

# Define2Validate - Validate CDISC Dataset-XML with corresponding Define-XML metadata

Key message: We developed an **open-source tool** that validates Dataset-XML against the rules or metadata defined in Define-XML.

Masafumi Okada M.D., Ph.D.  
University Hospital Medical Information Network Research Center,  
(UMIN), University of Tokyo, Japan.  
Contact: sokada-tuk@umin.ac.jp



## Background

**Define-XML** defines the metadata of CDISC dataset. However, though there are some tools to validate dataset against SDTM standards or regulatory validation rules, there is still **no reference tool** to validate datasets against the metadata and business rules that is defined by corresponding **Define-XML**.

## Objective

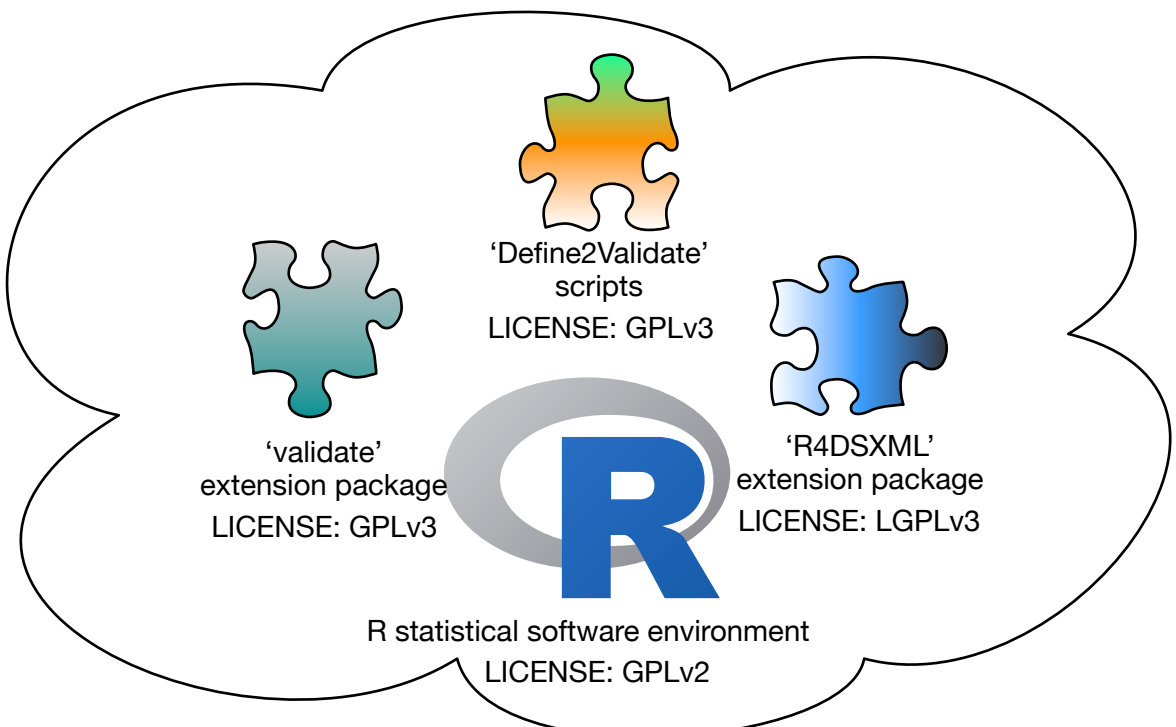
To prove the concept of validating dataset against the metadata defined in Define-XML, we developped an open-source tool to validate datasets against corresponding Define-XML.

## Methods

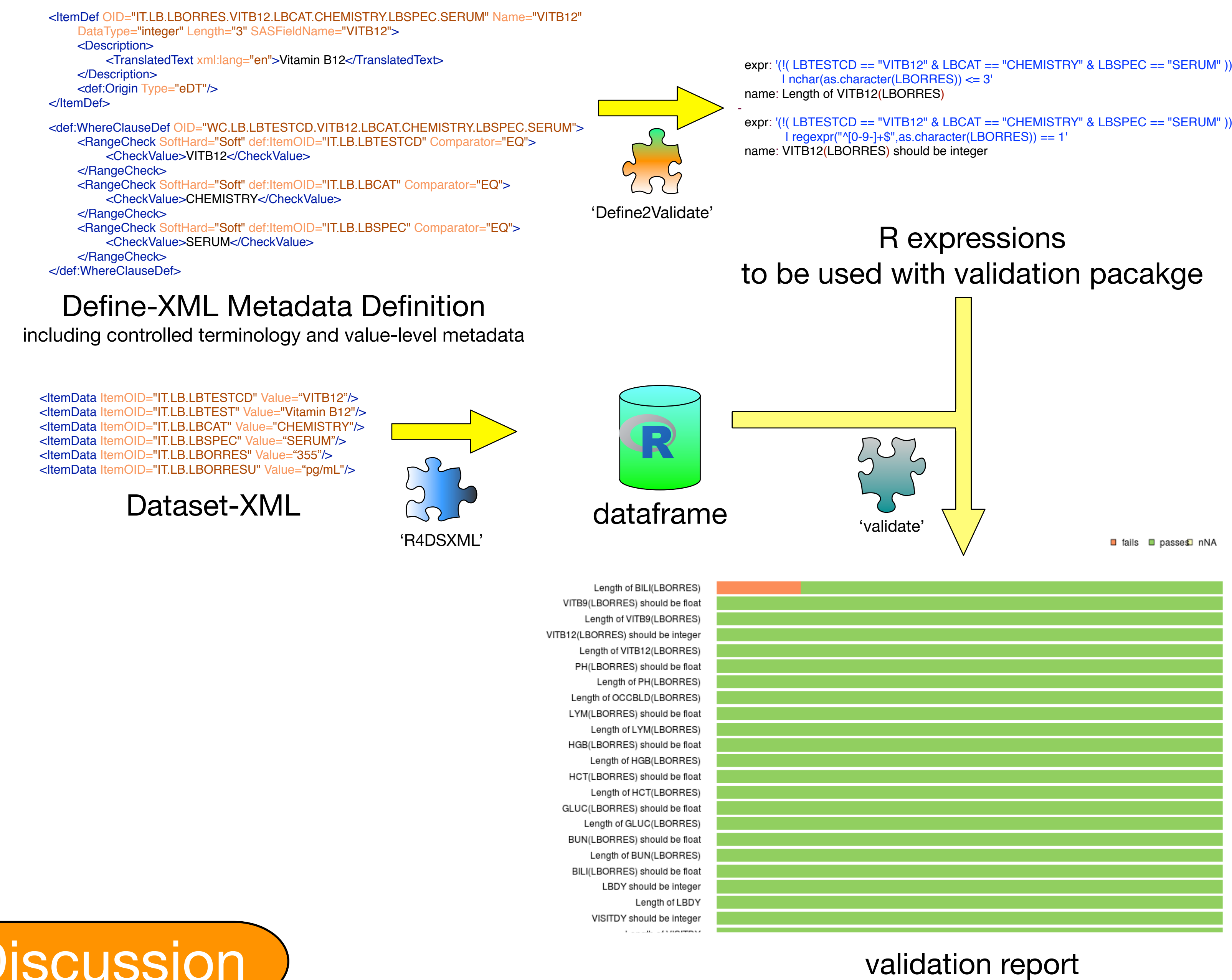
We implemented a validation tool on the R statistical programming environment[1]. We named the tool as “Define2Validate”. To read the content of Define-XML and Dataset-XML, we adopted a R package “R4DSXML” written by Ippei Akiya[2]. Define2Validate reads Define-XML, then converts the rules to which the dataset conforms into a set of R expressions. To perform tests against the dataset with the expressions, we adopted a R package “validate” written by Mark van der Loo[3]. We also adopted a R package “testthat” written by Hadley Wickham[4] to implement Define2Validate. All tools are licensed under open-source licenses.

## Results

With Define2Validate, we could validate the dataset against the metadata defined in Define-XML and generate a report of conformance. Though currently not all business rules are implemented in Define2Validate, it supports variable-level metadata, value-level metadata, and controlled terminology.



All components are open source software



Source code:

<https://github.com/mokjpn/Define2Validate>

Live demonstration available:

<https://cuda.umin.ac.jp/s/Define2ValidateDemo/>

Define2Validate Demonstration

Choose your Define-XML v2.0

Browse... Upload datasets

Choose your Dataset-XML v1.0

Browse... Upload datasets

Set the domain of your Dataset-XML

CM

Validate

Source code

If you find any 'false' in 'Table' tab, click the cell. Then the data with corresponding error will appear in 'Failed Records' tab.

Table	Figure	Failed Records						
Show: 20	entries							
rule	items	passes	fails	nNA	error	warning	expression	Search
Length of DOMAIN	2	2	0	0	false	false	nchar(as.character(DOMAIN)) <= 2	
DOMAIN is mandatory	2	2	0	0	false	false	is.na(DOMAIN)	
DOMAIN should follow coded CL.DOMAIN	0	0	0	0	true	false	as.character(DOMAIN) %in% CT[CL] "OID" == "CL.DOMAIN", "CodedValue"	
Length of CMSEQ	2	2	0	0	false	false	nchar(as.character(CMSEQ)) <= 2	
CMSEQ is mandatory	2	2	0	0	false	false	is.na(CMSEQ)	
CMSEQ should be integer	2	2	0	0	false	false	regexpr("^[0-9]+\$", as.character(CMSEQ)) == 1	
Length of CMTRT	2	2	0	0	false	false	nchar(as.character(CMTRT)) <= 23	
CMTRT is mandatory	2	2	0	0	false	false	is.na(CMTRT)	
Length of CMDOSE	2	2	0	0	false	false	nchar(as.character(CMDOSE)) <= 4	
CMDOSE should be integer	2	2	0	0	false	false	regexpr("^[0-9]+\$", as.character(CMDOSE)) == 1	
Length of CMDOSU	2	2	0	0	false	false	nchar(as.character(CMDOSU)) <= 8	
CMDOSU should follow coded CL.C78417.CMDOSU	0	0	0	0	true	false	as.character(CMDOSU) %in% CT[CL] "OID" == "CL.C78417.CMDOSU", "CodedValue"	
CMSTOTC should be Date	2	2	0	0	false	false	regexpr("^[0-9]{4}-[0-9]{2}-[0-9]{2}\$", as.character(CMSTOTC)) == 1	

Showing 1 to 13 of 13 entries

Previous

1

Next

## Discussion

Define-XML stores metadata to which the dataset should conform. The metadata should be defined at the very early stage of planning a clinical trial. However, currently there is no reference tool to validate the data against the metadata defined in Define-XML. Thus not so many investigators writes Define-XML at the beginning of trials. Indeed, some investigators “generate” Define-XML to match the actual dataset after the data collection. That practice might spoil some philosophy of data and metadata standards. Our tool would help researchers to validate the data against the pre-defined metadata which is described in Define-XML.

## References

1. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
2. Ippei Akiya(2016). R4DSXML package. <https://github.com/DataDrivenInc/R4DSXML>
3. Mark van der Loo(2016). validation package. <https://CRAN.R-project.org/package=validate>
4. Hadley Wickham(2016). testthat package. <https://CRAN.R-project.org/package=testthat>

This work was supported by JSPS KAKENHI Grant Number JP15K15218 and by the Project Promoting Clinical Trials for Development of New Drugs from Japan Agency for Medical Research and development, AMED.