PharmaSUG 2015 - Paper PO24

Introducing a Similarity Statistic to Compare Data Libraries for Improving Program Efficiency for Similar Clinical Trials

Taylor Markway, PRA Health Sciences, Raleigh, North Carolina
Amanda Johnson, PRA Health Sciences, Raleigh, North Carolina

ABSTRACT

Measuring the similarity between studies allows programmers to follow the best development strategies. If the similarity is overestimated development strategies focused on code reuse will lead to a breakdown in quality or timelines by overextending resources attempting to maintain an unrealistic pace. Alternatively, underestimating the similarity creates inefficiencies through duplication of effort.

This paper introduces a similarity statistic (Study Similarity Factor or SSF) to quantify similarity between two studies. This statistic is developed to determine the similarity between any two SAS ® data libraries with no requirements for following any industry standards such as CDISC. However, the statistic's value is enhanced through the widespread adaptation of such standards.

This paper will detail the code using Base SAS ® to generate the SSF and show that the SSF can be tailored to suit an organization's particular needs. In the examples presented throughout this paper the SSF is a normalized frequency of matching combinations of unique data sets and variables as well as variable only matches.

Through live study examples this paper will show how a systematic standard for measuring study similarity leads to more informed decision making and improved efficiency in software development. This tool can be made part of the planning process to make programming in clinical research more efficient by answering the question 'How similar is Study A to Study B?' This information can then be used to assess a particular development strategy; for example, using the same resource team on studies with overlapping timelines.

INTRODUCTION

Two earlier papers detail code on how to compare databases and data libraries. These papers are in DBCompare: A powerful tool to compare data definitions of datasets across multiple databases (Balakrishna Dandamudi NESUG 2006) and Programmatically Comparing Data Libraries (Stephen Hamburg NESUG 2006). The SSF builds on these methods by consolidating the raw data they generate into concise information for making decisions and to easily communicate similarity of studies outside of the SAS programming team.

The code detailed in Dandamudi and Hamburg's papers provide similar detailed outputs of the differences and similarities between two databases at a variable and dataset level. Thereby answering the question of 'What are all the ways these two studies are alike, and different?' This output is useful to the SAS programmer once a similar study has been identified, but how does one initially find that study? This paper builds on the methods and techniques previously discussed to create a single summary statistic utilizing the detailed reports in order to quickly assess which studies are the most similar.

Using the similarity of case report form (CRF) pages is a proxy measure that can be improved upon by using the SSF which measure the similarity of the database directly. From this direct measurement decision are more informed and a vast number of studies can be compared against quickly.

THE SSF

The first and most basic SSF we propose is a frequency count of the similar datasets and variable combinations. Table 1 shows an SSF of 62.5 along with the number of dataset-variable combinations as well as the count of unique combinations in each study. The dataset-variable determination of the SSF in this example is based on the assumption that differences in data type, length, format and label are negligible and do not require significant program modifications for code reuse.

Table 1. Report generation example of Study Similarity Factor (bold and italic).

STATUS	COUNT	PERCENT
BOTH	2401	***62.5***
Study A	904	23.5
Study B	536	14.0

The exact computation of the SSF could be determined in many different ways. What is proposed in this paper is one of the simplest iterations. Of key importance to using an SSF is that once a method of computation is determined that it is used consistently in order to accurately guide decisions going forward. Later in the paper pitfalls and limitations will be

DIGESTING THE CODE

An example of the code used to calculate the SSF in Table 1 is below. In the example below a DATA step is used to merge metadata from the CONTENTS procedure. When the data is merged a status flag is created which indicates if a dataset-variable combination is in common or unique to one of the libraries. The status flag in the example below has values of “BOTH”, “&NAM1” and “&NAM2” where the value of &NAM1 and &NAM2 are determined by user input.

Code Block 1. An example of creating the SSF.

```
*Define Macro to obtain database metadata*;
%macro get_metadata(lib=, out=);
proc contents data=&lib.._all_ out=&out. noprint;
run;

proc sort data=&out;
    by memname name;
run;
%mend get_metadata;

*Define Macro to Calculate SSF *;
%macro get_ssf(lib1=, nam1=, lib2=, nam2=);

*Call macro to get proc contents of all data in library *;
%get_metadata(lib=&lib1., out=lib1);
%get_metadata(lib=&lib2., out=lib2);

*Data step to create similarity status flag*;
data combine;
    length status $7.;
    merge lib1 (in=in1) lib2 (in=in2);
    by memname name;

    if in1 and in2 then status = ‘BOTH’;
    else if in1 and not(in2) then status=”&NAM1.”;
    else if not(in1) and in2 then status=”&NAM2.”;
run;

*Percent Similarity Determination*;
proc freq data=combine noprint;
    table status / out=overall;
run;

%mend get_ssf;
```

Code Block 1. An example of creating theSSF.

With some slight modifications to the %get_ssf macro in Code Block 1 an entire list of potentially similar studies can have an SSF calculated to determine the best candidate. An example of the type of output which can be generated from this approach can be seen in Table 2.

Table 2 shows the SSF for Study A when compared to five different studies. The difference between Table 1 and Table 2 is that for each study Table 2 only displays the SSF (the ‘BOTH’ status percentage in Table 1). This allows users to quickly identify which studies have the most similar database, in Table 2 this would be Study F. Based on this knowledge we know Study F would be a good starting place for reuse of specifications and programs.

Table 2. SSF for list of studies

STUDY	SSF
Study B	62.5
Study C	13.1
Study D	35.2
Study E	43.6
Study F	80.8

USING THE SSF

By assessing similarity within the CDISC standard environment the SDTM domains represent a rigid structure between the more flexible CDASH eCRF data collection and the analysis ADaM datasets. By calculating the SSF one can improve programming efficiencies by finding the ideal starting point for SDTM programming.

In the case study taken from actual clinical trials one can see how the SSF can improve efficiency and inform software development strategies.

CODE IN ACTION – CASE STUDY

The SSF was calculated on a set of four phase III studies which had already been closed out. These four studies were picked because they had the same leadership, same project team and same timelines. The resulting SSF from these four studies was then used to determine the resourcing strategy for a new set of two studies. First the four phase III studies and their resourcing strategies will be reviewed. Next we will show how the SSF was used to analyze what happened and how this helped determine the resourcing on two new studies.

The four phase III studies which had been closed out were two pairs of similar studies which initially followed the same resourcing strategy of having one team reuse the SAS code from one study on the other. However, for one pair of studies, which we will call A & B it quickly became apparent that the team was spending a large amount of time having to update the code between A & B and would miss the targets if action was not taken. The studies A & B were then divided up amongst two separate teams. The other pair of similar studies, which we will call C & D, did not encounter this issue and reuse of code with one team was a successful strategy.

What led the A & B team to fail while the C & D team had success?

Table 3. SSF for Case Study

STUDY	SSF
Study A & B	78.7
Study C & D	97.8

The SSF for the studies is summarized in Table 3. From Table 3 it is clear that C & D are much more similar than A & B. So the one team approach was a success when the SSF was 97.8 and a failure when it was 78.7. It is important to note that A & B had the same sponsor, phase, indication and design – they only differed in route of administration. Likewise for C & D - they too only differed in route of administration. To give these numbers some context, two unrelated studies (different design/phase/indication/sponsor) matched at 22.9 and two studies with same the same sponsor, phase, design, but different indication match at 56.

The project management for the new pair of studies was pushing to use one team and simply recycle the code to be as efficient as possible. However, when the SSF was calculated for the two new studies it was 60.1. When this data was taken to the project and programming management team it was decided that a one team strategy would not be followed and we would start programming with two teams.

From this example it can be seen how the SSF provides an objective measure of database similarity that can be communicated to the project and management team for gauging the feasibility development of strategies.

PITFALLS AND LIMITATIONS

This section of the paper discusses potential pitfalls in SSF calculation methods and some of the limitations of calculating the SSF from studies’ SAS data sets. Enhancements to the SSF can easily be introduced to weight certain types of metadata or to take additional parameters into account. The important thing to bear in mind when making these changes is to ensure when comparing SSF values that they were calculated in the same way.

EDC SYSTEM – THE IMPORTANCE IT HOLDS

It is suggested that comparison of studies takes place for two studies with the same electronic data capture (EDC) system. This is because comparisons of two studies with different EDC systems are bound to be dissimilar in regard to the SSF. If the SSF is thought of as the percentage similarity between the data captured between two studies then the SSF would be an underestimate of similarity due to system variables when comparing across different EDC systems and overestimate when comparing within the same system. One way to account for this is to remove the EDC System variables from the metadata data sets prior to calculating the SSF. For example, Table 1 displays the SSF result prior to removal of EDC System Variables while Table 4 displays the SSF result after the EDC System Variables have been removed.

Table 4. Report generation example of Study Similarity Factor with EDC variables removed.

STATUS	COUNT	PERCENT
BOTH	863	***37.6***
Study A	432	18.8
Study B	1001	43.6

Comparing Table 1 to Table 4 the value of the SSF has dropped by 24.9 after the removal of EDC system Variables. Depending on the number of CRF pages and therefore the size of the database the impact could vary.

LIMITATIONS

This statistics can only be calculated after a database has been designed and datasets can be extracted. Therefore it is not applicable to the bidding process and early stages of the planning process. It is also important to note that if the calculation of the statistic is changed then its usefulness to compare successful and unsuccessful development strategies becomes less informative - for example, to keep the EDC variables in or out of the calculation.

CONCLUSION

The SSF provides an objective measure of study similarity. With the wider adaptation of the CDISC standards SAS programming in the pharmaceutical industry is moving towards automation. Until full automation becomes a reality reusing similar code can be a programmer most efficient strategy. The SSF provides a way to make reusing code more efficient.

This paper introduces just one way of calculating the similarity between studies. With this method this paper showed that by measuring the similarity between studies programmers gain additional knowledge which can help development strategies. The method and implementation of this statistic can be improved upon and expanded and with wider use of such statistics we can all work to improve the answer to the question: “how similar is study A to B?”

REFERENCES

- Hamburg, Steven. 2006. “Programmatically Comparing Data Libraries.” NESUG 2006. Available at http://www.lexjansen.com/nesug/nesug06/cc/cc29.pdf
- Dandamudi, Balakrishna. 2006. “DBCompare: A powerful tool to compare data definitions of datasets across multiple databases”. NESUG 2006. Available at http://www.lexjansen.com/nesug/nesug06/cc/cc28.pdf

ACKNOWLEDGMENTS

The authors of this paper would like to thank our colleagues at PRA Health Sciences for their support.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Name: Taylor Markway
Enterprise: PRA Health Sciences
Address: 4130 Parklake Avenue Suite 400
City, State ZIP: Raleigh, North Carolina, 27612
Work Phone: 919-788-3055
E-mail: markwaytaylor@prahs.com
Web: www.prahs.com

Name: Amanda Johnson
Enterprise: PRA Health Sciences
Address: 4130 Parklake Avenue Suite 400
City, State ZIP: Raleigh, North Carolina, 27612
Work Phone: 919-786-8658
E-mail: anjohnson@alumni.peace.edu
Web: www.prahs.com