

# Complex Laboratory Data, Strategies and Tools for a Way out of the Maze

Benjamin Szilagyi, Novartis Pharma AG, Basel, Switzerland

Christof Binder, BIOP AG Basel, Switzerland

## ABSTRACT

For programmers assigned to work with laboratory data from a multi center study without a central lab, two reactions are common: Active suppression of past experiences or wet palms. The problems faced with this task are manifold and complex. They include: Different formats of data from the different centers, varying units for the same parameters, normal ranges that a) change over time, b) are dependent on the age and/or gender of the subject, c) are not easily comparable (% vs. counts) between centers, d) missing. To analyze the data a uniform representation across all centers is needed but not easily achieved without considerable data management and programming effort. The aim of this paper is to sketch a framework for programmers to streamline the process from raw data to analysis data sets. The key points of this framework include: Early definition of the format and ways of transmission of the data from the centers to the programmer, a set of modules with clearly defined interfaces, use of conversion and unit dictionaries, standardize internal data structures and appropriate use of PROC SQL. This framework has proven to be efficient to handle a variety of lab data and is saleable to meet future requirements with minimal effort.

Keywords: Laboratory data, conversion, SI-Units, SAS<sup>®</sup> Proc SQL, standardization

## INTRODUCTION

### THE PROBLEM (OR YOUR MILEAGE MAY VARY)

Working with laboratory data in a multi-center study without a central lab can challenge the programmer as the data delivered can be very heterogeneous and inconsistent. Local laboratories each have their own operating procedures that can not be easily standardized. They use different equipment to analyze the samples; the calibration will therefore vary.

Factors like these, especially when unforeseen, may increase data management and programming time tremendously, not only raising overall costs, but potentially compromising data quality.

### LABORATORY VALUES

The naming convention for the individual lab parameters can vary considerably, esp. if different languages are spoken in the participating centers (Haemoglobin, HGB, Haem, etc.). Also units of the measurements can be labeled differently (i.e. /ML vs. 10E\*/L) or be actually different (ng/dL vs. pmol/L).

Country-specific formatting of the data, e.g. Dates (DDMMMYYYY vs. MM/DD/YYYY), separation of thousands place (1000 vs. 1'000), use of a decimal point or comma, are further factors potentially leading to inconsistencies in the raw data.

### NORMAL RANGES

It can be very difficult to obtain complete and correct normal ranges for all parameters. The higher the number of individual laboratories the more time needed to be invested to receive all data required.

A reason for additional complexity can be demographic variations between countries and especially between continents, reflected in age and gender-specific normal and plausibility ranges different from each other (gender specification being neither "female" nor "male" but "not applicable" can occur quite likely in Asian countries!).

### FORMAT OF THE TRANSPORT

The format in which the data is collected and passed on to the programmer varies. There are numerous formats in which the data can be collected, each with its own "features". Plain ASCII text files, spread sheets, databases are the more commonly used ones.

## SOLUTIONS, FINDING A WAY OUT

*"An ounce of prevention is worth a pound of cure." (Benjamin Franklin)*

### 1. INPUT

Some sources of variability are beyond the reach of a programmer. It is quite unlikely that a country changes their date format because of her/him. However the earlier in the process of defining the format and transfer of the data the programmer has a say, the better are the chances that data will arrive in a form that it can be used without huge amount of data management and programming effort. It is therefore essential that the programmer has a clear idea of how the data should be structured and communicate the requirements as early as possible to the CRF designer, the

trial leader and the investigators depending on how early he/she can get involved. This means agreeing on a specification document that details the way the data should be structured and formatted. It may include the acceptable name for a parameter, the units, convention on the date format used, agreements on how repeat samples and repeated analysis of the same samples will be reported etc. The programmer should aim to find a format to transfer the data usable at all sites and provide an example template with complete columns and rows to describe what he/she needs.

This template should be user-friendly and clear. Have a close look at the first transfers to help the laboratory to implement correctly the specifications.

## 2. CONVERSION TO STANDARD

### INPUT NORMALIZATION

Once the data arrived on the programmers desk it should be imported into a convenient data structure to get the uniform presentation needed. Hopefully, the programmer and data manager were successful in implementing a standard format for data-transfer and therefore one single import program can be used for all centers to transfer the raw lab data and normal ranges from any format into SAS datasets. To detail all possible ways data can be read into SAS datasets is beyond the scope of this paper (but see *Andrew T. Kuligowski, SUGI 26, 2001*, as a start). As long as accuracy and uniform formatting across all laboratories is given, the essential requirements are present for efficient coding.

### CONVERSION TO STANDARD

The procedures described below should be programmed within separate modules, each module executing one step towards the final standard laboratory dataset. Interfaces between the modules are clearly defined and standardized. Advantages of modular programming have been discussed in previous papers (*Fuping Peng, PharmaSUG 2003*) and the same basic principles apply for the framework presented here.

Module 1) Translation of parameter names and units, transformation of raw dataset into standard structure  
 First the parameters of the raw data need to be named consistently. If this can not be achieved already at the laboratories, either a dictionary or a format can be used to translate the name. The programmer should once set a standard of "unit naming" (e.g. Ron Haas 2004) and keep it, i.e. "/ML" gets translated into "10E\*/L" or "10\*\*3/L" to "10E3/L". In both cases dictionary or format, the file should be kept externally (i.e. text file for formats or a dataset as dictionary) to be able to reuse and expand it in future studies. This procedure applies for both the units for lab values as well as for normal ranges.

Module 1 provides datasets with standardized parameter and unit names, containing one parameter per observation. This structure enables a seamless merge in the next modules.

#### Module 2a) Laboratory values to SI

For comparing data across laboratories they need to be present in the same unit.

A widely accepted choice is the SI unit for the parameter. It is not difficult but needs an initial effort to set up a dictionary that can be used to do this conversion automatically. As a starting point, present conversion tables on the internet or e-books (see references) could be used. This dictionary will become more detailed with future studies when new conversions get added to the already existing ones. The initial effort can therefore be seen as a useful investment. The dictionary structure could look as displayed in table 1.

par_name	rec_unit	si_unit	factor	label
HGB	mmol/L	g/L	16.113	Haemoglobin
HGB	g/L	g/L	1	Haemoglobin
INSUL	mg/L	pmol/L	172.2	Insulin
INSUL	pmol/L	pmol/L	1	Insulin

Table 1: A small sample of a conversion dataset

Observations with factor 1 are present to handle raw data which are already provided with the desired SI-unit. This way a simple merge by "par\_name" and "rec\_unit" is possible without any pre-selection of parameters that need to be converted from others which don't.

Module 2a provides a dataset with lab values, converted into standard SI-units, the parameter's label, a patient identifier, the patient's demographic data needed to select the correct normal range (e.g. age, gender) and the sample date.

#### Module 2b) Normal range to SI

If normal ranges are delivered separately, Module 2b converts them the same way into SI units as Module 2a for lab values, and stores the normal ranges in a standardized dataset. For each laboratory parameter this dataset contains at least the following information: Laboratory identification, the converted lower and upper limit of the normal range, the corresponding unit, the lower and upper limit for age, the gender information and the two dates in SAS-format, indicating the validity start and end date of this particular range. For each combination of these variables there is one observation in the dataset. Additionally there might be information about the plausibility range for data validation purposes.

### Module 3) Laboratory values and normal range merge

Selecting the correct normal range for a lab value is complicated, as potentially many conditions (center, age interval, gender, version date period, ...) have to be taken into consideration. This is where PROC SQL shines. Using data step for merges with by variables fitting into a range of other variables (i.e. sample date into version date period, or age into lower and upper age interval) can not be easily done within one step in contrast to PROC SQL, where the code for the join could look like this:

```
proc sql;
  create table lab_final
  as select
      L.center,
      L.par_name,
      L.sex,
      L.age,
      L.sample_dt,
      N.agelow,
      N.agehigh,
      N.versdt_low,
      N.versdt_high

  from labnorm as N, labdat as L

  where L.center = N.center and
        L.par_name = N.par_name and
        L.sex = N.sex and
        L.age > N.agelow and
        L.age <= N.agehigh and
        L.sample_dt > N.versdt_low and
        L.sample_dt <= N.versdt_high
;
quit;
```

Module 3 provides the final output dataset for lab values. All values are converted into the SI-unit, parameter names are translated into the standard name and the correctly converted normal range for each value is present on the same observation. This dataset structure provides a very flexible basis for a wide variety of output layouts. It helps minimizing the programming effort needed to implement required changes.

### CONCLUSION (EXTENDIBILITY)

Laboratory data is inherently difficult to handle. The modular approach presented here offers a way to reduce the pain of working with it because of the following features:

- Handling different formats of input data is separated from the rest of the process. Module 1 (the import module) is the one most likely to change between studies. Because its output dataset is standardized the other modules should not be affected.
- Normalization of lab values and normal ranges is built around reusable resources (formats/dictionaries) that get better/more complete with each study they are used in.
- The final dataset will have all the necessary information to provide a good starting point to implement different output needs.

### REFERENCES

#### WEBSITE:

SI Units for Clinical Data  
(University of North Carolina at Chapel Hill)  
[http://www.unc.edu/~rowlett/units/scales/clinical\\_data.html](http://www.unc.edu/~rowlett/units/scales/clinical_data.html)

#### PAPERS:

Andrew T. Kuligowski  
Introducing External Data to the SAS® System – the Interactive Session  
SUGI 26, Long Beach California 2001

Fuping Peng  
"A Modular Approach to Develop Patient Profile Application"  
PharmaSUG Miami 2003

**e-BOOK:**

Ron Haas

"Clinical Lab SI Unit Conversion Factors" Database of 120+ common clinical lab analytes with conversion factors for standard US units to SI units and for SI to standard units.

Download: [http://www.memoware.com/?screen=doc\\_detail&doc\\_id=13546&p=category%5E!Medicine~!](http://www.memoware.com/?screen=doc_detail&doc_id=13546&p=category%5E!Medicine~!)

**ACKNOWLEDGEMENTS**

We would like to thank Ann Marie Martin (Novartis Pharma AG) and Michael Pickering (BIOP AG Basel) for their valuable input and encouragement on writing this paper.

**TRADEMARKS:**

SAS<sup>®</sup> is a registered trademark of SAS Institute Inc., in the USA and other countries.

**CONTACT INFORMATION:**

Benjamin Szilagy  
Senior Statistical Programmer  
Biostatistics & Statistical Reporting  
Novartis Pharma AG  
CH-4056 Basel  
Switzerland

Christof Binder  
Senior Head of Programmers  
BIOP AG  
Aeschenplatz 4  
CH-4052 Basel  
Switzerland