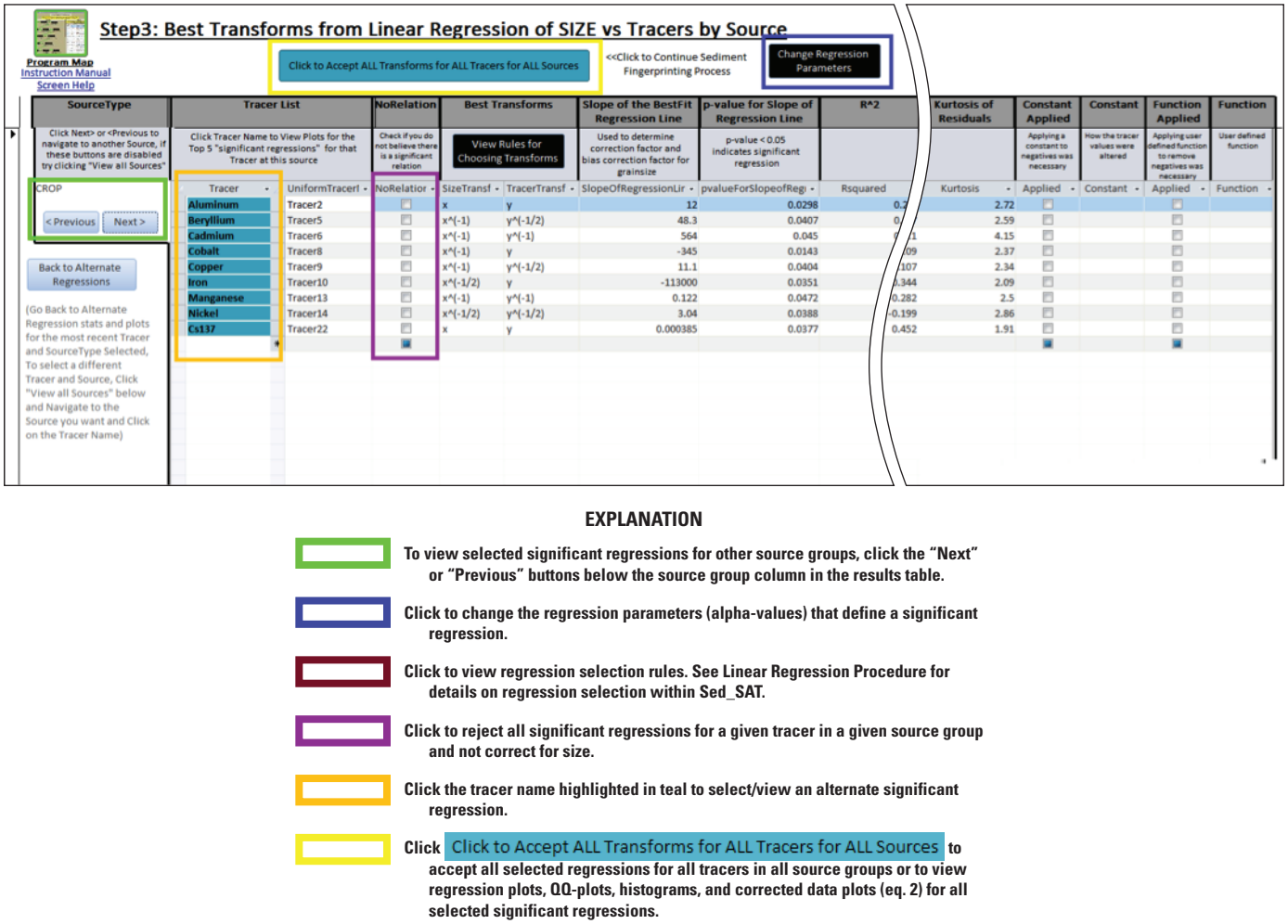
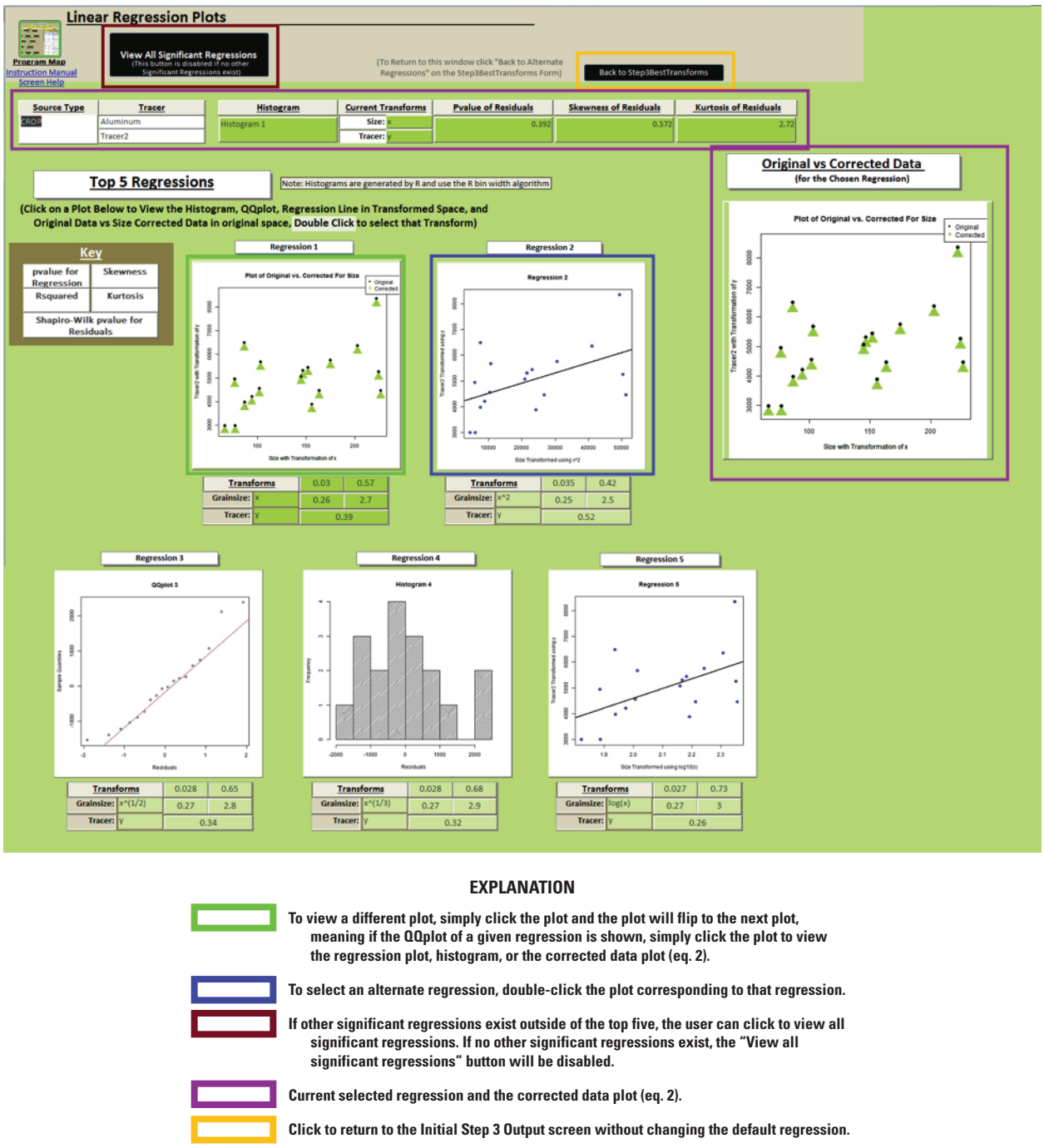


Figure 17–1. Step 3 Initial Output screen.

Changing/Rejecting Chosen Regression

The user has the option to override the regression rules from *Linear Regression Procedure* and choose an alternate regression or choose not to correct a tracer for size at all. To view alternate regressions and (or) QQ-plots of residuals, histograms, regression plots, and corrected data plots (eq. 2) for the current regression, click the tracer name highlighted in teal on the *Initial Step 3 Output* screen.

The top five regressions according to the rules in Linear Regression Procedure are shown on the first screen after clicking the tracer name on the Initial Step 3 Output screen with their associated plots and statistics.



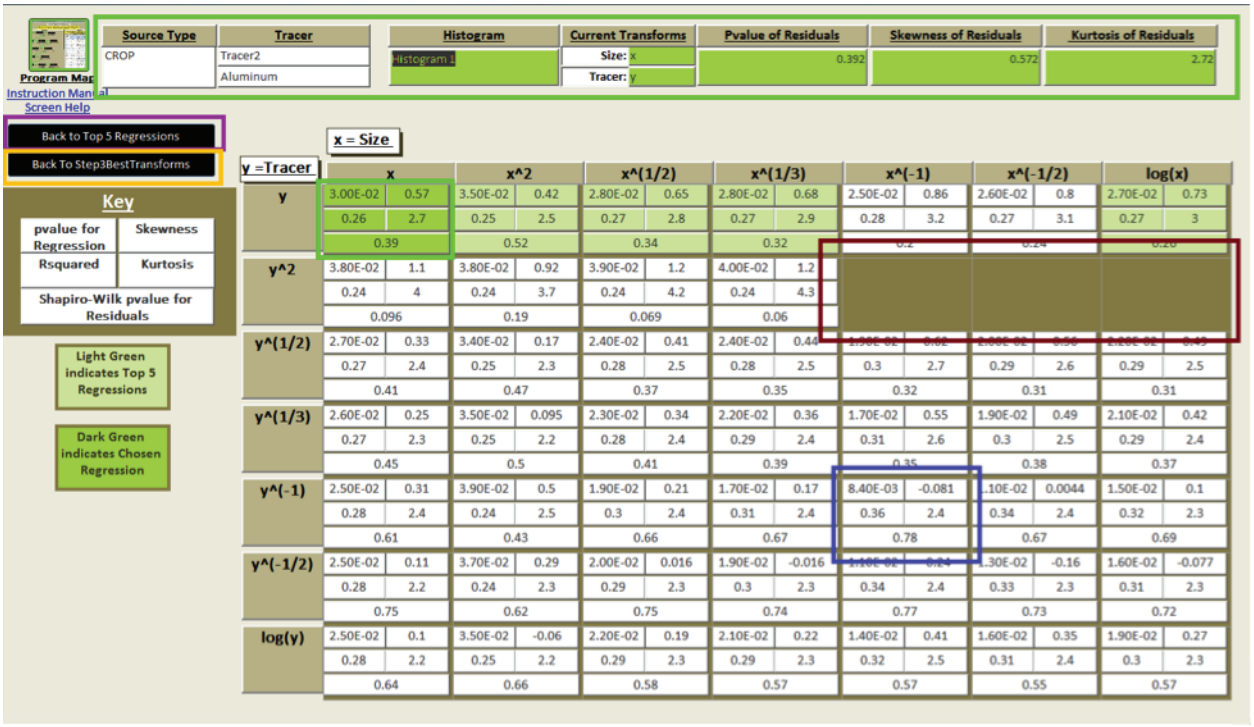
EXPLANATION

- **To view a different plot, simply click the plot and the plot will flip to the next plot, meaning if the QQplot of a given regression is shown, simply click the plot to view the regression plot, histogram, or the corrected data plot (eq. 2).**
- **To select an alternate regression, double-click the plot corresponding to that regression.**
- **If other significant regressions exist outside of the top five, the user can click to view all significant regressions. If no other significant regressions exist, the "View all significant regressions" button will be disabled.**
- **Current selected regression and the corrected data plot (eq. 2).**
- **Click to return to the Initial Step 3 Output screen without changing the default regression.**

Figure 17–2. Step 3 Initial Change Regression screen displays the top 5 regressions.

View All Significant Regressions

After clicking View All Significant Regressions (This button is disabled if no other Significant Regressions exist) on the *Changing/Rejecting Chosen Regression* screen shown above, the user can choose from a matrix of all significant regressions for the selected tracer in the selected source group. The current selected regression is highlighted in dark green, and the top five regressions according to the hierarchy detailed in the *Linear Regression Procedure* are highlighted in light green. Numbers shown in the matrix conform to the statistics listed in key on the left-hand side of the screen.



EXPLANATION

Current selected regression (dark green in matrix).

To view all output plots for a given regression, click the cell in the matrix that corresponds to that regression and then click the plot to flip through the plots. To select a regression, click Choose this Regression **after clicking the cell corresponding to the regression.**

Blank cells in the matrix correspond to regressions that failed the significance test according to the user selected regression parameters, and those regressions cannot be selected.

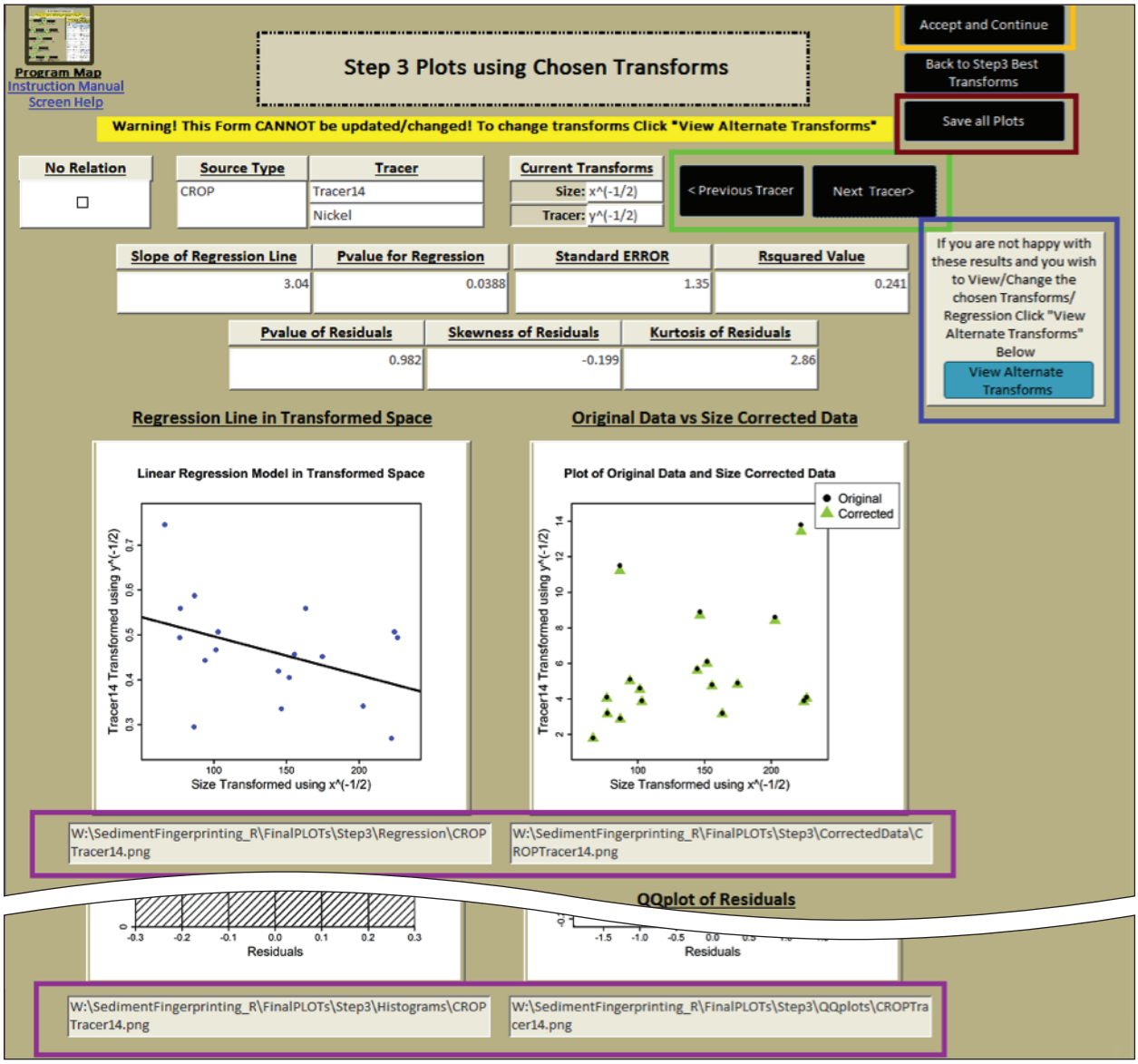
Click to go back to the previous screen, Changing/Rejecting Chosen Regression.

Click to go back to the Initial Step 3 Output screen.

Figure 17–3. Step 3 Choose Regression screen displays all significant regressions.

Step 3 Plots

After all regressions have been selected, the user can review the statistics and all plots for all of the selected regressions (either by default or user selection) before applying size-correction factors to the source dataset. It is strongly recommended that the user take the time to review each regression before proceeding to the next step.



EXPLANATION

- Click to view another selected regression in another tracer and (or) another source group.
- Click to change/view other significant regressions for a given tracer in a given source group.
- Click to save all plots for all tracers in an alternate location. Plots can also be saved at the end of the analysis in the Export Data/Tables/Plots step.
- The current locations of the plots are given below the plot.
- Click to apply size-correction factors (eq. 2) to tracers and continue to Start Step 4: Second Linear Regression for Organic Content.

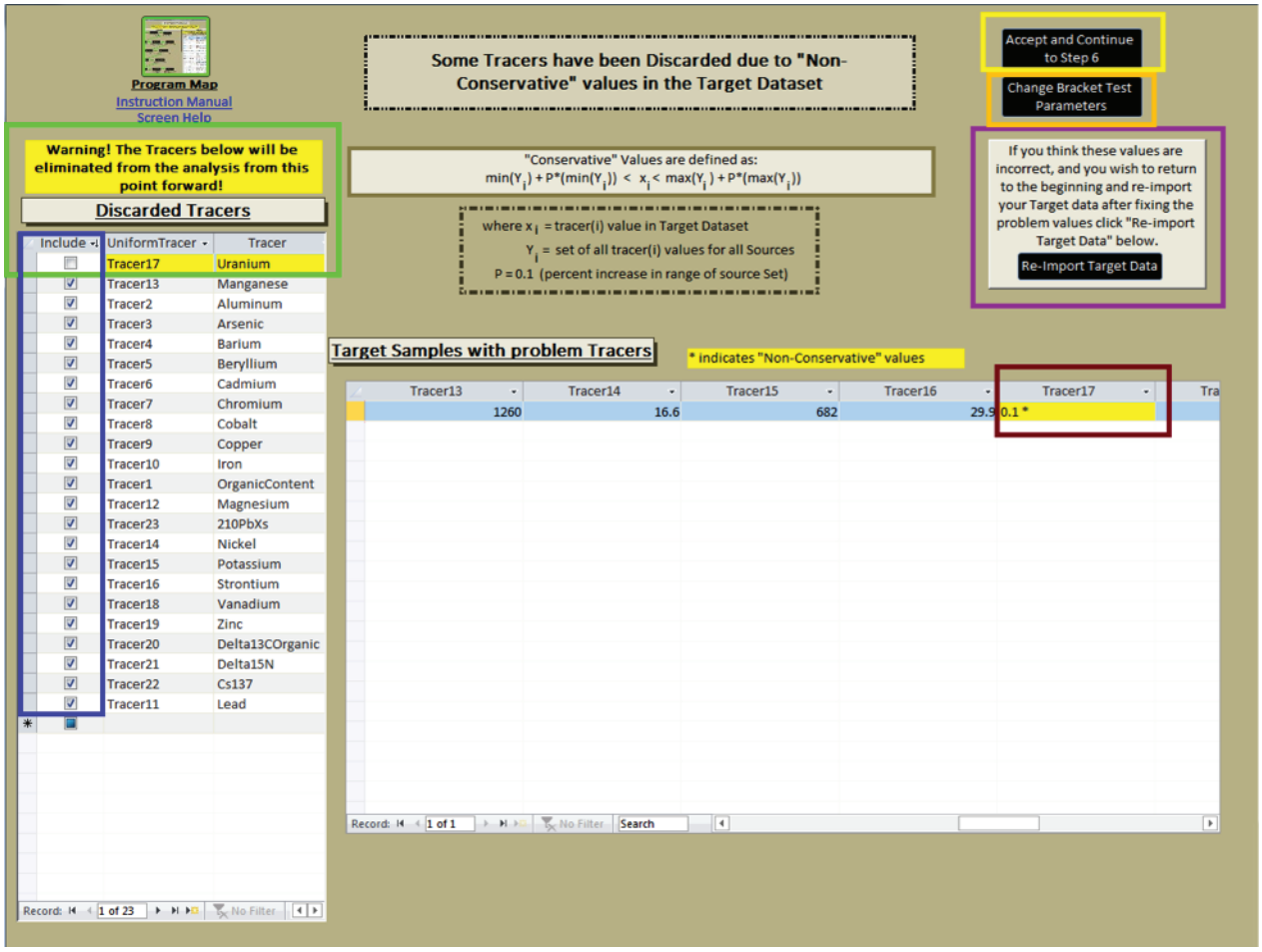
Figure 17–4. Step 3 Plots screen appears once all regressions have been accepted by the user on the *Initial Step 3 Output* screen.

Appendix 18. Step 4 Output

Step 4 output has the same structure as Step 3 output in *appendix 17*; simply replace “size” with “organic content” in descriptions.

Appendix 19. Step 5 Output

For details on the bracket test procedure, see *Start Step 5: Bracket Test*.



EXPLANATION

Nonconservative tracers are highlighted in yellow and by default are not included in the analysis from this point forward.

To include a nonconservative tracer, click the checkbox in the "Include" column.

Target samples are shown in the center table with nonconservative values flagged with "*" and highlighted in yellow.

If a flagged value appears in the target dataset that is incorrect or a mistake, return to the beginning and re-import the target dataset.

Click to change the range of the "bracket," enter a new value for "*p*" from equation 3.

Click to remove selected nonconservative tracers and proceed to Start Step 6: Multivariate Normality Test.

Figure 19–1. Step 5 Bracket Test Results screen will appear if any tracers fail the bracket test.

> ***Example Datasets Hint***
>
> Only "Target_10" will have any tracers that fail the bracket test assuming that all steps leading up to the bracket test were run using defaults. All other target samples will pass the bracket test for all tracers. Target samples are never removed in the bracket test, only tracers.

Appendix 20. Step 6 Output

The five multivariate normality tests applied to the source dataset and the boxCox() (Lorenz, 2015b)-transformed source dataset are listed in table 7. Each test receives a pass-fail score of 1 for pass and 0 for fail. The "MVN score" is the sum of the pass-fail scores for each multivariate normality test. For more details on the multivariate normality and homoscedasticity testing procedure, see *Start Step 6: Multivariate Normality Test*.

Step 6 Output

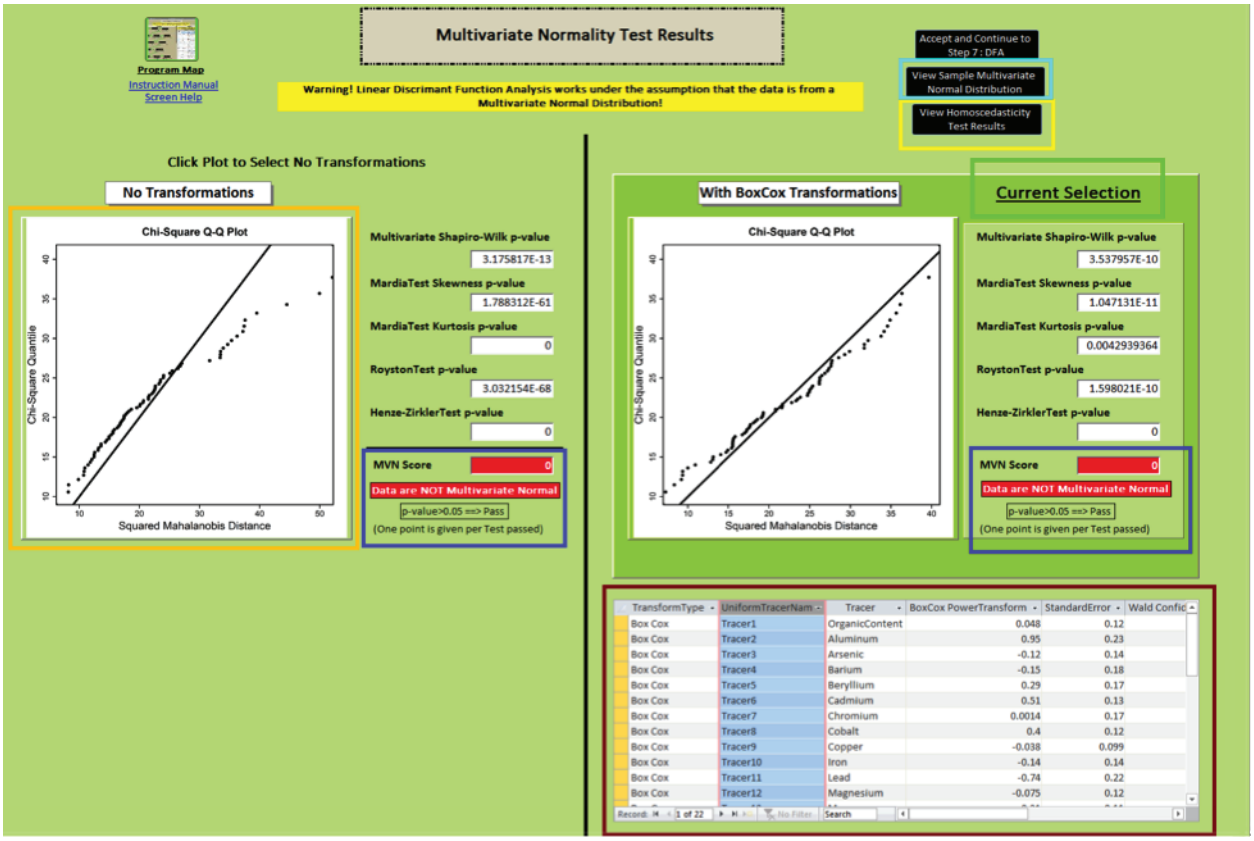


Figure 20–1. Step 6 Initial Output screen.

Homoscedasticity Results Screen

The betadisper() function from the vegan R-package (Oksanen and others, 2016) applies a multivariate version of the Levene test for homogeneity of variances (Levene, 1960).

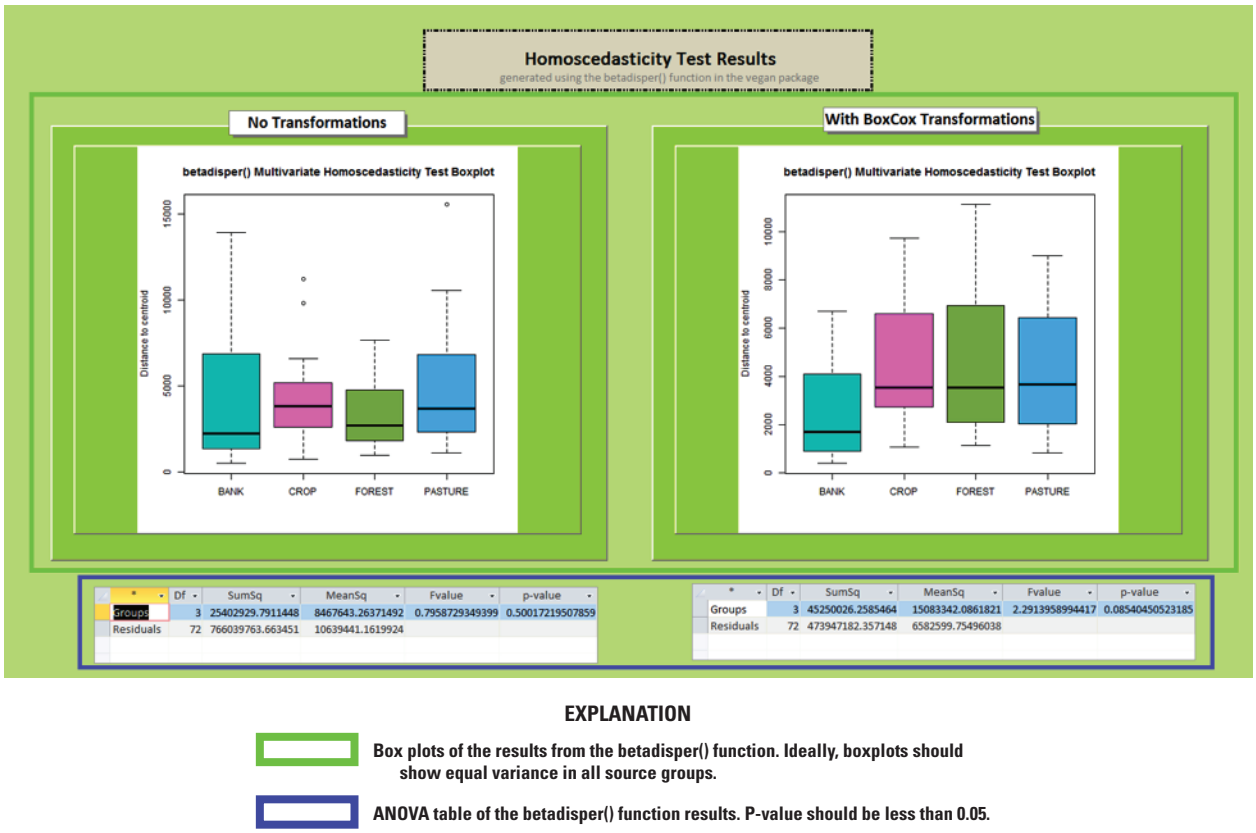
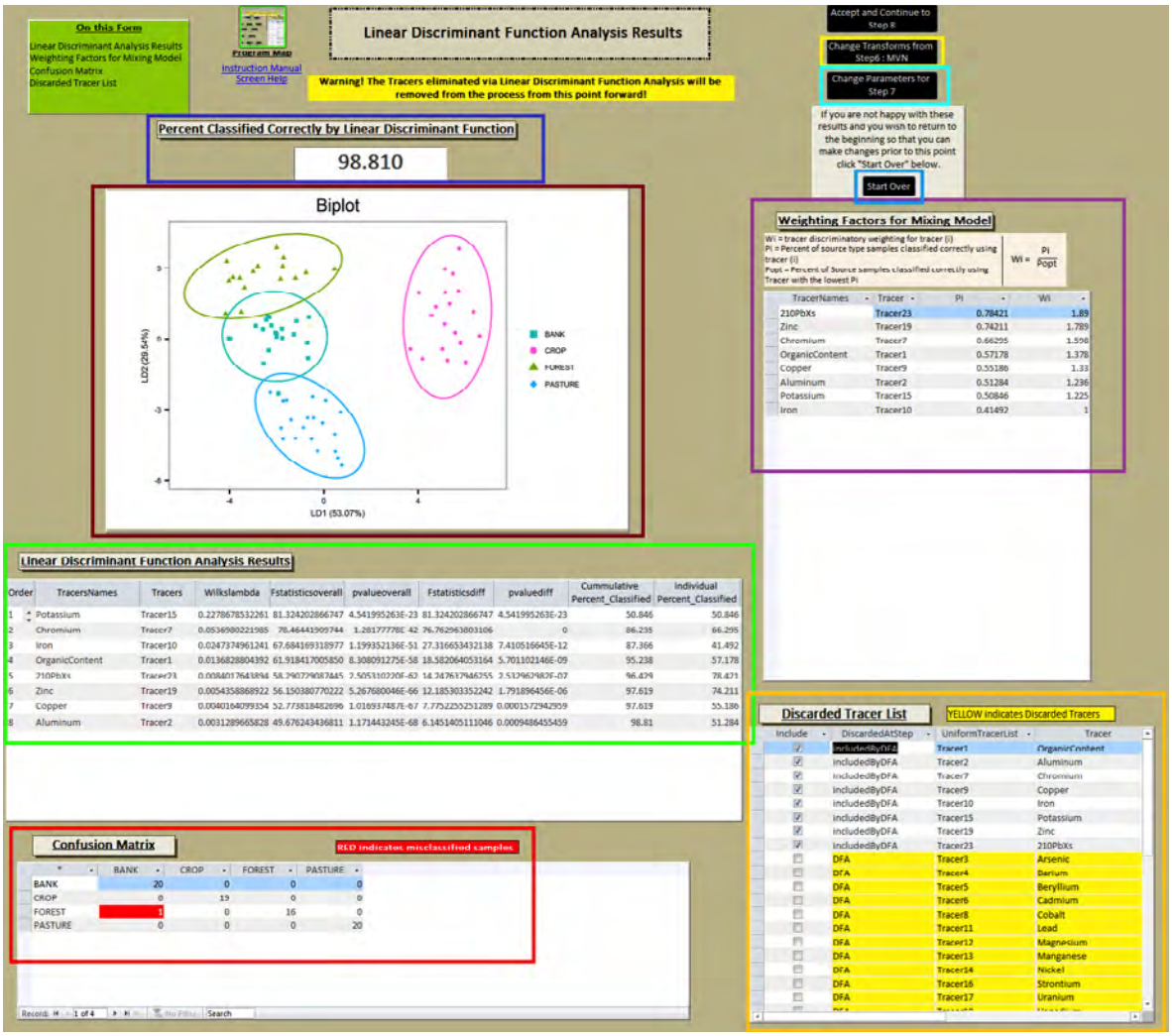


Figure 20–2. Homoscedasticity Test Results.

Appendix 21. Step 7 Output

For more details on the forward stepwise linear discriminant function analysis procedure, see *Start Step 7: Forward Stepwise Linear Discriminant Function Analysis*.



EXPLANATION

Forward stepwise linear discriminant function analysis results of the greedy.wilks() function (Weihs and others, 2005) with tracers in order of selection.

The confusion matrix (Kohavi and Provost, 1998) of source samples classified by type in leave-one-out cross validation (Lachenbruch and Mickey, 1968). The columns indicate the original source group designation, and the rows are the source group predicted by the linear discriminant function. The total percent of source samples correctly classified is given at the top.

Biplot of the first two linear discriminants.

The weighting factors are given in descending order, which does not necessarily correspond to the order of selection in (Forward Stepwise Linear discriminant function analysis results). $W_i = 1$ for the weakest indicator; all other tracers have $W_i > 1$.

The list of discarded tracers along with the step at which they were eliminated. Only tracers with the "Include" checkbox checked are analyzed from this point forward. This table is locked. Tracers discarded in Step 7 can be included if and only if uniform weighting is applied in Start Step 8: Mixing Model and Error Analysis.

Click to change whether or not data are transformed in Start Step 6: Multivariate Normality Test prior to running forward stepwise linear discriminant function analysis.

Click to change the p-value threshold for the greedy.wilks() function (Weihs and others, 2005).

Click to start a new run at the import data step.

Figure 21–1. Step 7 Output Screen.

Appendix 22. Step 8 Output

For more details on the mixing model and error analysis procedure see, *Start Step 8: Mixing Model and Error Analysis*.

Step 8 Preparation

Figure 22–1. Step 8 Choose Target Samples screen allows the user to select target samples for the mixing model.

Mixing Model Results (With Monte-Carlo Simulation)

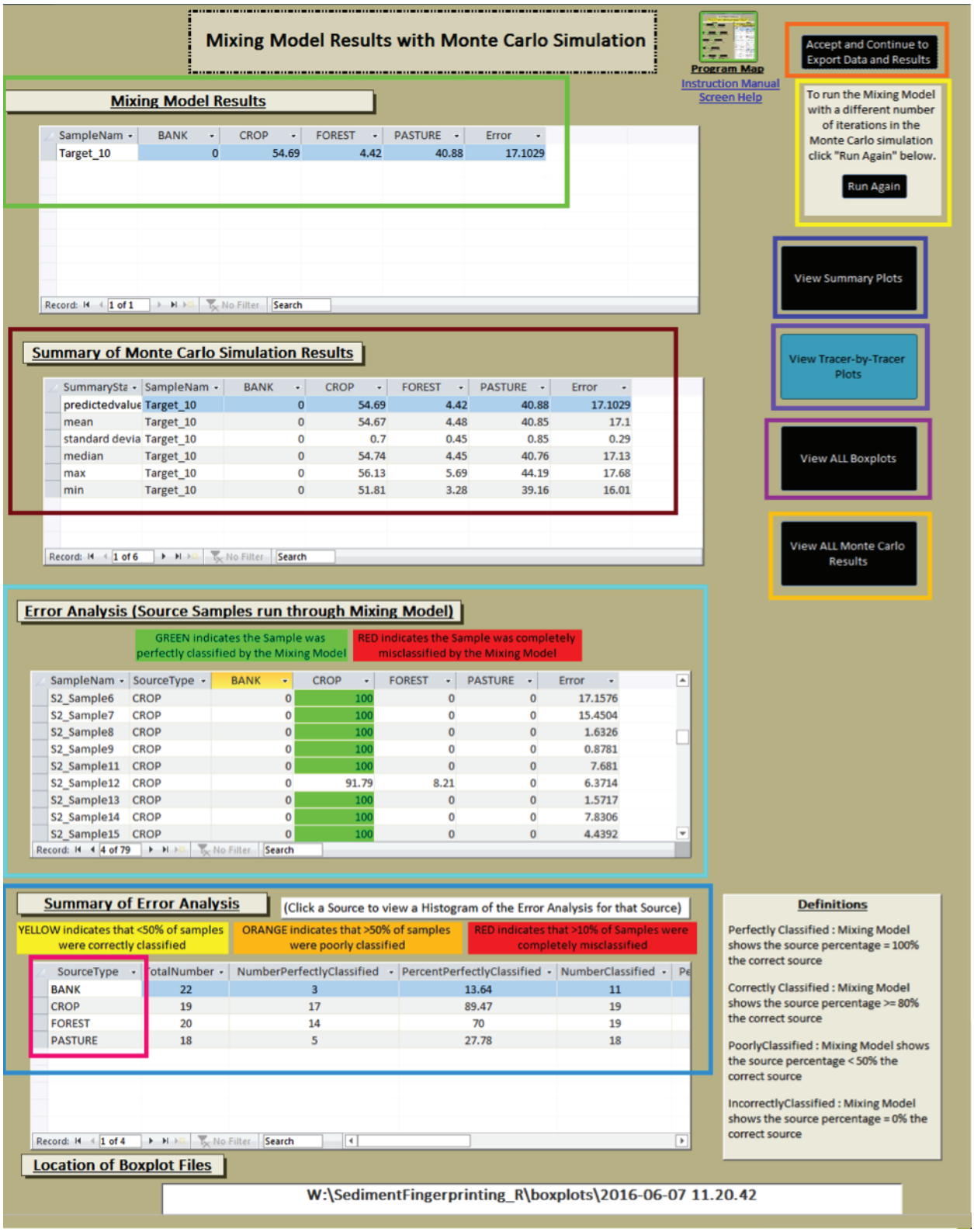


Figure 22–2. Mixing Model Results screen if the user elected to run a *Monte-Carlo Simulation*.

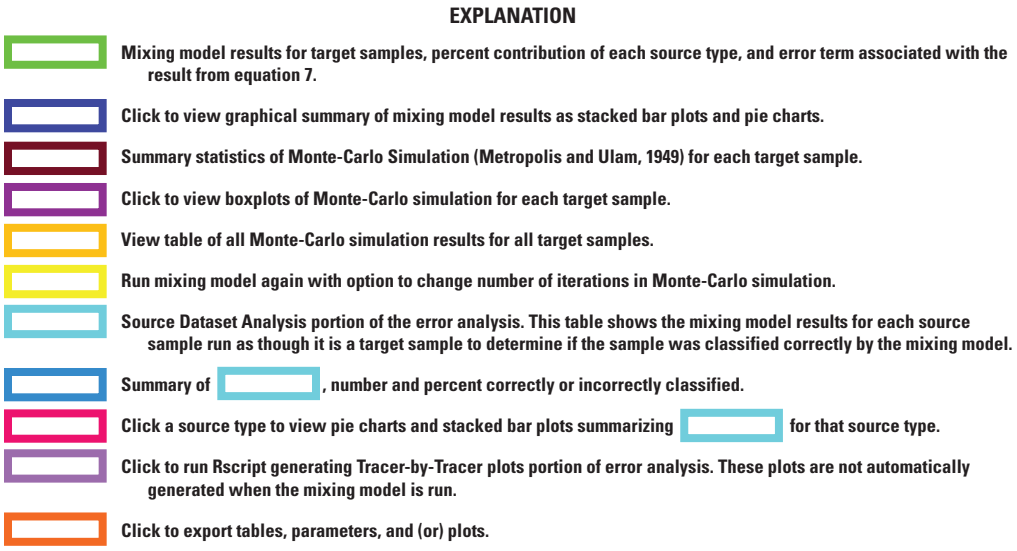
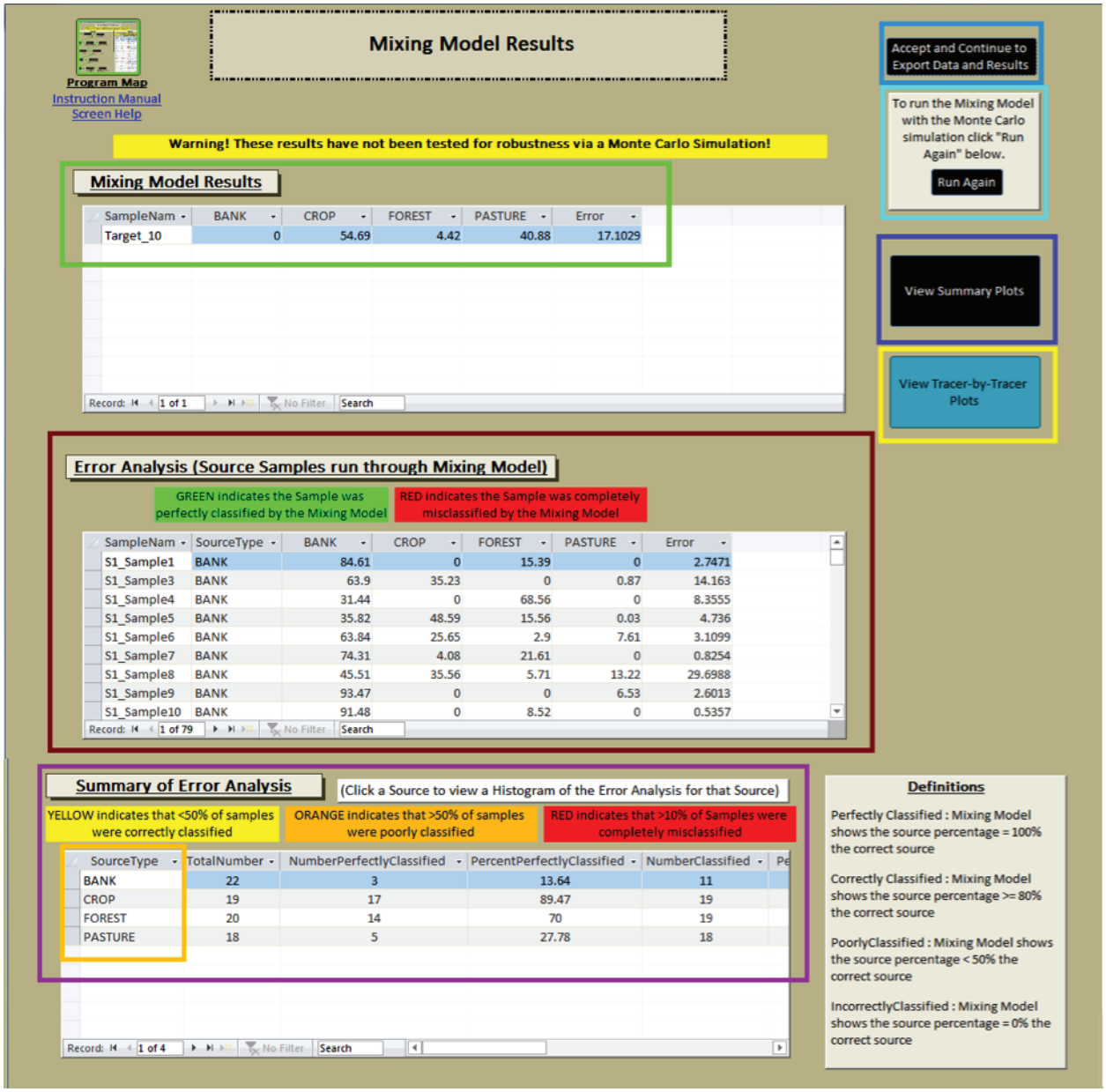


Figure 22–2. Mixing Model Results screen if the user elected to run a *Monte-Carlo Simulation*.—Continued

Mixing Model Results (Without Monte-Carlo Simulation, *N*=0)



EXPLANATION

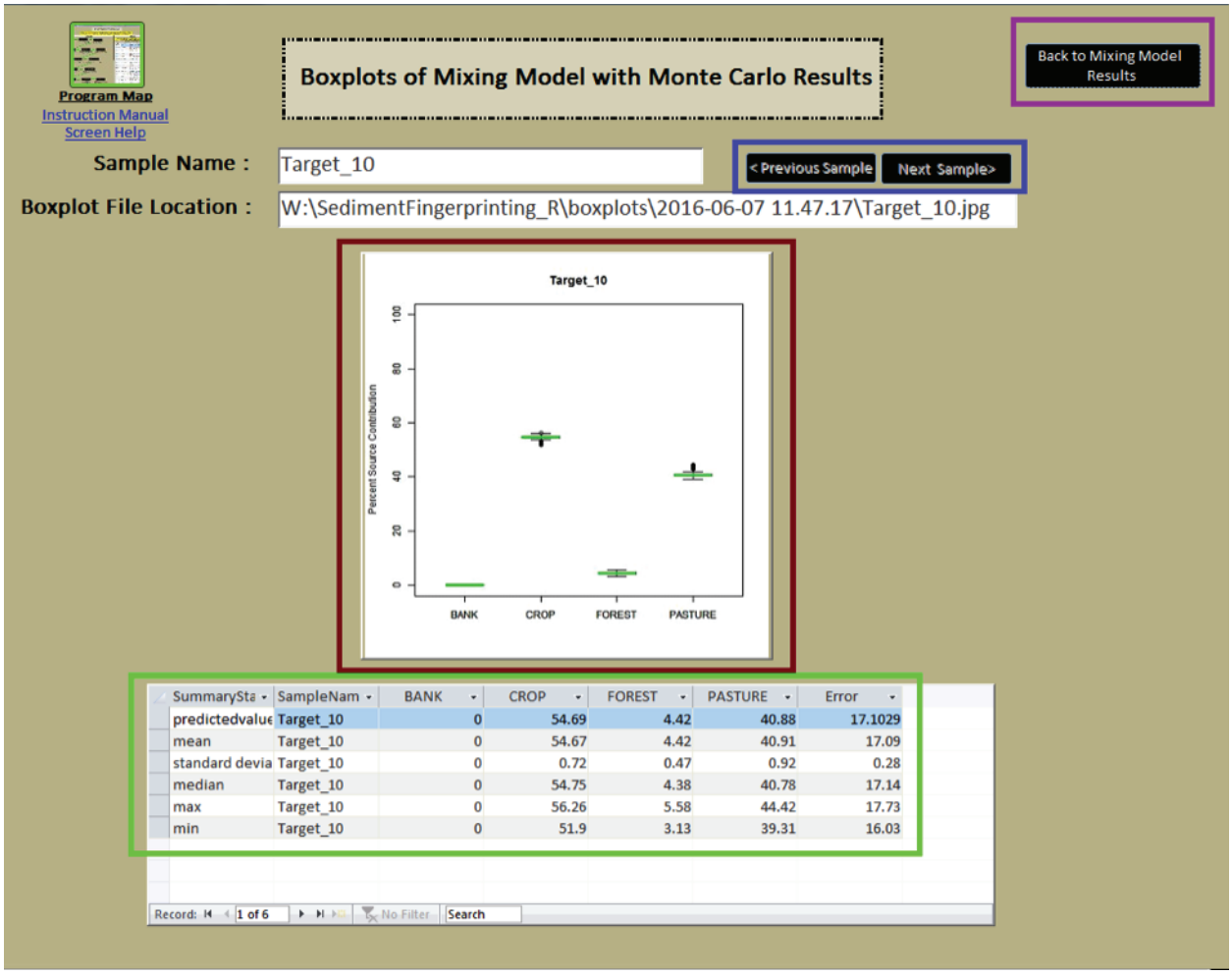
- Mixing model results for target samples, percent contribution of each source type, and error term associated with the result from equation 7.
- Click to view graphical summary of mixing model results as stacked barplots and pie charts.
- Source Dataset Analysis portion of the error analysis. This table shows the mixing model results for each source sample run as though it is a target sample to determine if the sample was classified correctly by the mixing model.
- Summary of [Source Dataset Analysis], number and percent correctly or incorrectly classified.
- Click a source type to view pie charts and stacked bar plots summarizing [Summary of Error Analysis] for that source type.
- Click to run Rscript generating Tracer-by-Tracer plots portion of error analysis. These plots are not automatically generated when the mixing model is run.
- Click to run mixing model again with Monte-Carlo Simulation.
- Click to export tables, parameters, and (or) plots.

Figure 22–3. Mixing Model Results screen if the user chose not to run a *Monte-Carlo Simulation*.

Error Analysis

Monte-Carlo Simulation

This screen will appear if the user clicks View ALL Boxplots on the Mixing Model Results (With Monte-Carlo Simulation) output screen. Running a Monte-Carlo simulation is optional. For more details on the Monte-Carlo simulation procedure, see *Monte-Carlo Simulation*.



EXPLANATION

Summary statistics of Monte-Carlo Simulation for the target sample shown.

Click to navigate to another target sample.

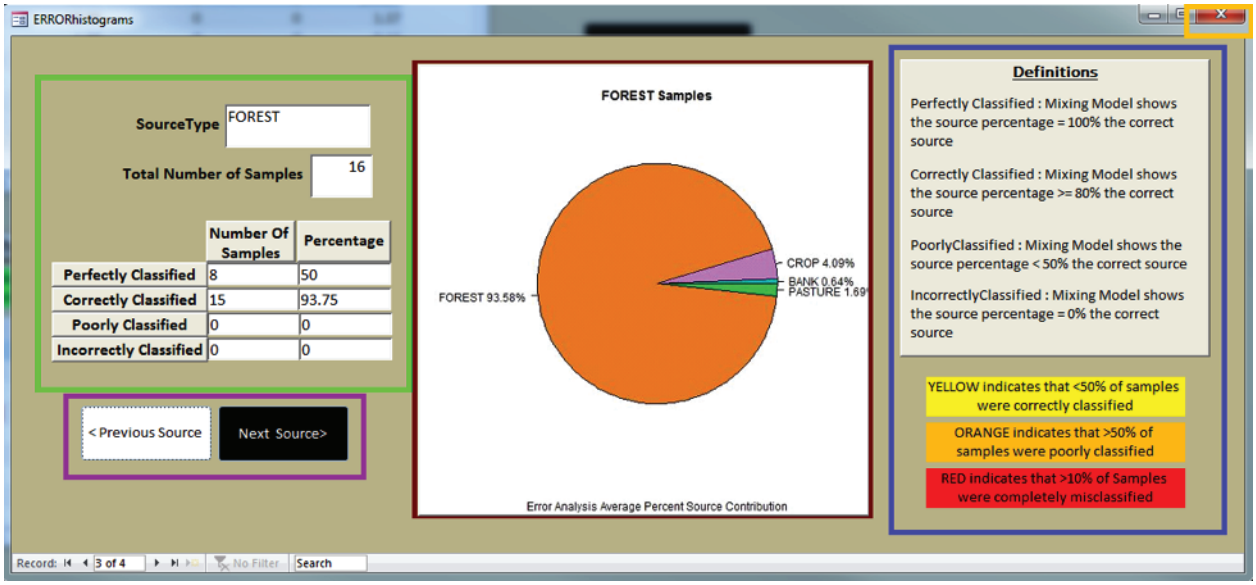
Boxplot of Monte-Carlo simulation. Mixing model results (eq. 7) based on all source samples are given as a green line. The median value is the black line.

Click to return to all mixing model results.

Figure 22–4. *Monte-Carlo Simulation* Results screen.

Source Dataset Analysis

This screen will appear if the user clicks a source type in the summary of Source Dataset Analysis table. Shown as ▭ in *Mixing Model Results (With Monte-Carlo Simulation)* and ▭ in *Mixing Model Results (Without Monte-Carlo Simulation, N=0)*. For more details on this portion of the error analysis procedure see Source Dataset Analysis.



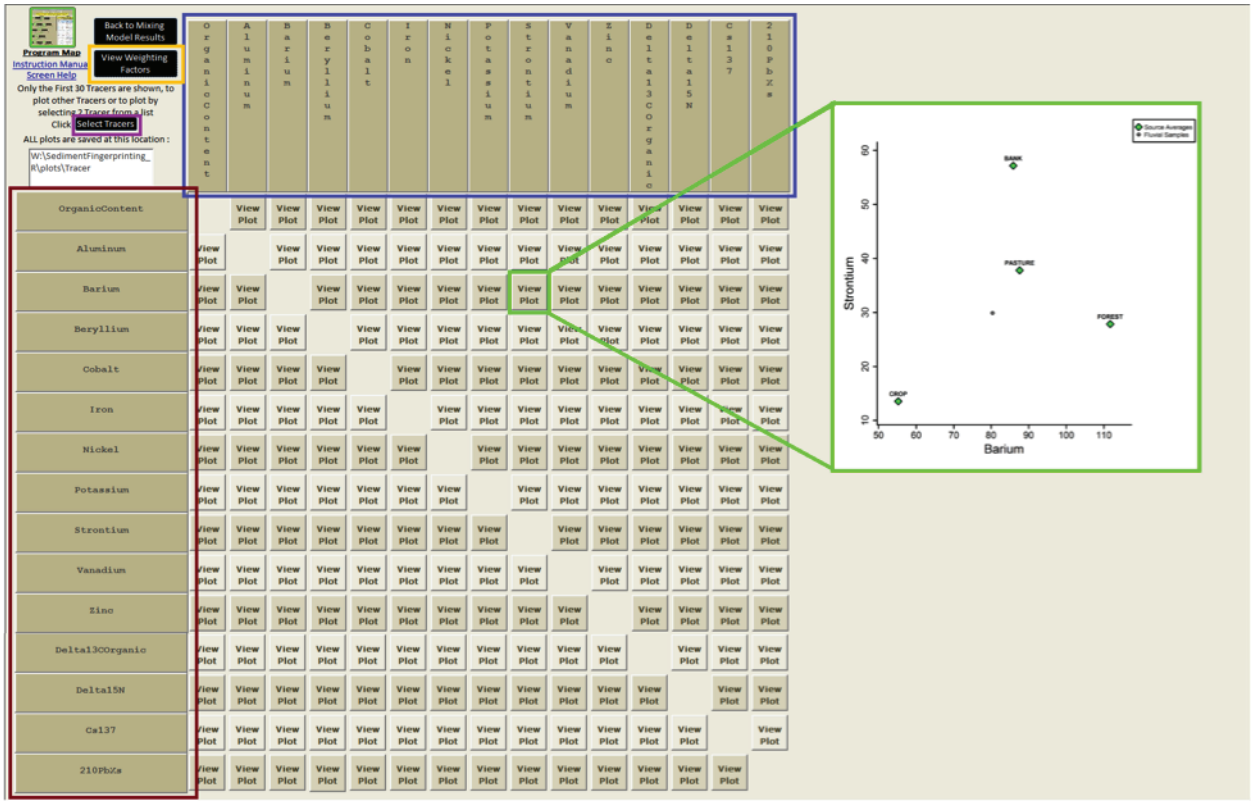
EXPLANATION

- **Summary of mixing model results for the selected source type.**
- **Definitions of levels of classification (i.e. "perfectly classified," "correctly classified," etc.)**
- **If pie chart is shown, click to flip and view a stacked bar plot of all source sample in the selected source type. If stacked bar plot is shown, click to view pie chart.**
- **Click to navigate to another source type.**
- **Click to close pop-up and return to mixing model results.**

Figure 22–5. Source Dataset Analysis screen.

Tracer-by-Tracer Plots

This screen will appear if the user clicks View Tracer-by-Tracer Plots on the initial Step 8 Output screen (fig. 22–2 or fig. 22–3.) To view Tracer-by-Tracer plots, click the View Plot cell, and a plot will appear with the x-axis corresponding to the tracer listed horizontally in the matrix and the y-axis corresponding to the tracer listed vertically in the matrix for the selected cell. Only the first 30 tracers (of the tracers used in the Mixing Model) will appear in the matrix. If more than 30 tracers remain to be used in the mixing model (post Start Step 7: Forward Stepwise Linear Discriminant Function Analysis), the user can select tracers for the x- and y-axis from a list of tracers by clicking Select Tracers.



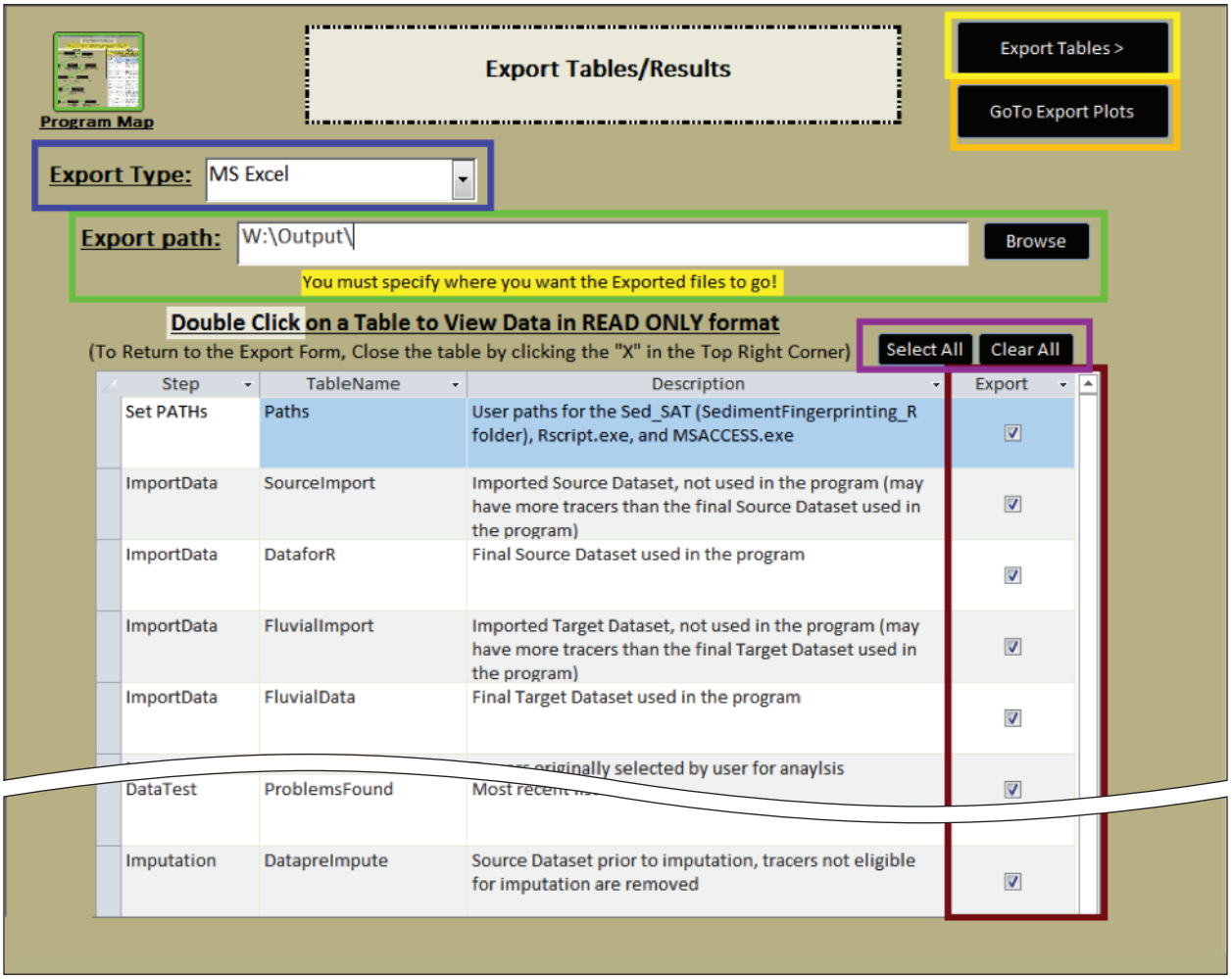
EXPLANATION

- **Click to view tracer-by-tracer plot.**
- **Tracers on x-axis.**
- **Tracers on y-axis**
- **Only the first 30 tracers are shown in the matrix; if your dataset contains more than 30 tracers (post Start Step 7: Forward Stepwise Linear Discriminant Function Analysis), click to select tracers to plot.**
- **Click to view Weighting Factors table to inform the user choice in selecting tracers to plot.**
- **Click to return to Mixing Model results.**

Figure 22–6. Tracer-by-Tracer Plots screen.

Appendix 23. Export Screens

Export Tables



EXPLANATION

Browse to the desired output folder.

Select export file type; if "MS Access" (Microsoft Access®), other fields will appear asking for the database name.

Select tables to export.

Click to select or clear all selected tables for export.

Click to export plots.

Export tables will be organized in files by step if is "MS Excel" (Microsoft Excel®) or in sections of a database if "MS Access" (Microsoft Access®).

Figure 23–1. Export Tables and Results Screen.

Export Plots

Figure 23–2. Export Plots screen.

Prepared by USGS West Trenton Publishing Service Center

For additional information, contact:
Director, MD-DE-DC Water Science Center
U.S. Geological Survey
5522 Research Park Drive
Baltimore, MD 21228

Visit our website:
http://md.water.usgs.gov

Visit the Sed_SAT software repository:
https://doi.org/10.5066/F76Q1VBX

ISSN 2331-1258 (online)
https://doi.org/10.3133/ofr20171062