PharmaSUG China

From Coal Mining to Data Mining: Advancing Programming Management for Clinical Projects with Text Analytics

Zhouming (Victor) Sun, MedImmune Corp., Gaithersburg, MD 20878, USA

ABSTRACT

Both coal mining and data mining involve the process of extracting valuable materials from raw resources to form economically useful packages. Due to the theoretical similarities behind these two different types of mining, the process of coal mining, as will be described, serves as inspiration for the development of the Simultaneous Monitoring of Analysis and Reporting Toolkit (SMART), a method for data mining based on the techniques of text analytics using Base SAS@ 9+.

Established with a general SAS macro, SMART is a versatile toolkit that allows for the reporting of the real-time status of programming activities. By using the techniques of text analytics to find explicit relationships between documents by classifying documents into predefined or data-driven categories, SMART makes management of clinical project programming more effective and dynamic. This paper will introduce the concepts of SMART, followed by a presentation of its four key processes. It will additionally demonstrate the power of text analytics in extracting useful information while providing a helpful roadmap for project leaders to efficiently manage programming activities independently of a project leader's programming skill or experience.

INTRODUCTION

Many changes have been made along the path of my career, from working as a mining engineer more than 20 years ago at the Coal Mine Research and Design Institute in China to currently serving as a Senior Statistical Programmer at MedImmune, an internationally well-known biological research and development arm of AstraZeneca. Through these changes from a coal to a data miner, the importance of extracting valuable materials has remained constant. By comparing the coal mining and pharmaceutical industries, we find that many similarities in the theory of extraction and mining, despite differences in practice.

There will first be a review of the methods of data mining of selecting, exploring, and modelling large amounts of data to uncover previously unknown patterns for a business advantage, as has been widely done in pharmaceutical companies. The paper will then address how to use text analytics as a sub-group of data mining techniques through using the latest version of Base SAS@ 9+ product in more practical manner.

Based on the popular text analytics approach Perl Regular Expressions (PRX) and in conjunction with the establishment of hash object and metadata, SMART(1) develops a new approach for improving the programming management of clinical projects, allowing for the production of sets of real-time status reports of concurrent programming activities. Those reports can be then used for sourcing planning, progress monitoring, delivery observing, validation checking, or summary auditing, making management more efficient and dynamic.

COAL MINING PROJECT VERSUS IND/NDA SUBMISSION

For a coal mining project, there are multiple phases that must be completed from the start of a project to the final site closure and abandonment, as shown below in Figure 1. Similarly, multiple phases are required when conducting an IND/NDA submission in a pharmaceutical company, as shown in Figure 2. These two processes are compared side-by-side in Table 1, with a focus on the fundamental aspects of risk/safety (2) and productivity/efficacy.

Figure 1 Outline of Coal Mining Methods

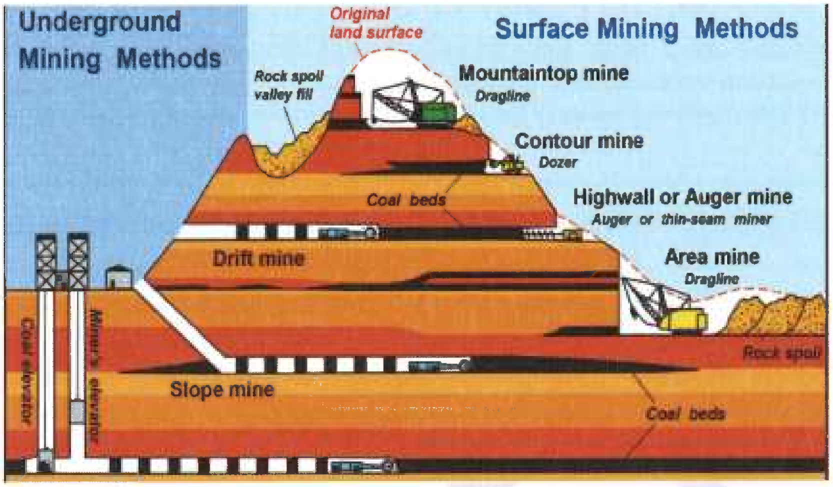


Figure 2 Outline of IND/NDA Submission

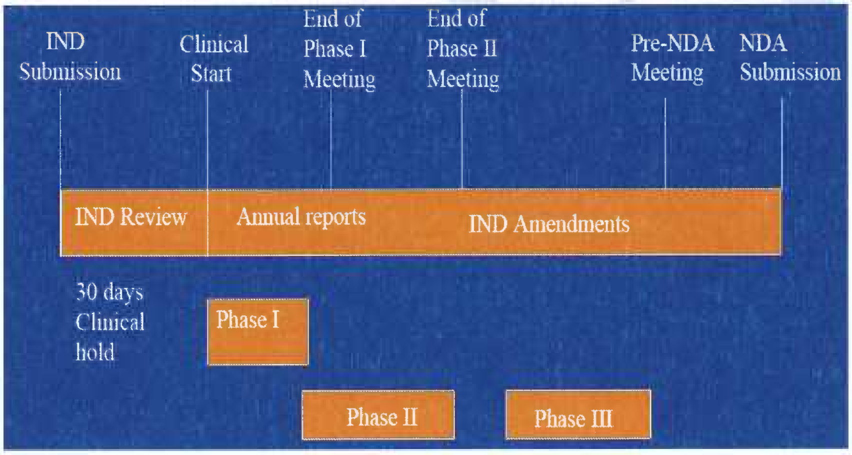


Table 1 Comparison of Processes of Coal Mining Project and IND/NDA Submission

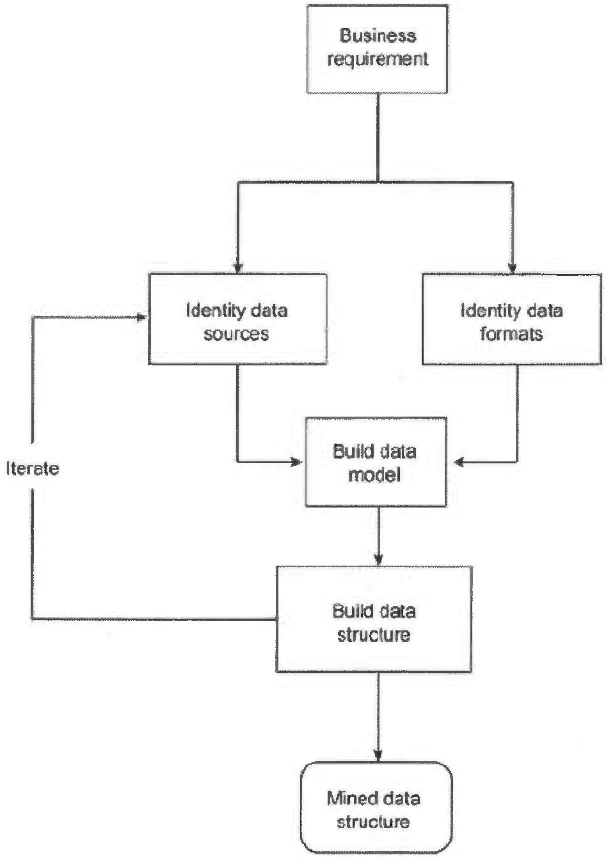
	Coal Mining Project		**IND/NDA Submission**	
A	Initial Phase	- Exploration - Investigation - Data Collection	Early Phase IND	Pre-Clinical IND Submission
B	Design Phase	- Pre-evaluation Study - Concept/Scoping Study - Pre-feasibility Study - Feasibility Study - Detailed Engineering - Site Construction - Operations - Safety/Risk Assessment - Auditability - Closure/abandonment	IND Amendments	- Phase 1 - Phase 2 - Phase 3
C	Regulatory	MSHA(US Mine Safety and Health Administration)	FDA(US Food and Drug Administration)	NDA - Pre-NDA Meeting - NDA Submission

DATA MINING AND TEXT ANALYSTICS

DATA MINING AND TECHNIQUES

Fundamentally, data mining is about processing data and identifying patterns and trends in the information. Data mining principles have been around for many years, but, with the advent of *big data*, data mining is even more prevalent [1]. Figure 3 outlines the general process of data mining.

Figure 3 Process of Data Mining



There are various approaches to data mining that have been applied in different industries. The most commonly used techniques include artificial neural networks, decision trees, and the nearest-neighbor method. Each of these techniques analyzes data in different ways.

Artificial neural networks are non-linear, predictive models that learn through training. Although they are powerful predictive modeling techniques, some of the power comes at the expense of ease of use and deployment. One area where auditors can easily use them is when reviewing records to identify fraud and fraud-like actions. Because of their complexity, they are better employed in situations where they can be used and reused, such as reviewing credit card transactions every month to check for anomalies.

Decision trees are tree-shaped structures that represent decision sets. These decisions generate rules, and are used to classify data. Decision trees are the favored technique for building understandable models. Auditors can use them to assess, for example, whether the organization is using an appropriate cost-effective marketing strategy that is based on the assigned value of the customer, such as profit.

Nearest-neighbor method classifies dataset records based on similar data in a historical dataset. Auditors can use this approach to define a document that is interesting to them and ask the system to search for similar items.

TEXT ANALYTICS AND TECHNIQUES

Text analysis, also called text mining or Information Extraction (IE), is the process of extracting salient information from documents, which can be in multiple languages, about pre-specified types of events, entities or relationships. These facts are then usually entered automatically into a database or spreadsheet, which may then be used to analyze the data for trends, used to give a natural language summary, or used for indexing purposes in Information Retrieval (IR) applications.

Text analytics is a multi-step process that involves accessing the unstructured text, parsing the text and turning it into actionable data, and analyzing the newly created data. There are four most commonly used techniques in practice for text analytics.

Text parsing automatically extracts terms and phrases from parts of speech, as well as "stemming" words, to reduce words to their root forms (e.g. run, ran, running would all map to run).

Automatic text cleaning checks spelling automatically in the specified language.

Dimension reduction uses techniques such as Singular Value Decomposition to automatically relate similar terms and documents and avoid having to generate industry-specific ontologies (categories of words or phrases).

Text clustering groups documents into common themes and topics based in their content. The clusters generated are then used to either generate hypotheses or as additional inputs into another more traditional data mining model.

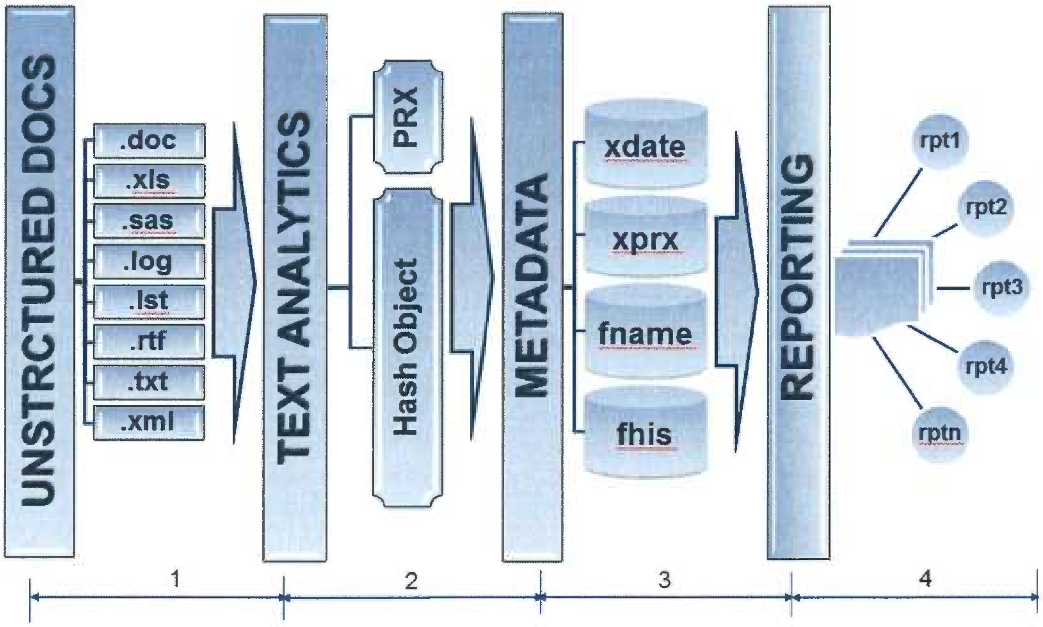
SIMULTANEOUS MONITORING OF ANALYSIS AND REPORTING TOOLKITS (SMART)

By using text analytics/Perl Regular Expressions (PRX) approach in conjunction with the hash object and metadata's establishment to find explicit relationships among unstructured documents and to classify them into predefined or data driven categories, SMART is established to serve as a new approach to improve the programming management of clinical projects.

KEY PROCESSES

The SMART consists of four key processes, as shown in Figure 4 below. They are (1) the collection of unstructured documents; (2) the use of text analytics per predefined business requirements; (3) the establishment of metadata; and (4) the creation of reports.

Figure 4 Key Processes of SMART



THE COLLECTION OF UNSTRUCTURED DOCUMENTS

This process groups together all documents that are related to programming for a project. They may exist in different formats (i.e. .doc, .exl, .lst, .sas, log) but can be arranged in an unstructured format and compared against any table (column vs. rows) from the database or any dataset (variable vs. record) from SAS$^{@}$.

TEXT ANALYTICS PER PREDEFINED BUSINESS REQUIRMENTS

Text analytics utilize PRX(3) in conjunction with the hash object(4) provided in SAS$^{@}$ 9^{+} in order to mine all desired information from the collected documents per the predefined requirements. A few functions used with PRX in SAS$^{@}$ 9^{+} are listed in Table 2 below.

Table 2 Major Functions, Syntax and Purpose of PRX

Function	Syntax	Purpose
PRXPARSE	PRXPARSE (Perl-regular-expression)	To define a Perl regular expression to be used later by the other Perl regular expression functions
PRXMATCH	PRXMATCH (pattern-id *or* regular-expression, string)	To locate the position in a string, where a regular expression match is found. This function returns the first position in a string expression of the pattern described by the regular expression. If this pattern is not found, the function returns a zero.
CALL PRXSUBSTR	CALL PRXSUBSTR(pattern-id, string, start, <length>)	To be used with the PRXPARSE function to locate the starting position and length of a pattern within a string. The PRXSUBSTR call routine serves much the same purpose as the PRXMATCH function plus it returns the length of the match as well as the starting position.
CALL PRXPOSN	CALL PRXPOSN(pattern-id, capture-buffer-number, start, <length>)	To return the position and length for a capture buffer (a subexpression defined in the regular expression). Used in conjunction with the PRXPARSE and one of the PRX search functions (such as PRXMATCH).
CALL PRXNEXT	CALL PRXNEXT(pattern-id, start, stop, position, length)	To locates the nth occurrence of a pattern defined by the PRXPARSE function in a string. Each time you call the PRXNEXT routine, the next occurrence of the pattern will be identified.
PRXPAREN	PRXPAREN(pattern-id)	To function returns a value indicating the largest capture-buffer number that found a match. You may want to use this function with the PRXPOSN function.
CALL PRXCHANGE		To substitute one string for another. One advantage of using PRXCHANGE over TRANWRD is that you can search for strings using wild cards. Note that you need to use the "s" operator in the regular expression to specify the search and replacement expression (see the explanation following the program).
CALL PRXFREE		
CALL PRXDEBUG		

THE ESTABLISHMENT OF METADATA

Multiple metadata are created during this procedure, based off of the results from certain structures of text analytics, to facilitate a fast-paced reporting process.

THE CREATION OF REPORTS

In this stage, specific reports can be selected and produced to help programming managers plan sourcing, monitor progress, observe delivery, check validation, and summarize audits.

ILLUSTRATING OUTPUTS WITH EXAMPLES

Four outputs among the set of reports produced by SMART are shown below as examples of some metadata content and illustrate the usefulness of managing programming. Unlike other approaches of tracking clinical projects with requirement of manual based-on inputs for each reporting cycle(5), the outputs from SMART are the production of sets of real-time status reports of concurrent programming activities, and only the baseline information per a project plan is necessary.

Table 3 shows the conventional approach, in which each individual involved in the project has to enter his or her status in a timely manner into an Excel file using a tracking tool. A project leader will then create the status report using the Excel file. Delays in status updates, overwrites by multiple users, or missing revisions of the Excel file will lead to inaccurate status report results.

Table 3 Statistical Programming Status Report
(Based on Information Provided by Individuals)

OVERALL
Reported on 13MAR2014/15:15

```
                                                PRODUCTION              VALIDATION
PRIORITY  CATEGORY 1           CATEGORY 2       STATUS                  STATUS
------------------------------------------------------------------------------------------------

       1  DATASET              BASELINE             5/   5 ( 100%)          5/   5 ( 100%)
                               EFFICACY            11/  11 ( 100%)         11/  11 ( 100%)
                               SAFETY               4/   4 ( 100%)          4/   4 ( 100%)
                               TOTAL               20/  20 ( 100%)         20/  20 ( 100%)

          TABLE                BASELINE             9/   9 ( 100%)          9/   9 ( 100%)
                               EFFICACY            32/  32 ( 100%)         32/  32 ( 100%)
                               SAFETY              18/  18 ( 100%)         18/  18 ( 100%)
                               TOTAL               59/  59 ( 100%)         59/  59 ( 100%)

          FIGURE               EFFICACY            18/  18 ( 100%)         18/  18 ( 100%)
                               SAFETY               5/   5 ( 100%)          5/   5 ( 100%)
                               TOTAL               23/  23 ( 100%)         23/  23 ( 100%)

       2  DATASET              BASELINE             3/   3 ( 100%)          3/   3 ( 100%)
                               EFFICACY             5/   5 ( 100%)          5/   5 ( 100%)
                               SAFETY               4/   4 ( 100%)          4/   4 ( 100%)
                               TOTAL               12/  12 ( 100%)         12/  12 ( 100%)

          TABLE                BASELINE             8/   8 ( 100%)          8/   8 ( 100%)
                               EFFICACY            66/  66 ( 100%)         66/  66 ( 100%)
                               SAFETY              27/  27 ( 100%)         27/  27 ( 100%)
                               TOTAL              101/ 101 ( 100%)        101/ 101 ( 100%)

          FIGURE               EFFICACY            27/  27 ( 100%)         27/  27 ( 100%)
                               SAFETY              11/  11 ( 100%)         11/  11 ( 100%)
                               TOTAL               38/  38 ( 100%)         38/  38 ( 100%)

       3  LISTING              BASELINE             0/  16 ( 0.0%)      NA
                               EFFICACY             0/  15 ( 0.0%)      NA
                               SAFETY               0/  17 ( 0.0%)      NA
                               TOTAL                0/  48 ( 0.0%)      NA
```

Table 4 below is based on real-time information of the complete production and validation of datasets and Tables, Figures, and Listings (TFLs), in accordance with specified business rules. This information is directly extracted from the current documents and does not depend on the input from each individual. Therefore, the most current status report will reflect the concurrent progress of the programming activities.

Table 4 Statistical Programming Status Report
(Based on PRX Results By Predefined Rules)

OVERALL
Reported on 13MAR2014/15:55

```
                                                PRODUCTION              VALIDATION
PRIORITY  CATEGORY 1           CATEGORY 2       STATUS                  STATUS
------------------------------------------------------------------------------------------------

       1  DATASET              BASELINE             5/   5 ( 100%)          4/   5 (80.0%)
                               EFFICACY             9/  11 (81.8%)          8/  11 (72.7%)
                               SAFETY               2/   4 (50.0%)          3/   4 (75.0%)
                               TOTAL               16/  20 (80.0%)         15/  20 (75.0%)

          TABLE                BASELINE             9/   9 ( 100%)          7/   9 (77.8%)
```

		EFFICACY	32/ 32 (100%)	31/ 32 (96.9%)
		SAFETY	16/ 18 (88.9%)	17/ 18 (94.4%)
		TOTAL	57/ 59 (96.6%)	55/ 59 (93.2%)
	FIGURE	EFFICACY	15/ 18 (83.3%)	12/ 18 (66.7%)
		SAFETY	5/ 5 (100%)	4/ 5 (80.0%)
		TOTAL	20/ 23 (87.0%)	16/ 23 (69.6%)
2	DATASET	BASELINE	2/ 3 (66.7%)	2/ 3 (66.7%)
		EFFICACY	5/ 5 (100%)	4/ 5 (80.0%)
		SAFETY	3/ 4 (75.0%)	3/ 4 (75.0%)
		TOTAL	10/ 12 (83.3%)	9/ 12 (75.0%)
	TABLE	BASELINE	6/ 8 (75.0%)	7/ 8 (87.5%)
		EFFICACY	62/ 66 (93.9%)	65/ 66 (98.5%)
		SAFETY	25/ 27 (92.6%)	23/ 27 (85.2%)
		TOTAL	93/ 101 (92.1%)	95/ 101 (94.1%)
	FIGURE	EFFICACY	24/ 27 (88.9%)	20/ 27 (74.1%)
		SAFETY	10/ 11 (90.9%)	10/ 11 (90.9%)
		TOTAL	34/ 38 (89.5%)	30/ 38 (78.9%)
3	LISTING	BASELINE	0/ 16 (0.0%)	NA
		EFFICACY	0/ 15 (0.0%)	NA
		SAFETY	0/ 17 (0.0%)	NA
		TOTAL	0/ 48 (0.0%)	NA

Table 5 below provides more detailed information on the validation status at an individual level. The example below indicates there eight different users involved in this task whose require validation of their analysis of datasets and TFLs. Each individual has different assignments, and some individuals have completed their validation under different priorities and categories. Depending on the delivery timeline, a project leader can make any necessary resource adjustments to the report.

Table 5 Statistical Programming Status Report

BY VALIDATOR
Reported on 16MAR2014/ 8:49

PRIORITY	CATEGORY 1	CATEGORY 2	VALIDATION PROGRAMMER	PRODUCTION STATUS	VALIDATION STATUS
1	DATASET	BASELINE	PROGRAMMER_7	5/ 5 (100%)	4/ 5 (80.0%)
		EFFICACY	PROGRAMMER_10	4/ 6 (66.7%)	3/ 6 (50.0%)
			PROGRAMMER_4	3/ 3 (100%)	3/ 3 (100%)
			PROGRAMMER_8	2/ 2 (100%)	2/ 2 (100%)
		SAFETY	PROGRAMMER_7	2/ 4 (50.0%)	3/ 4 (75.0%)
		TOTAL	PROGRAMMER_10	4/ 6 (66.7%)	3/ 6 (50.0%)
			PROGRAMMER_4	3/ 3 (100%)	3/ 3 (100%)
			PROGRAMMER_7	7/ 9 (77.8%)	7/ 9 (77.8%)
			PROGRAMMER_8	2/ 2 (100%)	2/ 2 (100%)
	TABLE	BASELINE	PROGRAMMER_7	9/ 9 (100%)	7/ 9 (77.8%)
		EFFICACY	PROGRAMMER_5	8/ 8 (100%)	8/ 8 (100%)
			PROGRAMMER_8	17/ 17 (100%)	16/ 17 (94.1%)
			PROGRAMMER_9	7/ 7 (100%)	7/ 7 (100%)
		SAFETY	PROGRAMMER_7	16/ 18 (88.9%)	17/ 18 (94.4%)
		TOTAL	PROGRAMMER_5	8/ 8 (100%)	8/ 8 (100%)
			PROGRAMMER_7	25/ 27 (92.6%)	24/ 27 (88.9%)
			PROGRAMMER_8	17/ 17 (100%)	16/ 17 (94.1%)
			PROGRAMMER_9	7/ 7 (100%)	7/ 7 (100%)
	FIGURE	EFFICACY	PROGRAMMER_10	8/ 11 (72.7%)	9/ 11 (81.8%)
			PROGRAMMER_8	7/ 7 (100%)	3/ 7 (42.9%)
		SAFETY	PROGRAMMER_6	1/ 1 (100%)	3/ 1 (100%)
			PROGRAMMER_7	3/ 3 (100%)	1/ 3 (100%)
			PROGRAMMER_8	1/ 1 (100%)	1/ 1 (100%)
		TOTAL	PROGRAMMER_10	8/ 11 (72.7%)	9/ 11 (81.8%)
			PROGRAMMER_6	1/ 1 (100%)	3/ 1 (100%)
			PROGRAMMER_7	3/ 3 (100%)	4/ 3 (50.0%)
			PROGRAMMER_8	8/ 8 (100%)	4/ 8 (50.0%)
2	DATASET	BASELINE	PROGRAMMER_10	2/ 1 (100%)	1/ 1 (100%)
			PROGRAMMER_7	2/ 2 (100%)	1/ 2 (50.0%)
		EFFICACY	PROGRAMMER_5	2/ 2 (100%)	2/ 2 (100%)
			PROGRAMMER_8	3/ 3 (100%)	2/ 3 (66.7%)
		SAFETY	PROGRAMMER_10	1/ 1 (100%)	3/ 1 (100%)
			PROGRAMMER_7	2/ 3 (66.7%)	3/ 3 (100%)
		TOTAL	PROGRAMMER_10	1/ 2 (50.0%)	1/ 2 (50.0%)
			PROGRAMMER_5	2/ 2 (100%)	2/ 2 (100%)
			PROGRAMMER_7	4/ 5 (80.0%)	4/ 5 (80.0%)
			PROGRAMMER_8	3/ 3 (100%)	2/ 3 (66.7%)
	TABLE	BASELINE	PROGRAMMER_7	6/ 8 (75.0%)	7/ 8 (87.5%)
		EFFICACY	PROGRAMMER_10	13/ 13 (100%)	12/ 13 (92.3%)

```
                                                       VALIDATION                      PRODUCTION            VALIDATION
PRIORITY   CATEGORY 1            CATEGORY 2            PROGRAMMER                      STATUS                STATUS
---------------------------------------------------------------------------------------------------------------------------------

        2  TABLE                 EFFICACY              PROGRAMMER_3                      8/   8 ( 100%)        8/   8 ( 100%)
                                                       PROGRAMMER_5                     18/  19 (94.7%)       19/  19 ( 100%)
                                                       PROGRAMMER_8                     18/  21 (85.7%)       21/  21 ( 100%)
                                                       PROGRAMMER_9                      5/   5 ( 100%)        5/   5 ( 100%)
                                 SAFETY                PROGRAMMER_6                      1/   1 ( 100%)       21/   1 (87.5%)
                                                       PROGRAMMER_7                     22/  24 (91.7%)        2/  24 ( 100%)
                                                       PROGRAMMER_8                      2/   2 ( 100%)        2/   2 ( 100%)
                                 TOTAL                 PROGRAMMER_10                    13/  13 ( 100%)       12/  13 (92.3%)
                                                       PROGRAMMER_3                      8/   8 ( 100%)        8/   8 ( 100%)
                                                       PROGRAMMER_5                     18/  19 (94.7%)       19/  19 ( 100%)
                                                       PROGRAMMER_6                      1/   1 ( 100%)       28/   1 (87.5%)
                                                       PROGRAMMER_7                     28/  32 (87.5%)       23/  32 ( 100%)
                                                       PROGRAMMER_8                     20/  23 (87.0%)        5/  23 ( 100%)
                                                       PROGRAMMER_9                      5/   5 ( 100%)        5/   5 ( 100%)

           FIGURE                EFFICACY              PROGRAMMER_10                     1/   1 ( 100%)        1/   1 ( 100%)
                                                       PROGRAMMER_3                     12/  12 ( 100%)        9/  12 (75.0%)
                                                       PROGRAMMER_8                     11/  14 (78.6%)       10/  14 (71.4%)
                                 SAFETY                PROGRAMMER_4                      3/   3 ( 100%)        3/   3 ( 100%)
                                                       PROGRAMMER_5                      4/   4 ( 100%)        4/   4 ( 100%)
                                                       PROGRAMMER_8                      3/   4 (75.0%)        3/   4 (75.0%)
                                 TOTAL                 PROGRAMMER_10                     1/   1 ( 100%)        1/   1 ( 100%)
                                                       PROGRAMMER_3                     12/  12 ( 100%)        9/  12 (75.0%)
                                                       PROGRAMMER_4                      3/   3 ( 100%)        3/   3 ( 100%)
                                                       PROGRAMMER_5                      4/   4 ( 100%)        4/   4 ( 100%)
                                                       PROGRAMMER_8                     14/  18 (77.8%)       13/  18 (72.2%)

        3  LISTING               BASELINE              Not Applicable                    0/  16 ( 0.0%)       NA
                                 EFFICACY              Not Applicable                    0/  15 ( 0.0%)       NA
                                 SAFETY                Not Applicable                    0/  17 ( 0.0%)       NA
                                 TOTAL                 Not Applicable                    0/  48 ( 0.0%)       NA
```

Table 6 below illustrates some information about the contents of a file. A file, as defined in SMART, includes any SAS program that can generate analysis datasets, TFLs, or other useful information. The contents of a file depend on the business requirements and are classified with the file item (i.e. log check, macro used, execution date/time and source datasets, etc.). Only one type of file (analysis dataset) and two files (adsl.sas and ds0.sas) are shown in Table 6. With this information, a project leader can focus on what he or she is specifically interested in to see if any additional checks are necessary (i.e. Have source data been updated or have macros been revised, but the program has not rerun yet? Whether do all analysis datasets execute in correct order?).

Table 6 Contents of Files from PRX Metadata

```
                                            Reported on 14MAR2014/16:33

Type              Name      Item              Result
------------------------------------------------------------------------------------------------------------------------------

Analysis Dataset  adsl      Log Check         WARNING: Unable to copy SASUSER registry to WORK registry. Because of this, you
                                              will not see registry customiz
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                            Macro Used         /SASDATA/cars/prod/xxxxxx/macros/attrib.sas
                                               /SASDATA/cars/prod/xxxxxx/macros/delexver.sas
                                               /SASDATA/cars/prod/xxxxxx/macros/dsetproc.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/left.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/trim.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/verify.sas
                            Ran Date          05:04 Thursday, February 13, 2014
                            Source Dataset    random
                                              randkit
                                              kitsched
                                              random
                                              randkit
                                              kitsched
                                              subject
                                              sf
                                              dm
                                              ic
                                              ex
                                              eot
                                              eos
                                              im
                                              pc
                                              ie
                                              random
                                              ex
                                              ptga
                                              joint68
                                              ex
                                              imedcd
                                              ic

Type              Name      Item              Result
-----------------------------------------------------------------------------------------------------------------------------

Analysis Dataset  ds0       Log Check         WARNING: Unable to copy SASUSER registry to WORK registry. Because of this, you
                                              will not see registry customiz
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                                              NOTE: Input data set is empty.
                            Macro Used         /SASDATA/cars/prod/xxxxxx/macros/attrib.sas
                                               /SASDATA/cars/prod/xxxxxx/macros/dsetproc.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/left.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/trim.sas
                                               /apps/SAS/SASFoundation/9.3/sasautos/verify.sas
                            Ran Date          07:26 Thursday, February 6, 2014
                            Source Dataset    ie
                                              random
                                              ex
                                              ptga
                                              joint68
                                              ex
                                              imedcd
                                              ic
```

CONCLUSION

With the arrival of big data in clinical trials, data mining and text analytics are becoming increasingly important for clinical data analysis and presentation. In addition, due to the acceleration and advancement of standardization by CDASH/SDTM/ADaM(6) rapidly, competent management of programming resources, progresses and deliveries is becoming increasingly critical for the success and efficiency of programming activities geared towards the analysis and reporting of clinical studies. This paper introduces data mining, with a focus on text analytics, and presents SMART as a toolkit to face this great and inevitable challenge of data mining, hopefully allowing the efficient and dynamic management of programming activities.

REFERENCES

1. Zhouming (Victor) Sun, *Simultaneous Monitoring of Analysis and Reporting Toolkit (SMART): A New Approach for Improving Programming Management* Abstract submitted for PharmaSUG2014 – USA

2. Zhouming (Victor) Sun, *Reliability-Based Method for Stability of Mine Entry Design and Evaluation* 2000 Ph.D. Dissertation submitted to the College of Engineering and Mineral Resources at West Virginia University. http://wvuscholar.wvu.edu:8881//exlibris/dtl/d3_1/apache_media/L2V4bGlicmlzL2R0bC9kM18xL2FwYWNoZV9tZWRpYS80OTUz.pdf

3. SAS Perl Regular Expressions Tip Sheet http://support.sas.com/rnd/base/datastep/perl regexp/regexp-tip-sheet.pdf

4. SAS Hash Object Tip Sheet

5. Eric Vandervort, *Knowing When to Start, Where You Are, and How Far You Need to Go: Customized Software Tracks Project Workflow, Deliverables, and Communication,* SAS Global Forum 2013 http://support.sas.com/resources/papers/proceedings13/013-2013.pdf

6. Zhouming(Victor) Sun, *CDISC: Why SAS® Programmers Need to Know* 2003 PharmaSUG https://www.google.com/#q=Victor+Sun%2C+Why+SAS%C2%AE+Programmers+Need+to+Know

CONTACT INFORMATION

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

Name:	Zhouming (Victor) Sun
Enterprise:	Medimmune
Address:	1 Medimmune Way
City, State ZIP:	Gaithersburg, MD 20878
Work Phone:	301-398-2129
Fax:	
E-mail:	sunz@medimmune.com
Web:	