PharmaSUG 2014 - Paper DS10

Developing ADaM Specifications to Embrace Define-XML 2.0.0 Requirements

Lin Yan, Celgene Corporation, 300 Connell Drive, Berkeley Heights, NJ

ABSTRACT

The CDISC Define-XML 2.0.0 standard was released in March 2013. The new standard has quite a few new features and requirements. In order to adopt the new standards, sponsors need make changes accordingly in the ADaM specification files, which are usually used as programming guidance and as the basis of the generation of ADaM define-XML as well. This paper will discuss the new features and requirements in the Define-XML 2.0.0 that have direct impact to ADaM specifications. Also, this paper will use examples to illustrate what changes/enhancements in the ADaM specifications should be added to meet the requirements specified in Define-XML 2.0.0.

INTRODUCTION

Sponsors often use ADaM dataset specifications as ADaM programming guidance. The specification file is usually in Excel format and include dataset labels, dataset names, variable names, variable lengths, and variables types, controlled terminologies, etc. The specifications are usually used as the basis for the generation of ADaM Define.xml and Define.pdf.

Define-XML 2.0.0 was released in March 2013 by CDISC and now is accepted by FDA as the standards for study data submissions. The new standards have quite a few new features and requirements. For example, the new standards support result metadata and allow the omission of decode if the code is actually a enumerated item of controlled terminology. For CDISC controlled terminology terms, the C-code is required to be provided. For parameter value level metadata, a Where-clause has been introduced. This addition allows the sponsors to describe the value level meta of multiple PARAMCD's in one line. The new standards support clear links to external files such SAP or other supplementary documents. Most of these changes in define file requirements/ components imply the changes in ADaM specifications. However, it takes a lot of time and effort for sponsors to create valid ADaM Define-XML that is compliant to the new standards.

The changes from Define-XML 1.0 to Define-XML 2.0.0 are quite comprehensive. This paper will go through the changes that have direct impact to the ADaM specifications. This paper will use various examples (some examples from the Define-XML v2 package posted in CDISC website) to show how to make the corresponding changes in the ADaM specifications so that the information the components required by Define-XML are provided.

1. New features in Define-XML 2.0.0

The Define-XML 2.0.0, based on the CDISC Operational Data Model (ODM) version 1.3.2, is a major revision of the previous define xml standards Define-XML 1.0 which was named as Case Report Tabulation Data Definition Specification version 1.0.0 (CRT-DDS).

There are lot of new features and advanced techniques in the Define-XML v2.0.0. The following section lists only those major new features and key changes that have direct impact to ADaM metadata in the Define-XML v2.0.0.

1.1 User friendly features for parameter level metadata with the introduction of Where clause

Table 1.1.1 Value level metadata (ADaM followed SDTM format) in Define-XML v1.0

Source Variable	Value	Label	Type	Controlled Terms or Formats	Origin	Role	Source/Derivation
PARAMCD	ALB	Albumin (g/L)	integer		SDTM.LB		SDTM.LB.LBSTRESN

PARAMCD	BILI	Bilirubin (umol/L)	float		SDTM.LB		SDTM.LB.LBSTRESN
PARAMCD	ALT	Alanine Aminotransferase (U/L)	integer		SDTM.LB		SDTM.LB.LBSTRESN

In the Value Level Metadata in the previous version, each value of PARAMCD should be enumerated even some of them share the same metadata. In the Define-XML 2.0.0, a Where clause is introduced. A group of PARAMCDs that shares the same metadata can be grouped together and are displayed together as shown in table 1.2.

Table 1.1.2 Value level metadata in Define-XML v2.0.0

Variable	Where	Type	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
AVAL	PARAMCD IN ("ALB" (Albumin (g/L)), "ALT" (Alanine Aminotransferase (U/L)))	integer	8		Assigned	LB.LBSTRESN where LBTESTCD=PARAMCD
AVAL	PARAMCD EQ BILI (Bilirubin)	float	3		Assigned	LB.LBSTRESN where LBTESTCD=PARAMCD

The Where clause can even handle more complicated situation where more than one variable is involved, which will be discussed in section 2 of the paper.

1.2 Efficient and clear way to support CDISC Controlled Terminology

In Define 1.0, the Code Lists are always presented in pair, code and decode(table 1.2.1). In most of cases of enumerated terms, the identical values of codes and decodes are redundant. In addition, with the standardization to semantic level, CDISC/NCI defined more and more controlled terms. Those standardized terms (C-codes) are not reflected in the codelist metadata.

Table 1.2.1 Controlled terminology metadata in Define-XML v1.0

Coded Value	Decode
DOSE INCREASED	DOSE INCREASED
DOSE NOT CHANGED	DOSE NOT CHANGED
DOSE REDUCED	DOSE REDUCED

The enhancements in Define-XML v2.0.0 simplify the study controlled terminology metadata by allowing the controlled terms displayed without the redundant “decode” part. In addition, the CDISC/NCI C-codes are able to be displayed(if applicable). See table 1.2.2.

Table 1.2.2 Controlled terminology metadata in Define-XML v2.0.0

Permitted Value (Code)
DOSE NOT CHANGED [*C49504*]
DOSE REDUCED [*C49505*]
DRUG INTERRUPTED [*C49501*]

1.3 Enhanced support for document links

The Define-XML v2.0.0 adds support to explicit links to external reference documents such that derivation rules, SAPs, data reviewer’s guide, and dataset generation programs. The links can precisely link to the PDF page numbers, page ranges, or named destinations.

2. Develop ADaM specifications to meet Define-XML 2.0.0 requirements

In this section, examples will be given to illustrate what changes in ADaM specifications are made in order to embrace the new features and requirements.

2.1 Components in ADaM Define-XML file

Under Define-XML 2.0.0, in addition to external supporting documents, such as data reviewer's guide, complex analysis derivation rules (user defined), there are the following defined sections in a define file.

Dataset metadata
Variable metadata
Controlled Terminology metadata
Parameter Value Level metadata
Analysis Derivations
Comments
The analysis results metadata, as an extension to Define-XML 2.0.0, can be added as optional.

2.2 Dataset level metadata

The dataset level metadata defined in the current ADaM documents[1] is displayed as in the Table 2.1 with attributes of Dataset, Description, Class, Structure, Purpose, Keys, Location, Documentation

Table 2.2.1 Sample dataset level metadata

Dataset	Description	Class	Structure	Purpose	Keys	Location	Documentation
ADSL	Subject-Level Analysis	SUBJECT LEVEL ANALYSIS DATASET	one record per subject	Analysis	USUBJID	adsl.xpt	Screen Failures are excluded since they are not needed for this study analysis
ADQSADAS	ADAS-Cog Analysis	BASIC DATA STRUCTURE	One record per subject per parameter per analysis visit per analysis date	Analysis	USUBJID, PARAMCD, AVISIT, ADT	adqsadas.xpt	See referenced dataset creation program [adqsadas.sas] and Analysis Data Reviewer's Guide, Section 2.1 [Analysis Data Reviewer's Guide]

The description of a dataset will be used as the label of the dataset which is required to be within 40 characters.

Currently according to CDISC controlled terminology, there are three types of classes, SUBJECT LEVEL ANALYSIS DATASET, BASIC DATA STRUCTURE, and ADAM OTHER. The last type is the class for a dataset that follows ADaM fundamental principles and is compliant with dataset/variable naming conventions but does not belong to any of the first two types. A new data structure type for occurrence analysis may be added to the class in the future.

So any dataset, if following ADaM fundamental principles and compliant with dataset/variable naming conventions is considered as an ADaM dataset, regardless the data source. This implies a dataset is still an ADaM dataset even if the source data is not SDTM data. Of course, it is encouraged to build ADaM datasets based on SDTM data, which can better keep the traceability.

Under Define 2.0.0, enhanced links in Documentation allow sponsors to provide information in external files such as a SAS program or a data reviewer's guide. According to the FDA requirements, the SAS file can be provided as ASCII text file or PDF file and stored in the folders "programs". The data reviewer's guide or other supplement documents are in the same folder of Define-XML[2]. The content in each row of Documentation has an entry in Comments metadata section in the Define-XML version 2 examples.

To create the links correctly, the Comments specifications for the Documentation part in Table 2.2.1 could be set as shown in Table 2.2.2.

Table 2.2.2 Comments metadata for the documentation

CommentOID	Description	LinkFileID
COM.ADSL	Screen Failures are excluded since they are not needed for this study analysis	
COM.ADQSADAS	See referenced dataset creation program [adqsadas.sas] and Analysis Data Reviewer's Guide, Section 2.1 (Analysis Data Reviewer's Guide.pdf) [Analysis Data Reviewer's Guide]	[LF.ADQSADAS.PGM] [LF.ADRG]

The values of CommentOID connect the descriptions to the datasets, ADSL and ADQSADAS. The linkage can be provided for the long text (detailed instructions) in a separate file as shown in Table 2.2.3. The third column LinkFileID is for XML generation programming not for review display.

Table 2.2.3 Sample of links for the documentation in dataset specifications

LinkFileID	FileName	Page	FilePath
LF.ADQSADAS.PGM	adqsadas-sas.txt		../programs/ADQSADAS-sas.txt
LF.ADRG	analysis-data-reviewers-guide.pdf	Section2.1	

In Table 2.2.3, the first row is for the link to the program (ADQSADAS-sas.txt) which is placed under a relative path “programs”. The second row is for the link to the data reviewer’s guide and specifically section 2.1 which is shown in column Page.

2.3 Variable level metadata

The variable level metadata under Define-XML v2.0.0 can be displayed as shown in Table 2.3.1 with attributes of Variable, Label, Type, Length/Display Format, Controlled Terms or Format, and Source/Derivation/Comment.

Table 2.3.1 Sample of variable level metadata

Variable	Label	Type	Length / Display Format	Controlled Terms or Format	Source/Derivation/Comment
USUBJID	Unique Subject Identifier	text	11		Predecessor: DM.USUBJID
AGEGR1	Pooled Age Group 1	text	5	["<65", "65-80", ">80"] <AGEGR1>	Derived: Character variable derived from ADSL.AGEGR1N
AGEGR1N	Pooled Age Group 1 (N)	integer	8	["1" = "<65", "2" = "65-80", "3" = ">80"] <AGEGR1N>	Assigned: AGEGR1 = 1 if AGE <65. AGEGR1 = 2 if AGE 65-80. AGEGR1 = 3 if AGE >80.
TRTSDT	Date of First Exposure to Treatment	integer	date9.		Derived: SV.SVSTDTC when SV.VISITNUM=3, converted to SAS date

RFSTDTC	Subject Reference Start Date/Time	datetime	20	ISO8601	Predecessor: DM.RFSTDTC
DCREASCD	Reason for Discontinuation	text	18	DISCREAS	Derived: Grouping of DCDECOD values

The data type in the third column is CDISC ODM data type, not SAS data type. Six ODM data types were defined in the previous version of define-xml: text, integer, float, datetime, date, and time, among which only types integer and float are numeric counterparts in SAS. Others are corresponding character types in SAS. In the new define-xml v 2.0.0, four more ODM data types are added. These newly-added ODM data types are partialDate, partialTime, partialDatetime, incompleteDatetime, and durationDatetime. All of new ones correspond to character types in SAS. In variable level metadata, for character datetime variable RFSTDTC compliant with ISO8601 standards it has type of “datetime”.

The Length column is required if the data type is “text”, “integer”, or “float”. The value of the Length itself is an integer. Display format is for data type of float and other numeric date values. The Display format has higher priority than the Length, i.e for a date variable, it is Display format “date9” which will be displayed in the column instead of the length of the variable. For a float type variable, both Length and Significant Digits must be specified.

The Controlled Terms or Format column is used to present controlled terms or codelist if available. All the controlled terms or codelist will have an entry in the codelist metadata section. However if the number of the codelist items is small, less than 6, for instance, the items can be displayed in the variable metadata in addition to the codelist metadata such as ["<65", "65-80", ">80"] for variable AGEGR1. In that situation, brackets (“[]”) and (“<>”) are used to differentiate the numerated items and the name of the code list. Variable RFSTDTC follows international formatting standards ISO8601 as shown in the Controlled Terms or Format column. However ISO8601 does not required to be put in external code list as it is not a controlled terminology. If the codelist has a lot of items, say more than 5, it should be put in codelist section entirely to keep the variable level metadata reader-friendly.

The Source/Derivation/Comment is a critical part for ADaM specifications. Under define-XML 2.0.0, the source is explicitly classified as Predecessor, Derived, and Assigned.

For a variable directly copied from a variable in another dataset, the variable name together with the dataset name is indicated after the reserved key word “Predecessor” in the column Source/Derivation/Comment., e.g. DM.USUBJID or ADSL.TRT01P.

When a data specification designer gives values to a variable, kind of arbitrarily, the variable is “Assigned”, such as AGEGR1N in the table 2.3.1. Most of the coded terms, such as PARAMCD, AVISITN, DTYPE, are “Assigned” in source.

For “Assigned” variables in variable level metadata, extra explanations could be provided in section Comments (see table 2.3.2) as they are in Define-XML v2 examples. An external file can be referenced for long comments (refer to Table 2.2.3).

Table 2.3.2 Sample of “Assigned” variable (variable level metadata) in comments section

CommentOID	Description

COM.ADSL.AGEGR1N	AGEGR1N = 1 if AGE <65.
	AGEGR1N = 2 if AGE 65-80.
	AGEGR1N = 3 if AGE >80.

If a variable or the value of a variable is not collected but is derived from other variables through an algorithm, it is “Derived”. In Table 2.3.1, the values of variable AGEGR1 are derived from variable AGEGR1N. For any derived variable, the algorithm must be provided. The algorithm could be included in the computational Algorithms section as well. For long algorithms, external documents can be referenced. See table 2.3.3 for an example.

Table 2.3.3 Sample of "Derived" variable (variable level metadata) in Analysis Derivations section

Method	Type	Description	LinkFileID
CM.ADSL.AGEGR1	Computation	Character variable derived from ADSL.AGEGR1N	
CM.ADQSADAS.ANL01FL	Computation	ANL01FL = "Y" for the last observation at an analysis visit. Null otherwise. (complexalgorithms.pdf) [Complex Algorithms]	LF.ComplexAlgorithms

The algorithm for derived variable AGEGR1 is in column Description. For a complicated, long algorithm, it can be placed in a separate external PDF file and referenced. For example, the algorithm for ANL01FL has detailed long text which is in the complex algorithms file (complexalgorithms.pdf) and the link file can be set as Table 2.3.4 below. In the example below. In the example below, it is supposed that the file complexalgorithms.pdf is stored in the same folder of the define-xml. Otherwise, the relative file path should be specified.

Table 2.3.4 Sample of links for the documentation in variable level specifications

LinkFileID	FileName	Page	FilePath
LF.ComplexAlgorithms	ComplexAlgorithms		

The ADaM variable level specifications can be set as Table 2.3.5

Table 2.3.5 Sample of ADaM variable level specifications

Variable	Label	Type	Length / Display Format	Controlled Terms or Format Name	Source/Derivation/Comment	Parameter Value List
USUBJID	Unique Subject Identifier	text	11		Predecessor: DM.USUBJID	
AGEGR1	Pooled Age Group 1	text	5	["<65", "65-80", ">80"] <AGEGR1>	Derived: Character variable derived from ADSL.AGEGR1N	
DCREASCD	Reason for Discontinuation	text	18	DISCREAS		
AVAL	Analysis Value	integer	8		Derivations are described per parameter in the parameter value level metadata	AVAL

For explicitly links to parameter value level metadata of a variable, an indicator for the parameter value level metadata can be specified in the variable level metadata in column Parameter Value List as shown in Table 2.3.5. Variable AVAL has parameter value level metadata and the column Parameter Value List is filled with the name, AVAL, of the parameter value level metadata.

2.4 Parameter value level metadata

If a metadata attribute for a variable depends on the value of another variable (PARAM, for instance), parameter value level metadata is required to further describe the differences. The typical situation is in BDS datasets.

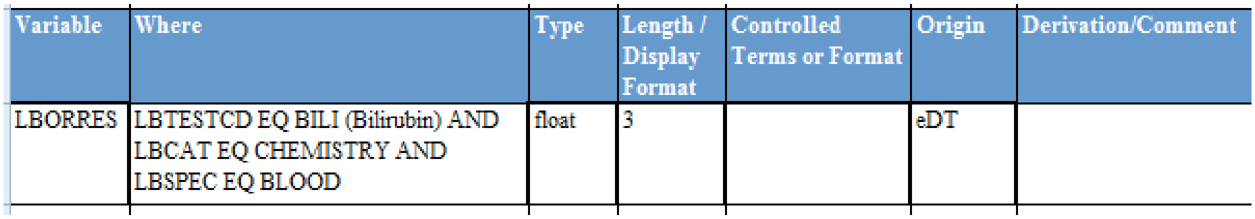
The display of parameter value level metadata is shown in table 2.4.1

Table 2.4.1 Sample of parameter value level metadata

Variable	Where	Type	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
AVAL	PARAMCD NE (ACTOT (Adas-Cog(11) Subscore)	integer	8		Derived	QS.QSSTRESN where QSTESTCD=PARAMCD
AVAL	PARAMCD EQ ACTOT (Adas-Cog(11) Subscore) AND DTYPE(Derivation Type) EQ Y	integer	8		Derived	Sum of ADAS scores for items 1 to 10, see Analysis Data Reviewers Guide (Page 3) for details on adjusting for missing values.

The new Define-XML created a Where clause which allows grouped items (PARAMCD) in one entry. This feature is a big user-friendly improvement which helps the define file creator avoid enumerate the cases one by one. The Where clause can even handle more complicated situations where more than one variable or a subject-level dataset is involved as well. See the excerpt from the CDSIC SDTM define-xml example in table 2.3.2.

Table 2.4.2 Sample of parameter value level metadata with complicated selection condition



Variable	Where	Type	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
LBORRES	LBTESTCD EQ BILI (Bilirubin) AND LBCAT EQ CHEMISTRY AND LBSPEC EQ BLOOD	float	3		eDT	

In table 2.4.2, variables LBTESTCD, LBCAT and LBSPEC are included in the Where clause.

Those parameter value level metadata core items, such as Type, Length/Display Format, Controlled Terms or Format, are identical to those in variable level metadata. The values in the detailed parameter value level metadata supersede the values in the variable level metadata. For example, if the maximum length for variable AVAL is 8 in variable level metadata but the length is defined as 3 in the parameter value level metadata for the values of a particular parameter, the length of the value is considered as 3 instead of 8. The length is used in XML file not in SAS dataset. As we all know that a SAS variable cannot have more than one type or length.

Same as those “Assigned” and “Derived” variables in variable level metadata, an assigned variable could have an entry Comments section and a derived in Analysis Derivations section respectively. See table

Table 2.4.3 Sample for “Assigned” variable (parameter level metadata) in Comments section

CommentOID	Description
COM.ADQSADAS.QSSEQ.ACTOT	Set QSSEQ to missing for post baseline records. Set to QS.QSSEQ where QS.VISIT=BASELINE and QS.QSTESTCD=ACTOT.

Table 2.4.4 Sample for “Derived” variable (parameter level metadata) in Analysis Derivations section

Method	Type	Description
CM.ADQSADAS.AVAL.ACITM01-ACITM14	Computation	QS.QSSTRESN where QSTESTCD=PARAMCD
CM.ADQSADAS.AVAL.ACTOT	Computation	Sum of ADAS scores for items 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, and 14, see Analysis Data Reviewers Guide (Page 3) for details on adjusting for missing values. Analysis Data Reviewer's Guide (analysis-data-reviewers-guide.pdf)

The long links are placed in external files and can be linked just as those in the variable level metadata links (table 2.3.4).

The ADaM parameter value-level specifications can be set as the table 2.4.5

Table 2.4.5 Sample of ADaM parameter value-level specifications

Parameter Value List	Source Variable	Type	Length / Display Format	Controlled Terms or Format	Origin	Derivation
ADQSADAS_AVAL	AVAL	integer	8		Derived	QS.QSSTRESN where QSTESTCD=PARAMCD
ADQSADAS_AVAL	AVAL	integer	8		Derived	Sum of ADAS scores for items 1 and 11, see Analysis Data Reviewers Guide (Page 3) for details on adjusting for missing values.
Where Variable1	Where Condition1	Comparator1	Where Variable2	Where Condition2	Comparator2	
PARAMCD	ACT01, ACT02, … , ACT11	IN	ANL01FL	Y	EQ	
PARAMCD	ACTOT	EQ	AVISIT	LAST	EQ	

The values used in Where clause are explicitly specified in the columns “Where Variable1”, “Where Condition1”, and “Comparator1”. Additional conditions can be expended in other set of columns just as shown in the table 2.4.5.

2.5 Controlled Terminology metadata

For controller terminology, it is required that all used items in a study be provided in the controlled terminology metadata [3]. The items can be displayed in the variable level metadata but still required to be placed in the controlled terminology metadata part.

The display of the controlled terminology metadata is shown in table 2.5.1

Table 2.5.1 Sample of controlled terminology metadata

Derivation Type [CL.DTYPE, C81224]	
Permitted Value (Code)	**Display Value (Decode)**
LOCF [*C81198*]	Last Observation Carried Forward

DURDISC [CL.DURDISC]	
Permitted Value (Code)	
<12	
>=12	

Under Define-XML 2.0.0, for controlled terms with an enumeration of allowed values only, decode is not required, e.g. DURDISC in table 2.3, only two levels, <12 and >=12.

If a controlled terms or code list item is defined by CDISC (collaboration with National Cancer Institute's Enterprise Vocabulary System), the name of the Codelist in the define file must match that in the CDISC controlled terminology list. Also the C-code of the item is added. For example, LOCF, defined by CDISC, the C-code C81198 is added to identify easily the item in the CDISC code list system.

If a item defined by a sponsor is added to an extensible CDISC defined code list, the item should be flagged as extended value without C-code.

In Define-XML 2.0.0, numeric significance can be described using the Rank attribute, e.g. a list of values "Low", "Medium", and "High". It is not designated for the display order which can be determined by the creator the define-XML either based on the order in the analysis output or the frequency of the occurrences.

Based on the requirements, an ADaM specifications for the controlled terminology can be set as the follows in table 2.5.2.

Table 2.5.2 Sample ADaM specifications for controlled terminology metadata

Code List	Code List Code	Code Full Name	Permitted Value (Code)	Display Value (Decode)	External Code ID	Rank
AGEGR1N		AGEGR1N	1	<65		
AGEGR2N		AGEGR2N	2	65-80		
AGEGR3N		AGEGR3N	3	>80		
DATEFL	C81223	Date Imputation Flag	D		C81212	
DATEFL	C81223	Date Imputation Flag	M		C81211	
DATEFL	C81223	Date Imputation Flag	Y		C81210	

The second column, Code List Code, provides the C-code for upper level code list. The low level C-code is listed in column External Code ID.

CONCLUSION

With the CDISC new define file standards, Define-XML 2.0.0, becomes available, ADaM specifications developers need to comprehend the changes and requirements and make corresponding changes in the working file of ADaM specifications. The changes/new features in parameter value level metadata, code list, and links to external files have direct impact to the ADaM specifications. The possible changes in the ADaM specifications are illustrated.

REFERENCES

[1] Analysis Data Model (ADaM) v2.1, http://www.cdisc.org/adam
Analysis Data Model (ADaM) Implementation Guide v1.0, http://www.cdisc.org/adam
Analysis Data Model (ADaM) Examples in Commonly Used Statistical Analysis Methods v1.0, http://www.cdisc.org/adam
Analysis Data Model (ADaM) Data Structure for Adverse Event Analysisv1.0, http://www.cdisc.org/adam
The ADaM Basic Data Structure for Time-to-Event Analyses v1.0, http://www.cdisc.org/adam
[2] Study Data Specifications v2.0, http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM312964.pdf
[3] CDISC Define-XML Specification Version 2.0, http://www.cdisc.org/define-xml

ACKNOWLEDGMENTS

The author thanks Celgene BDO programming management team for the support.

RECOMMENDED READING

- Analysis Data Model (ADaM) v2.1, http://www.cdisc.org/adam
- Analysis Data Model (ADaM) Implementation Guide v1.0, http://www.cdisc.org/adam
- Analysis Data Model (ADaM) Examples in Commonly Used Statistical Analysis Methods v1.0, http://www.cdisc.org/adam
- CDISC Define-XML Specification Version 2.0, http://www.cdisc.org/define-xml
- Define-XML v2 – What’s New, Lex Jansen, PhUSE 2013

CONTACT INFORMATION

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

Name: Lin Yan
Enterprise: Celgene Corporation
Address: 300 Connell Drive
City, State ZIP: Berkeley Heights, NJ 07922-2805
E-mail: lyan@celgene.com